

# **IMiS/Storage Connector**

**Imaging Systems Inc.**

Copyright © 2010-2019 Imaging Systems Inc

---

**Package**  
**com.imis.storageconnector**

## com.imis.storageconnector Class ArchiveCapabilities

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.ArchiveCapabilities
  
```

**All Implemented Interfaces:**  
Serializable, Comparable

public final class **ArchiveCapabilities**  
extends Enum

Defines the archive capabilities.

**See Also:**

[IArchive](#)

### Field Summary

public static final	<a href="#">AUDIT_QUERY</a> Archive is authorized for audit log queries.
public static final	<a href="#">CONTENT_MANAGEMENT</a> Archive is authorized for content management operations.
public static final	<a href="#">DRAFT_MANAGEMENT</a> Archive is authorized for draft management operations.
public static final	<a href="#">IMPORT_EXPORT</a> Archive is authorized for import/export operations.
public static final	<a href="#">REPORTS</a> Archive is authorized for querying reports.

### Method Summary

static <a href="#">ArchiveCapabilities</a>	<a href="#">valueOf</a> (String name)
static <a href="#">ArchiveCapabilities[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

```
compareTo
```

---

## Fields

### AUDIT\_QUERY

```
public static final com.imis.storageconnector.ArchiveCapabilities AUDIT_QUERY
```

Archive is authorized for audit log queries.

---

### IMPORT\_EXPORT

```
public static final com.imis.storageconnector.ArchiveCapabilities IMPORT_EXPORT
```

Archive is authorized for import/export operations.

---

### CONTENT\_MANAGEMENT

```
public static final com.imis.storageconnector.ArchiveCapabilities CONTENT_MANAGEMENT
```

Archive is authorized for content management operations.

---

### REPORTS

```
public static final com.imis.storageconnector.ArchiveCapabilities REPORTS
```

Archive is authorized for querying reports.

---

### DRAFT\_MANAGEMENT

```
public static final com.imis.storageconnector.ArchiveCapabilities DRAFT_MANAGEMENT
```

Archive is authorized for draft management operations.

---

## Methods

### values

```
public final static ArchiveCapabilities\[\] values()
```

---

### valueOf

```
public static ArchiveCapabilities valueOf(String name)
```

## com.imis.storageconnector Class ArchiveType

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.ArchiveType
  
```

**All Implemented Interfaces:**  
Serializable, Comparable

public final class **ArchiveType**  
extends Enum

Specifies an IMiS/Storage Connector supported archiving system.

### Field Summary

public static final	<a href="#">IMIS_ARCHIVE</a> IMiS/ARChive Storage Server archive.
public static final	<a href="#">UNKNOWN</a> Unknown archive type.

### Method Summary

static <a href="#">ArchiveType</a>	<a href="#">valueOf</a> (String name)
static <a href="#">ArchiveType[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields

#### UNKNOWN

public static final com.imis.storageconnector.ArchiveType **UNKNOWN**

(continued from last page)

Unknown archive type.

This value defines an invalid archive type which should never be used.

---

## **IMIS\_ARCHIVE**

```
public static final com.imis.storageconnector.ArchiveType IMIS_ARCHIVE
```

IMiS/ARChive Storage Server archive.

## Methods

### **values**

```
public final static ArchiveType\[\] values()
```

---

### **valueOf**

```
public static ArchiveType valueOf(String name)
```

## com.imis.storageconnector

### Class AuditLog

java.lang.Object

└─com.imis.storageconnector.AuditLog

public abstract class **AuditLog**  
extends Object

Represents a custom audit log message to write to IMiS/ARC Storage Server in operations that support audit log feature.

Custom audit log message is made up of either a regular string or C style message format string together with the list of arguments.

Audit log feature is supported by the following operations:

Storage.createObject, Storage.openObject, Storage.moveObject, Storage.deleteObject and Document.save.

**Note:** Audit log feature requires IMiS/ARC Storage Server version 7 or higher.

### Constructor Summary

public	<a href="#">AuditLog()</a>
--------	----------------------------

### Method Summary

void	<a href="#">clear()</a> Clears the audit log message and arguments.
abstract List	<a href="#">getArguments()</a> Gets the list of audit log message arguments.
abstract String	<a href="#">getMessage()</a> Gets the audit log message Audit log message can be either a regular string or C style message format string.
abstract void	<a href="#">setArguments(List value)</a> Sets the list of audit log message arguments.
abstract void	<a href="#">setMessage(String value)</a> Sets the audit log message.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructors

#### AuditLog

public **AuditLog()**

## Methods

### getMessage

```
public abstract String getMessage()
```

Gets the audit log message

Audit log message can be either a regular string or C style message format string.

**Returns:**

The audit log message.

**See Also:**

[setMessage\(String\)](#)

---

### setMessage

```
public abstract void setMessage(String value)
```

Sets the audit log message.

Audit log message can be either a regular string or C style message format string.

When C style message format string is used for audit log message, use [setArguments\(List\)](#) method to set the appropriate message arguments.

When a regular string is used make sure you invalidate previously set message arguments.

**Parameters:**

value - the audit log message.

**See Also:**

[setArguments\(List\)](#)

---

### getArguments

```
public abstract List getArguments()
```

Gets the list of audit log message arguments.

**Returns:**

The list of audit log message arguments.

**See Also:**

[setMessage\(String\)](#)

---

### setArguments

```
public abstract void setArguments(List value)
```

Sets the list of audit log message arguments.

The null value sets the audit log message arguments to the empty list.

**Parameters:**

value - the list of audit log message arguments.

**See Also:**

---



(continued from last page)

[setMessage\(String\)](#)

---

## **clear**

```
public void clear()
```

Clears the audit log message and arguments.

## com.imis.storageconnector

### Class AuditQuery

java.lang.Object

└-com.imis.storageconnector.AuditQuery

public class **AuditQuery**  
extends Object

Represents the audit log query.

#### Nested Class Summary

class	<a href="#">AuditQuery.ObjectParamsScope</a> AuditQuery.ObjectParamsScope
class	<a href="#">AuditQuery.QueryParamsType</a> AuditQuery.QueryParamsType
class	<a href="#">AuditQuery.ResultFilter</a> AuditQuery.ResultFilter
class	<a href="#">AuditQuery.ResultSortOrder</a> AuditQuery.ResultSortOrder
class	<a href="#">AuditQuery.SessionParamsScope</a> AuditQuery.SessionParamsScope

#### Constructor Summary

public	<a href="#">AuditQuery()</a> Initializes a new instance of the <a href="#">AuditQuery</a> class.
--------	---

#### Method Summary

List	<a href="#">getAddresses()</a> Gets the list of addresses representing session parameters.
String	<a href="#">getAddressFrom()</a> Gets the starting address of the range of addresses representing session parameters.
String	<a href="#">getAddressTo()</a> Gets the ending address of the range of addresses representing session parameters.
String	<a href="#">getComputerNameFrom()</a> Gets the starting computer name of the range of computer names representing session parameters.
List	<a href="#">getComputerNames()</a> Gets the list of computer names representing session parameters.
String	<a href="#">getComputerNameTo()</a> Gets the ending computer name of the range of computer names representing session parameters.

Calendar	<a href="#"><u>getDateFrom()</u></a> Gets the starting timestamp of the range of timestamps parameters.
Calendar	<a href="#"><u>getDateTo()</u></a> Gets the ending timestamp of the range of timestamps parameters.
Integer	<a href="#"><u>getEventIdFrom()</u></a> Gets the starting event identifier of the range of event identifiers representing object parameters.
<a href="#"><u>AuditQuery.QueryParametersType</u></a>	<a href="#"><u>getEventIdQueryType()</u></a> Gets the query type used when event identifiers are used as object parameters.
List	<a href="#"><u>getEventIds()</u></a> Gets the list of event identifiers representing object parameters.
Integer	<a href="#"><u>getEventIdTo()</u></a> Gets the ending event identifier of the range of event identifiers representing object parameters.
Set	<a href="#"><u>getFilter()</u></a> Gets the filter for the audit query results.
List	<a href="#"><u>getObjectIds()</u></a> Gets the list of object identifiers representing object parameters.
<a href="#"><u>AuditQuery.ObjectParametersScope</u></a>	<a href="#"><u>getObjectScope()</u></a> Gets the object parameters scope.
<a href="#"><u>AuditQuery.QueryParametersType</u></a>	<a href="#"><u>getSessionQueryType()</u></a> Gets the session parameters query type.
<a href="#"><u>AuditQuery.SessionParametersScope</u></a>	<a href="#"><u>getSessionScope()</u></a> Gets the session parameters scope.
<a href="#"><u>AuditQuery.ResultSortOrder</u></a>	<a href="#"><u>getSortOrder()</u></a> Gets the sort order of the audit query results.
String	<a href="#"><u>getUserNameFrom()</u></a> Gets the starting user name of the range of user names representing session parameters.
List	<a href="#"><u>getUserNames()</u></a> Gets the list of user names representing session parameters.
String	<a href="#"><u>getUserNameTo()</u></a> Gets the ending user name of the range of user names representing session parameters.
boolean	<a href="#"><u>isRecursive()</u></a> Gets the value indicating whether or not audit query should be performed recursively on selected object identifiers.
boolean	<a href="#"><u>isValid()</u></a> Gets the value indicating whether or not this object has a valid set of parameters.
void	<a href="#"><u>setAddressFrom()</u></a> (String value) Sets the starting address of the range of addresses representing session parameters.
void	<a href="#"><u>setAddressTo()</u></a> (String value) Sets the ending address of the range of addresses representing session parameters.

void	<a href="#"><u>setComputerNameFrom</u></a> (String value) Sets the starting computer name of the range of computer names representing session parameters.
void	<a href="#"><u>setComputerNameTo</u></a> (String value) Sets the ending computer name of the range of computer names representing session parameters.
void	<a href="#"><u>setDateFrom</u></a> (Calendar value) Sets the starting timestamp of the range of timestamps parameters.
void	<a href="#"><u>setDateTo</u></a> (Calendar value) Sets the ending timestamp of the range of timestamps parameters.
void	<a href="#"><u>setEventIdFrom</u></a> (Integer value) Sets the starting event identifier of the range of event identifiers representing object parameters.
void	<a href="#"><u>setEventIdQueryType</u></a> ( <a href="#"><u>AuditQuery.QueryParamsType</u></a> value) Sets the query type used when event identifiers are used as object parameters.
void	<a href="#"><u>setEventIdTo</u></a> (Integer value) Sets the ending event identifier of the range of event identifiers representing object parameters.
void	<a href="#"><u>setFilter</u></a> (Set value) Sets the filter for the audit query results.
void	<a href="#"><u>setObjectScope</u></a> ( <a href="#"><u>AuditQuery.ObjectParamsScope</u></a> value) Sets the object parameters scope.
void	<a href="#"><u>setRecursive</u></a> (boolean value) Sets the value indicating whether or not audit query should be performed recursively on selected object identifiers.
void	<a href="#"><u>setSessionQueryType</u></a> ( <a href="#"><u>AuditQuery.QueryParamsType</u></a> value) Sets the session parameters query type.
void	<a href="#"><u>setSessionScope</u></a> ( <a href="#"><u>AuditQuery.SessionParamsScope</u></a> value) Sets the session parameters scope.
void	<a href="#"><u>setSortOrder</u></a> ( <a href="#"><u>AuditQuery.ResultSortOrder</u></a> value) Sets the sort order of the audit query results.
void	<a href="#"><u>setUserNameFrom</u></a> (String value) Sets the starting user name of the range of user names representing session parameters.
void	<a href="#"><u>setUserNameTo</u></a> (String value) Sets the ending user name of the range of user names representing session parameters.

#### Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

(continued from last page)

## AuditQuery

```
public AuditQuery()
```

Initializes a new instance of the [AuditQuery](#) class.

## Methods

### getSessionQueryType

```
public AuditQuery.QueryParamsType getSessionQueryType()
```

Gets the session parameters query type.

**Returns:**

The session parameters query type. Defaults to `QueryParamsType.LIST`.

### setSessionQueryType

```
public void setSessionQueryType(AuditQuery.QueryParamsType value)
```

Sets the session parameters query type.

**Parameters:**

value - the session parameters query type.

### getSessionScope

```
public AuditQuery.SessionParamsScope getSessionScope()
```

Gets the session parameters scope.

**Returns:**

The session parameters scope. Defaults to `SessionParamsScope.Addresses`.

### setSessionScope

```
public void setSessionScope(AuditQuery.SessionParamsScope value)
```

Sets the session parameters scope.

**Parameters:**

value - the session parameters scope.

### getAddresses

```
public List getAddresses()
```

Gets the list of addresses representing session parameters.

Address format is `X.X.X.X/R`, where `X` is a number from 0 to 255 and `R` is a number from 0 to 32 and represents a number of bits used in the audit query search.

**Returns:**

The list of addresses as `String` objects.

(continued from last page)

---

## getAddressFrom

```
public String getAddressFrom()
```

Gets the starting address of the range of addresses representing session parameters.

Address format is  $X.X.X.X/R$ , where  $X$  is a number from 0 to 255 and  $R$  is a number from 0 to 32 and represents a number of bits used in the audit query search.

**Returns:**

The address that starts the range of addresses or `null` if there is no lower limit. Defaults to `null`.

---

## setAddressFrom

```
public void setAddressFrom(String value)
```

Sets the starting address of the range of addresses representing session parameters.

Address format is  $X.X.X.X/R$ , where  $X$  is a number from 0 to 255 and  $R$  is a number from 0 to 32 and represents a number of bits used in the audit query search.

**Parameters:**

`value` - the address that starts the range of addresses or `null` if there is no lower limit.

---

## getAddressTo

```
public String getAddressTo()
```

Gets the ending address of the range of addresses representing session parameters.

Address format is  $X.X.X.X/R$ , where  $X$  is a number from 0 to 255 and  $R$  is a number from 0 to 32 and represents a number of bits used in the audit query search.

**Returns:**

The address that ends the range of addresses or `null` if there is no upper limit. Defaults to `null`.

---

## setAddressTo

```
public void setAddressTo(String value)
```

Sets the ending address of the range of addresses representing session parameters.

Address format is  $X.X.X.X/R$ , where  $X$  is a number from 0 to 255 and  $R$  is a number from 0 to 32 and represents a number of bits used in the audit query search.

**Parameters:**

`value` - the address that ends the range of addresses or `null` if there is no upper limit.

---

## getUserNames

```
public List getUserNames()
```

Gets the list of user names representing session parameters.

**Returns:**

The list of user names as `String` objects.

---

## getUserNameFrom

```
public String getUserNameFrom()
```

---

(continued from last page)

Gets the starting user name of the range of user names representing session parameters.

**Returns:**

The user name that starts the range of user names or `null` if there is no lower limit. Defaults to `null`.

---

## setUserNameFrom

```
public void setUserNameFrom(String value)
```

Sets the starting user name of the range of user names representing session parameters.

**Parameters:**

`value` - The user name that starts the range of user names or `null` if there is no lower limit.

---

## getUserNameTo

```
public String getUserNameTo()
```

Gets the ending user name of the range of user names representing session parameters.

**Returns:**

The user name that ends the range of user names or `null` if there is no upper limit. Defaults to `null`.

---

## setUserNameTo

```
public void setUserNameTo(String value)
```

Sets the ending user name of the range of user names representing session parameters.

**Parameters:**

`value` - The user name that ends the range of user names or `null` if there is no upper limit.

---

## getComputerNames

```
public List getComputerNames()
```

Gets the list of computer names representing session parameters.

**Returns:**

The list of computer names as `String` objects.

---

## getComputerNameFrom

```
public String getComputerNameFrom()
```

Gets the starting computer name of the range of computer names representing session parameters.

**Returns:**

The computer name that starts the range of computer names or `null` if there is no lower limit. Defaults to `null`.

---

## setComputerNameFrom

```
public void setComputerNameFrom(String value)
```

Sets the starting computer name of the range of computer names representing session parameters.

**Parameters:**

`value` - the computer name that starts the range of computer names or `null` if there is no lower limit.

## getComputerNameTo

```
public String getComputerNameTo()
```

Gets the ending computer name of the range of computer names representing session parameters.

**Returns:**

The computer name that ends the range of computer names or `null` if there is no upper limit. Defaults to `null`.

---

## setComputerNameTo

```
public void setComputerNameTo(String value)
```

Sets the ending computer name of the range of computer names representing session parameters.

**Parameters:**

`value` - the computer name that ends the range of computer names or `null` if there is no upper limit.

---

## getObjectScope

```
public AuditQuery.ObjectParamsScope getObjectScope()
```

Gets the object parameters scope.

**Returns:**

The object parameters scope. Defaults to `ObjectParamsScope.OBJECT_IDS`.

---

## setObjectScope

```
public void setObjectScope(AuditQuery.ObjectParamsScope value)
```

Sets the object parameters scope.

**Parameters:**

`value` - the object parameters scope.

---

## getObjectIds

```
public List getObjectIds()
```

Gets the list of object identifiers representing object parameters.

**Returns:**

The list of object identifiers as `String` objects.

---

## getEventIdQueryType

```
public AuditQuery.QueryParamsType getEventIdQueryType()
```

Gets the query type used when event identifiers are used as object parameters.

**Returns:**

The event identifiers parameters query type. Defaults to `QueryParamsType.LIST`.

---

## setEventIdQueryType

```
public void setEventIdQueryType(AuditQuery.QueryParamsType value)
```

---



(continued from last page)

Sets the query type used when event identifiers are used as object parameters.

**Parameters:**

value - the event identifiers parameters query type.

---

## getEventIds

```
public List getEventIds()
```

Gets the list of event identifiers representing object parameters.

**Returns:**

The list of event identifiers as `Integer` values.

---

## getEventIdFrom

```
public Integer getEventIdFrom()
```

Gets the starting event identifier of the range of event identifiers representing object parameters.

**Returns:**

The event identifier that starts the range of event identifiers or `null` if there is no lower limit. Defaults to `null`.

---

## setEventIdFrom

```
public void setEventIdFrom(Integer value)
```

Sets the starting event identifier of the range of event identifiers representing object parameters.

**Parameters:**

value - the event identifier that starts the range of event identifiers or `null` if there is no lower limit.

---

## getEventIdTo

```
public Integer getEventIdTo()
```

Gets the ending event identifier of the range of event identifiers representing object parameters.

**Returns:**

The event identifier that ends the range of event identifiers or `null` if there is no upper limit. Defaults to `null`.

---

## setEventIdTo

```
public void setEventIdTo(Integer value)
```

Sets the ending event identifier of the range of event identifiers representing object parameters.

**Parameters:**

value - the event identifier that ends the range of event identifiers or `null` if there is no upper limit.

---

## getDateFrom

```
public Calendar getDateFrom()
```

Gets the starting timestamp of the range of timestamps parameters.

**Returns:**

The timestamp that starts the range of timestamps parameters or `null` if there is no lower limit. Defaults to `null`.

## setDateFrom

```
public void setDateFrom(Calendar value)
```

Sets the starting timestamp of the range of timestamps parameters.

**Parameters:**

value - the timestamp that starts the range of timestamps parameters or null if there is no lower limit.

---

## getDateTo

```
public Calendar getDateTo()
```

Gets the ending timestamp of the range of timestamps parameters.

**Returns:**

The timestamp that ends the range of timestamps parameters or null if there is no upper limit. Defaults to null.

---

## setDateTo

```
public void setDateTo(Calendar value)
```

Sets the ending timestamp of the range of timestamps parameters.

**Parameters:**

value - the timestamp that ends the range of timestamps parameters or null if there is no upper limit.

---

## getSortOrder

```
public AuditQuery.ResultSortOrder getSortOrder()
```

Gets the sort order of the audit query results.

**Returns:**

The sort order of the audit query results. Defaults to `ResultSortOrder.SESSION_TIMESTAMP_OBJECT`.

---

## setSortOrder

```
public void setSortOrder(AuditQuery.ResultSortOrder value)
```

Sets the sort order of the audit query results.

**Parameters:**

value - the sort order of the audit query results.

---

## getFilter

```
public Set getFilter()
```

Gets the filter for the audit query results.

This property has effect when [getObjectScope\(\)](#) is equal to `ObjectParamsScope.OBJECT_IDS`.

**Returns:**

The filter for the audit query object results. Default value is set to `EnumSet.allOf(ResultFilter.class)`.

---

(continued from last page)

---

## setFilter

```
public void setFilter(Set value)
```

Sets the filter for the audit query results.

### Parameters:

value - the filter for the audit query object results.

---

## isRecursive

```
public boolean isRecursive()
```

Gets the value indicating whether or not audit query should be performed recursively on selected object identifiers.

### Returns:

true if audit query should be performed recursively on selected object identifiers; otherwise false. Default value is set to false.

### See Also:

[getObjectIds\(\)](#)

---

## setRecursive

```
public void setRecursive(boolean value)
```

Sets the value indicating whether or not audit query should be performed recursively on selected object identifiers.

### Parameters:

value - true if audit query should be performed recursively on selected object identifiers; otherwise false.

### See Also:

[getObjectIds\(\)](#)

---

## isValid

```
public boolean isValid()
```

Gets the value indicating whether or not this object has a valid set of parameters.

### Returns:

true if AuditQuery has a valid set of parameters; otherwise false.

com.imis.storageconnector

Class AuditQuery.QueryParamsType



All Implemented Interfaces:  
Serializable, Comparable

public static final class **AuditQuery.QueryParamsType**  
extends Enum

Defines the audit log query parameters query types.

Field Summary	
public static final	<a href="#">LIST</a> Query parameters are organized as list.
public static final	<a href="#">RANGE</a> Query parameters are organized as range.

Method Summary	
static <a href="#">AuditQuery.QueryParamsType</a>	<a href="#">valueOf</a> (String name)
static <a href="#">AuditQuery.QueryParamsType[]</a>	<a href="#">values</a> ()

Methods inherited from class java.lang.Enum
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface java.lang.Comparable
compareTo

Fields

(continued from last page)

## LIST

```
public static final com.imis.storageconnector.AuditQuery.QueryParamsType LIST
```

Query parameters are organized as list.

---

## RANGE

```
public static final com.imis.storageconnector.AuditQuery.QueryParamsType RANGE
```

Query parameters are organized as range.

## Methods

### values

```
public final static AuditQuery.QueryParamsType\[\] values()
```

---

### valueOf

```
public static AuditQuery.QueryParamsType valueOf(String name)
```

## com.imis.storageconnector Class AuditQuery.SessionParamsScope

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.AuditQuery.SessionParamsScope
  
```

**All Implemented Interfaces:**  
Serializable, Comparable

```

public static final class AuditQuery.SessionParamsScope
extends Enum
  
```

Defines the audit log query session parameters scope types.

### Field Summary

public static final	<a href="#">ADDRESSES</a> Scope is limited with the list or range of IP addresses.
public static final	<a href="#">COMPUTER_NAMES</a> Scope is limited with the list or range of computer names.
public static final	<a href="#">USER_NAMES</a> Scope is limited with the list or range of user names.

### Method Summary

static <a href="#">AuditQuery.SessionParamsScope</a>	<a href="#">valueOf</a> (String name)
static <a href="#">AuditQuery.SessionParamsScope[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields

(continued from last page)

## ADDRESSES

```
public static final com.imis.storageconnector.AuditQuery.SessionParamsScope ADDRESSES
```

Scope is limited with the list or range of IP addresses.

---

## USER\_NAMES

```
public static final com.imis.storageconnector.AuditQuery.SessionParamsScope USER_NAMES
```

Scope is limited with the list or range of user names.

---

## COMPUTER\_NAMES

```
public static final com.imis.storageconnector.AuditQuery.SessionParamsScope  
COMPUTER_NAMES
```

Scope is limited with the list or range of computer names.

## Methods

### values

```
public final static AuditQuery.SessionParamsScope\[\] values()
```

---

### valueOf

```
public static AuditQuery.SessionParamsScope valueOf(String name)
```

## com.imis.storageconnector Class AuditQuery.ObjectParamsScope

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.AuditQuery.ObjectParamsScope
  
```

**All Implemented Interfaces:**  
Serializable, Comparable

public static final class **AuditQuery.ObjectParamsScope**  
extends Enum

Defines the audit log query object parameters scope types.

### Field Summary

public static final	<a href="#">EVENT_IDS</a> Scope is limited with the list or range of event identifiers.
public static final	<a href="#">OBJECT_IDS</a> Scope is limited with the list of object identifiers.

### Method Summary

static <a href="#">AuditQuery.ObjectParamsScope</a>	<a href="#">valueOf</a> (String name)
static <a href="#">AuditQuery.ObjectParamsScope[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields



(continued from last page)

## OBJECT\_IDS

```
public static final com.imis.storageconnector.AuditQuery.ObjectParamsScope OBJECT_IDS
```

Scope is limited with the list of object identifiers.

Ranges are not supported with object identifiers.

---

## EVENT\_IDS

```
public static final com.imis.storageconnector.AuditQuery.ObjectParamsScope EVENT_IDS
```

Scope is limited with the list or range of event identifiers.

## Methods

### values

```
public final static AuditQuery.ObjectParamsScope\[\] values()
```

---

### valueOf

```
public static AuditQuery.ObjectParamsScope valueOf(String name)
```

## com.imis.storageconnector

### Class AuditQuery.ResultSortOrder

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.AuditQuery.ResultSortOrder
  
```

#### All Implemented Interfaces:

Serializable, Comparable

public static final class **AuditQuery.ResultSortOrder**  
extends Enum

Defines the audit query results sort order types.

### Field Summary

public static final	<a href="#"><u>OBJECT_SESSION_TIMESTAMP</u></a> Sort objects by sessions and timestamps.
public static final	<a href="#"><u>OBJECT_TIMESTAMP_SESSION</u></a> Sort object by timestamps and sessions.
public static final	<a href="#"><u>SESSION_OBJECT_TIMESTAMP</u></a> Sort sessions by objects and timestamps.
public static final	<a href="#"><u>SESSION_TIMESTAMP_OBJECT</u></a> Sort sessions by timestamps and objects.
public static final	<a href="#"><u>TIMESTAMP_OBJECT_SESSION</u></a> Sort timestamps by object and sessions.
public static final	<a href="#"><u>TIMESTAMP_SESSION_OBJECT</u></a> Sort timestamps by sessions and objects.

### Method Summary

static <a href="#"><u>AuditQuery.ResultSortOrder</u></a>	<a href="#"><u>valueOf</u></a> (String name)
static <a href="#"><u>AuditQuery.ResultSortOrder[]</u></a>	<a href="#"><u>values</u></a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods inherited from interface** `java.lang.Comparable``compareTo`

## Fields

### SESSION\_TIMESTAMP\_OBJECT

```
public static final com.imis.storageconnector.AuditQuery.ResultSortOrder  
SESSION_TIMESTAMP_OBJECT
```

Sort sessions by timestamps and objects.

### SESSION\_OBJECT\_TIMESTAMP

```
public static final com.imis.storageconnector.AuditQuery.ResultSortOrder  
SESSION_OBJECT_TIMESTAMP
```

Sort sessions by objects and timestamps.

### TIMESTAMP\_SESSION\_OBJECT

```
public static final com.imis.storageconnector.AuditQuery.ResultSortOrder  
TIMESTAMP_SESSION_OBJECT
```

Sort timestamps by sessions and objects.

### TIMESTAMP\_OBJECT\_SESSION

```
public static final com.imis.storageconnector.AuditQuery.ResultSortOrder  
TIMESTAMP_OBJECT_SESSION
```

Sort timestamps by object and sessions.

### OBJECT\_SESSION\_TIMESTAMP

```
public static final com.imis.storageconnector.AuditQuery.ResultSortOrder  
OBJECT_SESSION_TIMESTAMP
```

Sort objects by sessions and timestamps.

### OBJECT\_TIMESTAMP\_SESSION

```
public static final com.imis.storageconnector.AuditQuery.ResultSortOrder  
OBJECT_TIMESTAMP_SESSION
```

Sort object by timestamps and sessions.

## Methods

### values

```
public final static AuditQuery.ResultSortOrder\[\] values()
```

---

## valueOf

public static [AuditQuery.ResultSortOrder](#) **valueOf**(String name)

## com.imis.storageconnector Class AuditQuery.ResultFilter

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.AuditQuery.ResultFilter
  
```

**All Implemented Interfaces:**  
Serializable, Comparable

public static final class **AuditQuery.ResultFilter**  
extends Enum

Defines enumeration for audit query results filtering.

### Field Summary

public static final	<a href="#">CLASSES</a> Entities of type <a href="#">EntityType.CLASS</a> should be included in audit query results.
public static final	<a href="#">DOCUMENTS</a> Entities of type <a href="#">EntityType.DOCUMENT</a> should be included in audit query results.
public static final	<a href="#">FOLDERS</a> Entities of type <a href="#">EntityType.FOLDER</a> should be included in audit query results.

### Method Summary

static <a href="#">AuditQuery.ResultFilter</a>	<a href="#">valueOf</a> (String name)
static <a href="#">AuditQuery.ResultFilter</a> <a href="#">er[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields

(continued from last page)

## CLASSES

public static final com.imis.storageconnector.AuditQuery.ResultFilter **CLASSES**

Entities of type [EntityType.CLASS](#) should be included in audit query results.

## FOLDERS

public static final com.imis.storageconnector.AuditQuery.ResultFilter **FOLDERS**

Entities of type [EntityType.FOLDER](#) should be included in audit query results.

## DOCUMENTS

public static final com.imis.storageconnector.AuditQuery.ResultFilter **DOCUMENTS**

Entities of type [EntityType.DOCUMENT](#) should be included in audit query results.

## Methods

### values

public final static [AuditQuery.ResultFilter\[\]](#) **values()**

### valueOf

public static [AuditQuery.ResultFilter](#) **valueOf**(String name)

## com.imis.storageconnector Class AuthCryptoAlgorithm

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.AuthCryptoAlgorithm
  
```

### All Implemented Interfaces:

Serializable, Comparable

public final class **AuthCryptoAlgorithm**  
extends Enum

Defines constants that represent the type of the cryptographic algorithm used when establishing a connection with IMiS/ARC Storage Server.

Constants are used as values for the [StorageConnector.OPTION\\_AUTH\\_CRYPTTO\\_ALGORITHM](#) IMIS\_ARC storage authentication cryptographic algorithm option.

### See Also:

[StorageConnector.openArchive\(ArchiveType, String, Map\)](#),  
[StorageConnector.openArchive\(ArchiveType, String, int, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, String, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, int, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, int, String, Map\)](#)

## Field Summary

public static final	<a href="#">AES</a> Advanced Encryption Standard
---------------------	---

## Method Summary

static <a href="#">AuthCryptoAlgorithm</a>	<a href="#">valueOf</a> (String name)
---	---------------------------------------

static <a href="#">AuthCryptoAlgorithm[]</a>	<a href="#">values</a> ()
---	---------------------------

### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods inherited from interface java.lang.Comparable

compareTo

(continued from last page)

## Fields

### AES

```
public static final com.imis.storageconnector.AuthCryptoAlgorithm AES
```

Advanced Encryption Standard

## Methods

### values

```
public final static AuthCryptoAlgorithm\[\] values()
```

---

### valueOf

```
public static AuthCryptoAlgorithm valueOf(String name)
```



## com.imis.storageconnector Class AuthCryptoCipherMode

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.AuthCryptoCipherMode
  
```

### All Implemented Interfaces:

Serializable, Comparable

public final class **AuthCryptoCipherMode**  
extends Enum

Defines constants that represent the block cipher mode of operation used when establishing a connection with IMiS/ARC Storage Server.

Constants are used as values for the [StorageConnector.OPTION\\_AUTH\\_CRYPTOCIPHERMODE](#) IMIS\_ARC storage authentication block cipher mode option.

### See Also:

[StorageConnector.openArchive\(ArchiveType, String, Map\)](#),  
[StorageConnector.openArchive\(ArchiveType, String, int, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, String, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, int, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, int, String, Map\)](#)

## Field Summary

public static final	<a href="#">CBC</a> Cipher-block chaining (CBC) mode
public static final	<a href="#">CFB</a> Cipher Feedback (CFB) mode (128bit)
public static final	<a href="#">ECB</a> Electronic codebook (EBC) mode
public static final	<a href="#">OFB</a> Output feedback (OFB) mode (128bit)

## Method Summary

static <a href="#">AuthCryptoCipherMode</a>	<a href="#">valueOf</a> (String name)
static <a href="#">AuthCryptoCipherMode</a> [ ]	<a href="#">values</a> ()

### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

**Methods inherited from interface** `java.lang.Comparable`

```
compareTo
```

## Fields

### CBC

```
public static final com.imis.storageconnector.AuthCryptoCipherMode CBC
```

Cipher-block chaining (CBC) mode

### ECB

```
public static final com.imis.storageconnector.AuthCryptoCipherMode ECB
```

Electronic codebook (EBC) mode

### OFB

```
public static final com.imis.storageconnector.AuthCryptoCipherMode OFB
```

Output feedback (OFB) mode (128bit)

### CFB

```
public static final com.imis.storageconnector.AuthCryptoCipherMode CFB
```

Cipher Feedback (CFB) mode (128bit)

## Methods

### values

```
public final static AuthCryptoCipherMode\[\] values()
```

### valueOf

```
public static AuthCryptoCipherMode valueOf(String name)
```

# com.imis.storageconnector

## Class AuthCryptoKeySize



All Implemented Interfaces:  
Serializable, Comparable

public final class **AuthCryptoKeySize**  
extends Enum

Defines constants that represent the size, and hence the strength, of the cryptographic key used when establishing a connection with IMiS/ARC Storage Server.

Constants are used as values for the [StorageConnector.OPTION\\_AUTH\\_CRYPTOKEYSIZE](#) IMIS\_ARC storage authentication key size option.  
**See Also:**

```
StorageConnector.openArchive(ArchiveType, String, Map),
StorageConnector.openArchive(ArchiveType, String, int, Map),
StorageConnector.openIMiSARCStorage(String, Map),
StorageConnector.openIMiSARCStorage(String, String, Map),
StorageConnector.openIMiSARCStorage(String, int, Map),
StorageConnector.openIMiSARCStorage(String, int, String, Map)
```

Field Summary	
public static final	<a href="#">KS128</a> Key strength: 128bit
public static final	<a href="#">KS192</a> Key strength: 192bit
public static final	<a href="#">KS256</a> Key strength: 256bit

Method Summary	
static <a href="#">AuthCryptoKeySize</a>	<a href="#">valueOf</a> (String name)
static <a href="#">AuthCryptoKeySize[]</a>	<a href="#">values</a> ()

Methods inherited from class java.lang.Enum	
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf	

Methods inherited from class java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

**Methods inherited from interface** `java.lang.Comparable``compareTo`

## Fields

### KS128

```
public static final com.imis.storageconnector.AuthCryptoKeySize KS128
```

Key strength: 128bit

### KS192

```
public static final com.imis.storageconnector.AuthCryptoKeySize KS192
```

Key strength: 192bit

### KS256

```
public static final com.imis.storageconnector.AuthCryptoKeySize KS256
```

Key strength: 256bit

## Methods

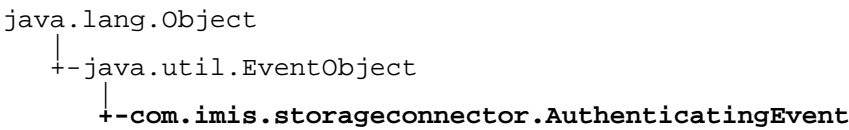
### values

```
public final static AuthCryptoKeySize\[\] values()
```

### valueOf

```
public static AuthCryptoKeySize valueOf(String name)
```

# com.imis.storageconnector Class AuthenticatingEvent



All Implemented Interfaces:  
Serializable

public final class **AuthenticatingEvent**  
extends EventObject

Provides data for authenticating event.

Fields inherited from class java.util.EventObject	
	source

Constructor Summary	
public	<a href="#">AuthenticatingEvent</a> (Object source, <a href="#">AuthType</a> type, Map options) Initializes a new instance of the AuthenticatingEventArgs class.

Method Summary	
boolean	<a href="#">cancel</a> () Gets the value indicating whether or not to cancel authentication.
Map	<a href="#">getOptions</a> () Gets the map of objects containing authentication data sorted by authentication options.
<a href="#">AuthType</a>	<a href="#">getType</a> () Gets the authentication type.
void	<a href="#">setCancel</a> (boolean value) Sets the value indicating whether or not to cancel authentication.

Methods inherited from class java.util.EventObject	
	getSource, toString

Methods inherited from class java.lang.Object	
	clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

(continued from last page)

## AuthenticatingEvent

```
public AuthenticatingEvent(Object source,  
                           AuthType type,  
                           Map options)
```

Initializes a new instance of the `AuthenticatingEventArgs` class.

**Parameters:**

source - the source of the event.

type - the authentication type.

options - the map of objects containing authentication data sorted by authentication options.

## Methods

### getType

```
public AuthType getType()
```

Gets the authentication type.

**Returns:**

The authentication type.

### getOptions

```
public Map getOptions()
```

Gets the map of objects containing authentication data sorted by authentication options.

**Returns:**

The map of objects containing authentication data sorted by authentication options.

### cancel

```
public boolean cancel()
```

Gets the value indicating whether or not to cancel authentication.

**Returns:**

true to cancel authentication; otherwise false. Default value is false.

### setCancel

```
public void setCancel(boolean value)
```

Sets the value indicating whether or not to cancel authentication.

**Parameters:**

value - true to cancel authentication; otherwise false.

## com.imis.storageconnector Class AuthType

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.AuthType
  
```

**All Implemented Interfaces:**  
Serializable, Comparable

public final class **AuthType**  
extends Enum

Defines constants that represent the type of the authentication used when establishing a connection with IMiS/ARC Storage Server.

Constants are used as values for the authentication key type option on IMIS\_ARC [Storage](#) or IMIS\_ARCHIVE [IArchive](#).

**See Also:**

[StorageConnector.OPTION\\_AUTH\\_TYPE](#), [StorageConnector.openIMiSARCStorage\(String, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, String, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, int, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, int, String, Map\)](#),  
[StorageConnector.openArchive\(ArchiveType, String, Map\)](#),  
[StorageConnector.openArchive\(ArchiveType, String, int, Map\)](#)

### Field Summary

public static final	<a href="#">KERBEROS5_SERVICE_TICKET</a> Kerberos 5 Ticket Granting Server (KRB5TGS) authentication (supported by IMiS/ARChive Storage Server v9.7 or later).
public static final	<a href="#">LDAP_CREDENTIALS</a> Lightweight Directory Access Protocol (LDAP) credentials (supported by IMiS/ARChive Storage Server v9.7 or later).
public static final	<a href="#">LOCAL_CREDENTIALS</a> Credentials for users registered in IMiS/ARChive local database (supported by IMiS/ARChive Storage Server v9.0 or later).
public static final	<a href="#">PRE_AUTHENTICATE</a> Provides discovery of authentication methods (supported by IMiS/ARChive Storage Server v9.7 or later).
public static final	<a href="#">PRE_SHARED_KEY</a> Pre shared key.

### Method Summary

static <a href="#">AuthType</a>	<a href="#">valueOf</a> (String name)
static <a href="#">AuthType[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

**Methods inherited from class** `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

**Methods inherited from interface** `java.lang.Comparable`

`compareTo`

## Fields

### PRE\_SHARED\_KEY

```
public static final com.imis.storageconnector.AuthType PRE_SHARED_KEY
```

Pre shared key.

### LOCAL\_CREDENTIALS

```
public static final com.imis.storageconnector.AuthType LOCAL_CREDENTIALS
```

Credentials for users registered in IMiS/ARChive local database (supported by IMiS/ARChive Storage Server v9.0 or later).

### LDAP\_CREDENTIALS

```
public static final com.imis.storageconnector.AuthType LDAP_CREDENTIALS
```

Lightweight Directory Access Protocol (LDAP) credentials (supported by IMiS/ARChive Storage Server v9.7 or later).

### KERBEROS5\_SERVICE\_TICKET

```
public static final com.imis.storageconnector.AuthType KERBEROS5_SERVICE_TICKET
```

Kerberos 5 Ticket Granting Server (KRB5TGS) authentication (supported by IMiS/ARChive Storage Server v9.7 or later).

### PRE\_AUTHENTICATE

```
public static final com.imis.storageconnector.AuthType PRE_AUTHENTICATE
```

Provides discovery of authentication methods (supported by IMiS/ARChive Storage Server v9.7 or later).

## Methods

#### values

```
public final static AuthType\[\] values()
```

#### valueOf

```
public static AuthType valueOf(String name)
```



## com.imis.storageconnector Class ConnectingEvent

```

java.lang.Object
  |
  +- java.util.EventObject
        |
        +- com.imis.storageconnector.ConnectingEvent
  
```

**All Implemented Interfaces:**  
Serializable

public class **ConnectingEvent**  
extends EventObject

Provides data for connecting event.

### Fields inherited from class java.util.EventObject

source

### Constructor Summary

public	<a href="#">ConnectingEvent</a> (Object source, com.imis.imisarc.client.ConnectingEvent e) Initializes a new instance of the ConnectingEvent class for session initialization stage.
--------	---

### Method Summary

boolean	<a href="#">cancel</a> () Gets the value indicating whether or not to cancel connecting.
java.security.Principal[]	<a href="#">getAcceptableIssuers</a> () Gets the CA certificate issuers subject names acceptable to the remote party, used when authenticating the client side.
String	<a href="#">getAuthenticationType</a> () Gets the key exchange algorithm used to authenticate the remote party.
String[]	<a href="#">getKeyTypes</a> () Gets the key algorithm type name(s), ordered with the most-preferred key type first, used when authenticating the client side.
java.security.cert.X509Certificate	<a href="#">getLocalCertificate</a> () Gets the local certificate used for establishing an SSL connection in the local certificate selection stage.
java.security.PrivateKey	<a href="#">getLocalCertificateKey</a> () Gets the local certificate private key used for establishing an SSL connection in the local certificate selection stage.
java.net.SocketAddresses	<a href="#">getLocalSocketAddress</a> () Gets the local socket address.

<code>javax.security.auth.callback.PasswordCallback</code>	<a href="#"><code>getPasswordCallback()</code></a> Gets the password callback used to provide the password for JRE default trusted CA certificates key store that is used to retrieve the trust anchors for remote certificate validation if they are not provided with <a href="#"><code>setTrustAnchors(Set)</code></a> method.
<a href="#"><code>ConnectionProtocol</code></a>	<a href="#"><code>getProtocol()</code></a> Gets the session protocol.
<code>java.security.cert.X509Certificate[]</code>	<a href="#"><code>getRemoteCertificateChain()</code></a> Gets the certificate chain used to authenticate the remote party.
<code>String</code>	<a href="#"><code>getRemoteHost()</code></a> Gets the remote host.
<code>java.net.SocketAddresses</code>	<a href="#"><code>getRemoteSocketAddress()</code></a> Gets the remote socket address.
<code>String[]</code>	<a href="#"><code>getSslProtocols()</code></a> Gets the names of the enabled SSL protocols used by the session.
<a href="#"><code>ConnectingStage</code></a>	<a href="#"><code>getStage()</code></a> Gets the connecting event stage.
<code>Set</code>	<a href="#"><code>getTrustAnchors()</code></a> Gets the trust anchors used for remote certificate verification.
<code>boolean</code>	<a href="#"><code>isRemoteCertificateAccepted()</code></a> Gets the value indicating whether or not the specified remote certificate chain is accepted for authentication in the remote certificate validation stage.
<code>void</code>	<a href="#"><code>setCancel(boolean value)</code></a> Sets the value indicating whether or not to cancel connecting.
<code>void</code>	<a href="#"><code>setLocalCertificate(java.security.cert.X509Certificate value)</code></a> Sets the local certificate used for establishing an SSL connection in the local certificate selection stage.
<code>void</code>	<a href="#"><code>setLocalCertificateKey(java.security.PrivateKey value)</code></a> Sets the local certificate private key used for establishing an SSL connection in the local certificate selection stage.
<code>void</code>	<a href="#"><code>setRemoteCertificateAccepted(boolean value)</code></a> Sets the value indicating whether or not the specified remote certificate is accepted for authentication in the remote certificate validation stage.
<code>void</code>	<a href="#"><code>setSslProtocols(String[] value)</code></a> Sets the names of the enabled SSL protocols used by the session.
<code>void</code>	<a href="#"><code>setTrustAnchors(Set value)</code></a> Sets the trust anchors used for remote certificate verification that can be set in session initialization stage.

**Methods inherited from class** `java.util.EventObject``getSource, toString`**Methods inherited from class** `java.lang.Object``clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructors

### ConnectingEvent

```
public ConnectingEvent(Object source,  
                        com.imis.imisarc.client.ConnectingEvent e)
```

Initializes a new instance of the ConnectingEvent class for session initialization stage.

**Parameters:**

source - the source of the event.

e - an IMiS/ARC Client connecting event arguments.

## Methods

### getStage

```
public ConnectingStage getStage()
```

Gets the connecting event stage.

**Returns:**

The connecting event stage.

---

### getProtocol

```
public ConnectionProtocol getProtocol()
```

Gets the session protocol.

**Returns:**

The session protocol.

---

### getRemoteHost

```
public String getRemoteHost()
```

Gets the remote host.

**Returns:**

The remote host.

---

### getRemoteSocketAddress

```
public java.net.SocketAddress getRemoteSocketAddress()
```

Gets the remote socket address.

**Returns:**

The SocketAddress of the server.

---

### getLocalSocketAddress

```
public java.net.SocketAddress getLocalSocketAddress()
```

(continued from last page)

Gets the local socket address.

The local socket address is a null reference only in unsecured session initialization stage when the session socket is not bound to the local socket address defined by the optional [StorageConnector.OPTION\\_OUTBOUND\\_ADDRESS](#) or [StorageConnector.OPTION\\_OUTBOUND\\_ADDRESS\\_V6](#) value.

**Returns:**

The SocketAddress of the client or a null reference.

---

## getSslProtocols

```
public String[] getSslProtocols()
```

Gets the names of the enabled SSL protocols used by the session.

**Returns:**

The array of strings representing enabled SSL protocols used by the session.

---

## setSslProtocols

```
public void setSslProtocols(String[] value)
```

Sets the names of the enabled SSL protocols used by the session.

**Parameters:**

value - the array of strings representing enabled SSL protocols used by the session.

**Throws:**

IllegalStateException - if used in connecting stage other than session initialization stage.

---

## getTrustAnchors

```
public Set getTrustAnchors()
```

Gets the trust anchors used for remote certificate verification.

**Returns:**

The set of TrustAnchor objects used in remote certificate verification stage.

---

## setTrustAnchors

```
public void setTrustAnchors(Set value)
```

Sets the trust anchors used for remote certificate verification that can be set in session initialization stage.

**Parameters:**

value - the set of TrustAnchor objects used for remote certificate verification.

**Throws:**

IllegalStateException - if used in connecting stage other than session initialization stage.

---

## getPasswordCallback

```
public javax.security.auth.callback.PasswordCallback getPasswordCallback()
```

Gets the password callback used to provide the password for JRE default trusted CA certificates key store that is used to retrieve the trust anchors for remote certificate validation if they are not provided with [setTrustAnchors\(Set\)](#) method. Password can be set in session initialization stage.

**Returns:**

(continued from last page)

The password callback or null if the key store is protected with default password.

**Throws:**

`IllegalStateException` - if used in connecting stage other than session initialization stage.

---

## getRemoteCertificateChain

```
public java.security.cert.X509Certificate[] getRemoteCertificateChain()
```

Gets the certificate chain used to authenticate the remote party.

**Returns:**

The certificate chain used to authenticate the remote party.

---

## getAuthenticationType

```
public String getAuthenticationType()
```

Gets the key exchange algorithm used to authenticate the remote party.

**Returns:**

The key exchange algorithm used to authenticate the remote party.

---

## isRemoteCertificateAccepted

```
public boolean isRemoteCertificateAccepted()
```

Gets the value indicating whether or not the specified remote certificate chain is accepted for authentication in the remote certificate validation stage.

**Returns:**

true if remote certificate is accepted for authentication; otherwise false.

---

## setRemoteCertificateAccepted

```
public void setRemoteCertificateAccepted(boolean value)
```

Sets the value indicating whether or not the specified remote certificate is accepted for authentication in the remote certificate validation stage.

**Parameters:**

value - true if remote certificate is accepted for authentication; otherwise false.

**Throws:**

`IllegalStateException` - if used in connecting stage other than remote certificate selection stage.

---

## getKeyTypes

```
public String[] getKeyTypes()
```

Gets the key algorithm type name(s), ordered with the most-preferred key type first, used when authenticating the client side.

**Returns:**

The key algorithm type name(s), ordered with the most-preferred key type first.

(continued from last page)

---

## getAcceptableIssuers

```
public java.security.Principal[] getAcceptableIssuers()
```

Gets the CA certificate issuers subject names acceptable to the remote party, used when authenticating the client side.

**Returns:**

An array of `Principal` objects representing CA certificate issuers subject names acceptable to the remote party.

---

## getLocalCertificate

```
public java.security.cert.X509Certificate getLocalCertificate()
```

Gets the local certificate used for establishing an SSL connection in the local certificate selection stage.

**Returns:**

The `X509Certificate` used for establishing an SSL connection.

---

## setLocalCertificate

```
public void setLocalCertificate(java.security.cert.X509Certificate value)
```

Sets the local certificate used for establishing an SSL connection in the local certificate selection stage.

**Parameters:**

`value` - the `X509Certificate` used for establishing an SSL connection.

**Throws:**

`IllegalStateException` - if used in connecting stage other than local certificate selection stage.

---

## getLocalCertificateKey

```
public java.security.PrivateKey getLocalCertificateKey()
```

Gets the local certificate private key used for establishing an SSL connection in the local certificate selection stage.

**Returns:**

The certificate `PrivateKey` used for establishing an SSL connection.

---

## setLocalCertificateKey

```
public void setLocalCertificateKey(java.security.PrivateKey value)
```

Sets the local certificate private key used for establishing an SSL connection in the local certificate selection stage.

**Parameters:**

`value` - the certificate `PrivateKey` used for establishing an SSL connection.

**Throws:**

`IllegalStateException` - if used in connecting stage other than local certificate selection stage.

---

## cancel

```
public boolean cancel()
```

Gets the value indicating whether or not to cancel connecting.

**Returns:**

(continued from last page)

true to cancel connecting; otherwise false. Default value is false.

---

## **setCancel**

```
public void setCancel(boolean value)
```

Sets the value indicating whether or not to cancel connecting.

### **Parameters:**

value - true to cancel connecting; otherwise false.

## com.imis.storageconnector

### Class ConnectingStage

```

java.lang.Object
  |
  +- java.lang.Enum
        +- com.imis.storageconnector.ConnectingStage
  
```

**All Implemented Interfaces:**  
 Serializable, Comparable

```

public final class ConnectingStage
extends Enum
  
```

Defines stages of the connecting event.

#### Field Summary

public static final	<a href="#">LOCAL_CERTIFICATE_SELECTION</a> Local certificate selection stage, where local certificate is provided.
public static final	<a href="#">REMOTE_CERTIFICATE_VALIDATION</a> Remote certificate validation stage, where remote certificate is accepted or not.
public static final	<a href="#">SESSION_INITIALIZATION</a> Session initialization stage, where enabled SSL protocols, local certificates store preference and/or a collection of local certificates are provided.

#### Method Summary

static <a href="#">ConnectingStage</a>	<a href="#">valueOf</a> (String name)
static <a href="#">ConnectingStage[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

#### Fields



(continued from last page)

---

## SESSION\_INITIALIZATION

```
public static final com.imis.storageconnector.ConnectingStage SESSION_INITIALIZATION
```

Session initialization stage, where enabled SSL protocols, local certificates store preference and/or a collection of local certificates are provided.

---

## REMOTE\_CERTIFICATE\_VALIDATION

```
public static final com.imis.storageconnector.ConnectingStage  
REMOTE_CERTIFICATE_VALIDATION
```

Remote certificate validation stage, where remote certificate is accepted or not.

---

## LOCAL\_CERTIFICATE\_SELECTION

```
public static final com.imis.storageconnector.ConnectingStage  
LOCAL_CERTIFICATE_SELECTION
```

Local certificate selection stage, where local certificate is provided.

## Methods

### values

```
public final static ConnectingStage\[\] values()
```

---

### valueOf

```
public static ConnectingStage valueOf(String name)
```

# com.imis.storageconnector

## Class ConnectionProtocol



All Implemented Interfaces:  
Serializable, Comparable

public final class **ConnectionProtocol**  
extends Enum

Represents the connection protocol.

Field Summary

public static final	<a href="#">SECURED</a> Secured.
public static final	<a href="#">UNKNOWN</a> Unknown.
public static final	<a href="#">UNSECURED</a> Unsecured.

Method Summary

static <a href="#">ConnectionProtocol</a>	<a href="#">valueOf</a> (String name)
static <a href="#">ConnectionProtocol[]</a>	<a href="#">values</a> ()

Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface java.lang.Comparable

compareTo

Fields

(continued from last page)

# UNKNOWN

public static final com.imis.storageconnector.ConnectionProtocol **UNKNOWN**

Unknown.

# UNSECURED

public static final com.imis.storageconnector.ConnectionProtocol **UNSECURED**

Unsecured.

# SECURED

public static final com.imis.storageconnector.ConnectionProtocol **SECURED**

Secured.

## Methods

### values

public final static [ConnectionProtocol\[\]](#) **values()**

### valueOf

public static [ConnectionProtocol](#) **valueOf**(String name)

## com.imis.storageconnector Class ContentTypeMapper

java.lang.Object

└─com.imis.storageconnector.ContentTypeMapper

```
public class ContentTypeMapper
    extends Object
```

Content type mapper from/to legacy content types.

### Constructor Summary

public	<a href="#">ContentTypeMapper()</a>
--------	-------------------------------------

### Method Summary

static String	<a href="#">fromLegacyContentType</a> (String contentType) Gets the content type from the input legacy content type.
static String	<a href="#">toLegacyContentType</a> (String contentType) Gets the legacy content type from the input content type.

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### ContentTypeMapper

```
public ContentTypeMapper()
```

## Methods

### fromLegacyContentType

```
public static String fromLegacyContentType(String contentType)
```

Gets the content type from the input legacy content type.

#### Parameters:

contentType - the input legacy content type (case sensitive).

#### Returns:

The legacy content type.

(continued from last page)

## toLegacyContentType

```
public static String toLegacyContentType(String contentType)
```

Gets the legacy content type from the input content type.

**Parameters:**

contentType - the input content type (case sensitive).

**Returns:**

The legacy content type.

## com.imis.storageconnector Class Document

java.lang.Object

└─com.imis.storageconnector.Document

### All Implemented Interfaces:

com.imis.IAutoCloseable

public abstract class **Document**  
extends Object  
implements com.imis.IAutoCloseable

Abstract base class for documents in storages supported by IMiS/Storage Connector.

**Note:** Only IMIS\_ARC storage document is currently implemented by IMiS/Storage Connector. Use one `Storage.createObject` or `Storage.openObject` to get an instance of a document.

### See Also:

[Storage.createObject\(String, String\)](#), [Storage.openObject\(String, DocumentAccess\)](#)

## Constructor Summary

protected	<a href="#">Document</a> ( <a href="#">Storage</a> parent, <a href="#">DocumentAccess</a> mode) Initializes a new instance of the Document class.
-----------	--

## Method Summary

abstract <a href="#">Document</a>	<a href="#">clone</a> ( <a href="#">DocumentAccess</a> mode) Creates a copy of a document.
abstract void	<a href="#">close</a> () Closes the document.
abstract void	<a href="#">delete</a> () Deletes the document.
<a href="#">DocumentAccess</a>	<a href="#">getAccessMode</a> () Gets a document access mode.
boolean	<a href="#">getAutoSave</a> () Gets a value indicating whether the document autosave is enabled.
abstract Date	<a href="#">getCreated</a> () Gets the creation date and time of the document.
String	<a href="#">getDefaultExtension</a> () Gets the default extension for the document.
String	<a href="#">getDefaultMime</a> () Gets the default MIME type for the document.
String[]	<a href="#">getExtensions</a> () Gets an array of available extensions for the document MIME type.

abstract String	<a href="#"><u>getId()</u></a> Gets the document identifier.
abstract InputStream	<a href="#"><u>getInputStream()</u></a> Gets the document data input stream.
abstract Date	<a href="#"><u>getLastAccessed()</u></a> Gets the date and time the document was last accessed.
String[]	<a href="#"><u>getMimes()</u></a> Gets an array of available MIME types for the document extension.
abstract Date	<a href="#"><u>getModified()</u></a> Gets the date and time the document was modified.
abstract OutputStream	<a href="#"><u>getOutputStream()</u></a> Gets the document data output stream.
abstract long	<a href="#"><u>getSize()</u></a> Gets the size of the document.
<a href="#"><u>Storage</u></a>	<a href="#"><u>getStore()</u></a> Gets the document storage.
boolean	<a href="#"><u>isClosed()</u></a> Gets a value indicating whether the document is closed.
boolean	<a href="#"><u>isModified()</u></a> Gets a value indicating whether the document is modified.
boolean	<a href="#"><u>isNew()</u></a> Gets a value indicating whether the document is new.
abstract void	<a href="#"><u>move()</u></a> (String profile) Moves the document to the storage profile.
abstract void	<a href="#"><u>save()</u></a> Saves the document.
void	<a href="#"><u>setAutoSave()</u></a> (boolean value) Sets a value indicating whether the document autosave is enabled.
void	<a href="#"><u>setExtensions()</u></a> (String[] extensions) Sets an array of available extensions for the document MIME type.
void	<a href="#"><u>setIsClosed()</u></a> (boolean value) Sets a value indicating whether the document is closed.
void	<a href="#"><u>setIsModified()</u></a> (boolean value) Sets a value indicating whether the document is modified.
void	<a href="#"><u>setIsNew()</u></a> (boolean value) Sets a value indicating whether the document is new.
void	<a href="#"><u>setMimes()</u></a> (String[] mimes) Sets an array of available MIME types for the document extension.

**Methods inherited from class** `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Methods inherited from interface `com.imis.IAutoCloseable`

```
close
```

## Constructors

### Document

```
protected Document(Storage parent,  
                   DocumentAccess mode)
```

Initializes a new instance of the `Document` class.

**Parameters:**

`parent` - a document storage.  
`mode` - document access mode.

**Throws:**

`NullPointerException` - if `parent` is a null reference.  
[StorageConnectorException](#) - if demo license expired.

## Methods

### getDefaultExtension

```
public String getDefaultExtension()
```

Gets the default extension for the document.

**Returns:**

A string containing the default document extension or a null reference if none available.

### getExtensions

```
public String[] getExtensions()
```

Gets an array of available extensions for the document MIME type.

**Returns:**

An array of extensions for the document MIME type.

### setExtensions

```
protected void setExtensions(String[] extensions)
```

Sets an array of available extensions for the document MIME type.

**Parameters:**

`extensions` - an array of extensions for the document MIME type.



(continued from last page)

---

## getDefaultMime

```
public String getDefaultMime()
```

Gets the default MIME type for the document.

**Returns:**

A string containing default document MIME type or a null reference if none available.

---

## getMimes

```
public String[] getMimes()
```

Gets an array of available MIME types for the document extension.

**Returns:**

An array of MIME types for the document extension.

---

## setMimes

```
protected void setMimes(String[] mimes)
```

Sets an array of available MIME types for the document extension.

**Parameters:**

mimes - an array of MIME types for the document extension.

---

## getAccessMode

```
public DocumentAccess getAccessMode()
```

Gets a document access mode.

**Returns:**

A document access mode.

---

## getStore

```
public Storage getStore()
```

Gets the document storage.

**Returns:**

A Storage instance.

---

## isClosed

```
public boolean isClosed()
```

Gets a value indicating whether the document is closed.

**Returns:**

true if the document is closed; otherwise, false.

---

## setIsClosed

```
protected void setIsClosed(boolean value)
```

(continued from last page)

Sets a value indicating whether the document is closed.

**Parameters:**

value - true if the document is closed; otherwise, false.

---

## isNew

```
public boolean isNew()
```

Gets a value indicating whether the document is new.

**Returns:**

true if the document is new; otherwise, false.

---

## setIsNew

```
protected void setIsNew(boolean value)
```

Sets a value indicating whether the document is new.

**Parameters:**

value - true if the document is new; otherwise, false.

---

## isModified

```
public boolean isModified()
```

Gets a value indicating whether the document is modified.

**Returns:**

true if the document is modified; otherwise, false.

---

## setIsModified

```
protected void setIsModified(boolean value)
```

Sets a value indicating whether the document is modified.

**Parameters:**

value - true if the document is modified; otherwise, false.

---

## getAutoSave

```
public boolean getAutoSave()
```

Gets a value indicating whether the document autosave is enabled.

**Returns:**

true if the document autosave is enabled; otherwise, false.

---

## setAutoSave

```
public void setAutoSave(boolean value)
```

Sets a value indicating whether the document autosave is enabled.

**Parameters:**

value - true to enable document autosave; otherwise, false.

---

## getId

```
public abstract String getId()  
    throws StorageConnectorException
```

Gets the document identifier.

**Returns:**

A string that represents the document identifier.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
IMIS\_ARC storage error:

- Object ID is not valid until new object is saved.

---

## getSize

```
public abstract long getSize()  
    throws StorageConnectorException,  
           IOException
```

Gets the size of the document.

**Returns:**

The size of the document, in bytes.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
IMIS\_ARC storage error:

- Operation on closed object is not allowed.

IOException - if an I/O error occurs.

---

## getCreated

```
public abstract Date getCreated()  
    throws StorageConnectorException,  
           IOException
```

Gets the creation date and time of the document.

**Note:** This feature is currently not supported on IMIS\_ARC storage.

**Returns:**

The Date of document creation.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
IOException - if an I/O error occurs.

---

## getModified

```
public abstract Date getModified()  
    throws StorageConnectorException,  
           IOException
```

(continued from last page)

Gets the date and time the document was modified.

**Note:** This feature is currently not supported on IMIS\_ARC storage.

**Returns:**

The Date the document was modified.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

[IOException](#) - if an I/O error occurs.

---

## getLastAccessed

```
public abstract Date getLastAccessed()  
    throws StorageConnectorException,  
           IOException
```

Gets the date and time the document was last accessed.

**Note:** This feature is currently not supported on IMIS\_ARC storage.

**Returns:**

The Date the document was last accessed.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

[IOException](#) - if an I/O error occurs.

---

## getInputStream

```
public abstract InputStream getInputStream()  
    throws StorageConnectorException,  
           IOException
```

Gets the document data input stream.

**Returns:**

The document data input stream.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
IMIS\_ARC storage errors:

- Operation on closed object is not allowed.
- Error getting object data input stream.

[IOException](#) - if an I/O error occurs.

---

## getOutputStream

```
public abstract OutputStream getOutputStream()  
    throws StorageConnectorException,  
           IOException
```

Gets the document data output stream.

**Returns:**

The document data output stream.

(continued from last page)

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
IMIS\_ARC storage errors:

- Operation on closed object is not allowed.
- Error getting object data output stream.

IOException - if an I/O error occurs.

---

## clone

```
public abstract Document clone(DocumentAccess mode)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Creates a copy of a document.

Use [isClosed\(\)](#) property to determine whether the current document is closed.

Use [isNew\(\)](#) property to determine whether the current document is new.

**Note:** This feature is currently not supported on IMIS\_ARC storage.

**Parameters:**

mode - a document access mode.

**Returns:**

A Document instance.

**Throws:**

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IOException - if an I/O error occurs.

---

## move

```
public abstract void move(String profile)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Moves the document to the storage profile.

Use [isClosed\(\)](#) property to determine whether the current document is closed.

Use [isNew\(\)](#) property to determine whether the current document is new.

**Parameters:**

profile - a name of the profile on the storage.

**Throws:**

NullPointerException - if profile is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
IMIS\_ARC storage errors:

- Operation on closed object is not allowed.
- New objects cannot be moved until they are saved and closed.
- Error occurred while saving object.
- Error occurred while closing object.
- Error occurred while moving object.
- Error occurred while opening session.
- Error getting session from the session pool.
- Error occurred while opening object.

IOException - if an I/O error occurs.

---

## delete

```
public abstract void delete()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Deletes the document.

Use [isClosed\(\)](#) property to determine whether the current document is closed.

After this operation completes, the document is closed.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
IMIS\_ARC storage errors:

- Operation on closed object is not allowed.
- Error occurred while removing object.
- Error getting session from the session pool.
- Error occurred while saving object.
- Error occurred while closing object.

IOException - if an I/O error occurs.

---

## save

```
public abstract void save()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Saves the document.

Use [isClosed\(\)](#) property to determine whether the current document is closed.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
IMIS\_ARC storage errors:

- Operation on closed object is not allowed.
- Error occurred while committing new object or object changes.

IOException - if an I/O error occurs.

---

## close

```
public abstract void close()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Closes the document.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
IMIS\_ARC storage errors:

- Error occurred while saving object.
- Error occurred while closing object.

IOException - if an I/O error occurs.

## com.imis.storageconnector Class DocumentAccess

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.DocumentAccess
  
```

**All Implemented Interfaces:**  
Serializable, Comparable

public final class **DocumentAccess**  
extends Enum

Defines constants for read, write, or read/write access to a document on the storage supported by IMiS/Storage Connector.

### Field Summary

public static final	<a href="#">READ</a> Read access to the document.
public static final	<a href="#">READ_WRITE</a> Read and write access to the document.
public static final	<a href="#">WRITE</a> Write access to the document.

### Method Summary

static <a href="#">DocumentAccess</a>	<a href="#">valueOf</a> (String name)
static <a href="#">DocumentAccess[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields



(continued from last page)

## READ

```
public static final com.imis.storageconnector.DocumentAccess READ
```

Read access to the document. Data can be read from the document.

---

## WRITE

```
public static final com.imis.storageconnector.DocumentAccess WRITE
```

Write access to the document. Data can be written to the document.

---

## READ\_WRITE

```
public static final com.imis.storageconnector.DocumentAccess READ_WRITE
```

Read and write access to the document. Data can be written to and read from the document.

## Methods

### values

```
public final static DocumentAccess\[\] values()
```

---

### valueOf

```
public static DocumentAccess valueOf(String name)
```

## com.imis.storageconnector Class IACSession

java.lang.Object

```

  |
  +--com.imis.imisarc.client.IASession
      |
      +--com.imis.storageconnector.IACSession
  
```

### All Implemented Interfaces:

com.imis.IAutoCloseable

```

public class IACSession
extends com.imis.imisarc.client.IASession
  
```

Represents the IMiS/ARC Client session that is handled by a user context session pool.

### Fields inherited from class com.imis.imisarc.client.IASession

MAX\_DATETIME, MIN\_DATETIME, PROPERTY\_AUDIT\_LOG\_REQUIRED, PROPERTY\_AUDIT\_LOG\_SUPPORTED, PROPERTY\_AUTHENTICATED\_USER\_ACCOUNT, PROPERTY\_CAPABILITIES, PROPERTY\_CLASS\_HIERARCHY\_DELIMITER, PROPERTY\_DOCUMENT\_HIERARCHY\_DELIMITER, PROPERTY\_FOLDER\_HIERARCHY\_DELIMITER, PROPERTY\_ROOT\_EFFECTIVE\_RIGHTS, PROPERTY\_SERVICE\_NAME, PROPERTY\_SERVICE\_VERSION, PROPERTY\_TRAFFIC\_ENCRYPTED

## Constructor Summary

public	<a href="#">IASession()</a> Initializes a new instance of the <a href="#">IASession</a> class.
public	<a href="#">IASession(IASessionPool owner)</a> Initializes a new instance of the <a href="#">IASession</a> class.

## Method Summary

int	<a href="#">acquire()</a> Acquires the reference to the session.
void	<a href="#">close()</a> Closes a session to IMiS/ARC Storage Server.
com.imis.imisarc.client.IASessionContext	<a href="#">getContext()</a> Gets the IMiS/ARC Client authenticated session context.
int	<a href="#">getId()</a> Gets the unique session identifier.
<a href="#">IASessionPool</a>	<a href="#">getOwner()</a> Gets the owner session pool.
Date	<a href="#">getTimeStamp()</a> Gets the time of the session last usage.
void	<a href="#">lock()</a> Locks the session by acquiring this session from the parent session pool.

int	<a href="#">release()</a> Releases the reference to the session.
void	<a href="#">setOwner</a> ( <a href="#">IASessionPool</a> value) Sets the owner session pool.
void	<a href="#">setTimeStamp</a> (Date value) Gets the time of the session last usage.
void	<a href="#">unlock()</a> Unlocks this session by releasing this session from the parent session pool.

#### Methods inherited from class `com.imis.imisarc.client.IACSession`

addAuthenticatedListener, addAuthenticatingListener, addConnectingListener, auditLogQuery, auditLogQuery, authenticate, close, createBinaryValue, createFileValue, createFileValue, createStringMaxValue, directoryGetGroupMembers, directoryGetMembers, directoryGetUserProfile, entityCheckIn, entityCheckOut, entityCreate, entityCreate, entityGetInfo, entityGetInfo, entityMove, entityOpen, entityOpen, entityRemove, entityRemove, entitySetSecurityClass, entitySetStatus, finalize, getArchives, getAuditLog, getAuthenticationData, getProperties, getPublicClassificationCode, getPublicClassificationCode, getReport, getSubEntities, getTemplates, logAddHandler, logAddHandlers, logGetHandlers, logGetLevel, logRemoveHandler, logRemoveHandlers, logSetLevel, objectCreate, objectCreate, objectMove, objectMove, objectOpen, objectOpen, objectRemove, objectRemove, objectResolveExternalId, objectResolveExternalId, objectSetExternalId, objectTransfer, open, open, open, open, open, removeAuthenticatedListener, removeAuthenticatingListener, removeConnectingListener, reopen, search, setAuthenticationData

#### Methods inherited from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface `com.imis.IAutoCloseable`

close

## Constructors

### IACSession

```
public IACSession()
```

Initializes a new instance of the [IACSession](#) class.

This constructor creates anonymous session.

#### Throws:

`SecurityException` - if the calling thread does not have permission to create an internal class loader.

### IACSession

```
public IACSession(IASessionPool owner)
```

(continued from last page)

Initializes a new instance of the [IACTSession](#) class.

This constructor creates session with unique identifier.

**Parameters:**

owner - an owner session pool.

**Throws:**

SecurityException - if the calling thread does not have permission to create an internal class loader.

## Methods

### getOwner

```
public IACTSessionPool getOwner()
```

Gets the owner session pool.

**Returns:**

The owner session pool.

---

### setOwner

```
public void setOwner(IACTSessionPool value)
```

Sets the owner session pool.

**Parameters:**

value - the owner session pool.

---

### getContext

```
public com.imis.imisarc.client.IASessionContext getContext()
```

Gets the IMiS/ARC Client authenticated session context.

**Returns:**

The IMiS/ARC Client authenticated session context or a null reference if session is not authenticated.

---

### getId

```
public int getId()
```

Gets the unique session identifier.

**Returns:**

The unique session identifier or -1 for anonymous session.

---

### getTimeStamp

```
public Date getTimeStamp()
```

Gets the time of the session last usage.

By design, the last usage is updated before the session is returned to the session pool.

**Returns:**

The time of the session last usage.

---

## setTimeStamp

```
public void setTimeStamp(Date value)
```

Gets the time of the session last usage.

By design, the last usage is updated before the session is returned to the session pool.

### Parameters:

value - the time of the session last usage.

---

## acquire

```
public int acquire()
```

Acquires the reference to the session.

This method increments and returns the session reference count.

### Returns:

The session reference count.

---

## release

```
public int release()
```

Releases the reference to the session.

This method decrements and returns the session reference count.

### Returns:

The session reference count.

---

## lock

```
public void lock()
```

Locks the session by acquiring this session from the parent session pool.

Use [unlock\(\)](#) method to unlock the session. This method does nothing on anonymous session.

### See Also:

[IASessionPool.acquire\(\)](#)

---

## unlock

```
public void unlock()
```

Unlocks this session by releasing this session from the parent session pool.

Use [lock\(\)](#) method to lock the session. This method does nothing on anonymous session.

### See Also:

[IASessionPool.release\(\)](#)

---

## close

```
public void close()
```

---

(continued from last page)

Closes a session to IMiS/ARC Storage Server.

## com.imis.storageconnector Interface IArchive

public interface **IArchive**  
extends com.imis.IAutoCloseable

Prescribes a way with which a client can communicate with an IMiS/Storage Connector compatible archive hosted on an archiving system.

IArchive interface provides methods and properties to access different archive services and functions such as directory services, entity information and lifecycle framework, etc.

**Note:** Only IMiS/ARChive Storage Server archive is currently supported by IMiS/Storage Connector. Use one of the `StorageConnector.openArchive` methods to access IMIS\_ARCHIVE archive.

**See Also:**

[StorageConnector.openArchive\(ArchiveType, String, Map\)](#),  
[StorageConnector.openArchive\(ArchiveType, String, int, Map\)](#)

### Method Summary

void	<a href="#">addAuthenticatingListener</a> ( <a href="#">IAuthenticatingListener</a> listener) Adds an <a href="#">IAuthenticatingListener</a> object.
void	<a href="#">addConnectingListener</a> ( <a href="#">IConnectingListener</a> listener) Adds an <a href="#">IConnectingListener</a> object.
InputStream	<a href="#">auditLogQuery</a> ( <a href="#">AuditQuery</a> parameters) Gets the audit log query results data input stream.
void	<a href="#">authenticate</a> () Authenticates user on archive.
<a href="#">IEntity</a>	<a href="#">changeEntityTemplate</a> ( <a href="#">EntityIdKind</a> idKind, String id, String templateId, <a href="#">IEntity</a> syncEntity) Opens a generic <a href="#">IEntity</a> from the archive given by the entity identifier in read-write mode and changes its template to the specified value.
String	<a href="#">checkInDraft</a> (String id, Short position, String comment) Checks in an entity draft.
String	<a href="#">checkOutEntity</a> ( <a href="#">EntityIdKind</a> idKind, String id, Integer sequence) Checks out an <a href="#">IEntity</a> .
void	<a href="#">close</a> () Closes all active sessions to an archive.
<a href="#">IClass</a>	<a href="#">createClass</a> (String templateId) Creates a new <a href="#">IClass</a> entity on the first level of entity hierarchy based on the template identifier which defines its metadata scheme and default ACL state.
<a href="#">IClass</a>	<a href="#">createClass</a> (String templateId, <a href="#">EntityIdKind</a> parentIdKind, String parentId) Creates a new <a href="#">IClass</a> or sub-class entity instance.
<a href="#">IDocument</a>	<a href="#">createDocument</a> (String templateId, <a href="#">EntityIdKind</a> parentIdKind, String parentId) Creates a new <a href="#">IDocument</a> entity instance.

<a href="#">IEntity</a>	<a href="#">createEntity</a> (String templateId, <a href="#">EntityIdKind</a> parentIdKind, String parentId) Creates a new IEntity or sub-entity entity instance.
<a href="#">IEntityReference</a>	<a href="#">createEntityReference</a> () Creates a new IEntityReference.
<a href="#">IFolder</a>	<a href="#">createFolder</a> (String templateId, <a href="#">EntityIdKind</a> parentIdKind, String parentId) Creates a new IFolder or sub-folder entity instance.
<a href="#">IReview</a>	<a href="#">createReview</a> ( <a href="#">IEntityStub</a> scope, Collection policies) Creates the review given the list of retention policies and scope entity.
<a href="#">IReview</a>	<a href="#">createReview</a> ( <a href="#">IEntityStub</a> scope, String queryExpression) Creates the review given the query expression and scope entity.
void	<a href="#">deleteEntity</a> ( <a href="#">EntityIdKind</a> idKind, String id) Deletes the IEntity and associated resources.
void	<a href="#">deleteEntity</a> ( <a href="#">EntityIdKind</a> idKind, String id, String description, Integer sequence) Deletes the IEntity and associated resources.
void	<a href="#">discardDraft</a> (String id) Discards an entity draft.
<a href="#">AuditLog</a>	<a href="#">getAuditLog</a> () Gets the archive AuditLog object.
<a href="#">IEntityCollection</a>	<a href="#">getDeletedEntities</a> ( <a href="#">DeletionKind</a> kind, Collection sortKeys) Returns the read-only collection of deleted entities using the specified sort keys and a collection of property names used for categorization.
<a href="#">IDeletionStub</a>	<a href="#">getDeletedEntity</a> ( <a href="#">EntityIdKind</a> idKind, String id) Queries archiving server for deleted entity public properties.
<a href="#">IDirectory</a>	<a href="#">getDirectory</a> () Gets the archive IDirectory.
com.imis.util.IReadOnlyMap	<a href="#">getDispositionHolds</a> () Gets the collection of disposition holds sorted by their internal identifiers.
<a href="#">IDraftStub</a>	<a href="#">getDraftInfo</a> (String id) Queries archiving server for draft public properties (metadata attributes) and other specifics which can be accessed without opening a draft.
<a href="#">IEntityCollection</a>	<a href="#">getDrafts</a> (Collection sortKeys, Collection categorizationPropertyNames) Returns the read-only collection of entity drafts using the specified sort keys and a collection of property names used for categorization.
<a href="#">IReadOnlyEntityRights</a>	<a href="#">getEffectiveRights</a> ( <a href="#">EntityIdKind</a> idKind, String id, <a href="#">IDirectoryEntity</a> subject) Gets the users effective rights.
int	<a href="#">getEntityCollectionPageCount</a> () Gets the entity collection page count.
int	<a href="#">getEntityCollectionPageSize</a> () Gets the entity collection page size.



Map	<a href="#">getEntityInfo</a> ( <a href="#">EntityIdKind</a> idKind, Collection ids) Queries archiving server for entities public properties (metadata attributes) and other specifics which can be accessed without opening entities.
Map	<a href="#">getEntityInfo</a> ( <a href="#">EntityIdKind</a> idKind, Collection ids, Set scope) Queries archiving server in the context of the specified user for the entities public properties (metadata attributes) and other specifics which can be accessed without opening a collection of entities and are specified by entity query scope.
<a href="#">IEntityStub</a>	<a href="#">getEntityInfo</a> ( <a href="#">EntityIdKind</a> idKind, String id) Queries archiving server for entity public properties (metadata attributes) and other specifics which can be accessed without opening an entity.
<a href="#">IEntityStub</a>	<a href="#">getEntityInfo</a> ( <a href="#">EntityIdKind</a> idKind, String id, Set scope) Queries archiving server in the context of the specified user for entity public properties (metadata attributes) and other specifics which can be accessed without opening an entity and are specified by entity query scope.
<a href="#">IReadOnlyEntityReference</a>	<a href="#">getEntityReference</a> (String referenceId) Gets the entity reference using the specified entity reference identifier.
String	<a href="#">getHost</a> () Gets the archive information as host name or IP address and port separated by colon (:).
String	<a href="#">getInternalId</a> (String externalId) Resolves the specified external identifier to the entity identifier that external identifier is associated with.
<a href="#">IEntityCollection</a>	<a href="#">getLogEntities</a> ( <a href="#">LogType</a> type, Collection sortKeys, Collection categorizationPropertyNames) Returns the read-only collection of logged entities using the specified sort keys and a collection of property names used for categorization.
com.imis.util.IReadOnlyMap	<a href="#">getProperties</a> () Gets the read-only map of archive properties.
String	<a href="#">getPublicClassificationCode</a> (String classificationCode) Converts the specified classification code in canonical format to the public classification code.
InputStream	<a href="#">getReport</a> ( <a href="#">ReportType</a> type) Gets the audit log query results data input stream.
com.imis.util.IReadOnlyMap	<a href="#">getRetentionPolicies</a> () Gets the collection of retention policies sorted by their internal identifiers.
<a href="#">IReviewStub</a>	<a href="#">getReviewInfo</a> ( <a href="#">EntityIdKind</a> idKind, String id) Gets the review document info.
<a href="#">IEntityCollection</a>	<a href="#">getReviews</a> (Collection sortKeys, Collection categorizationPropertyNames) Gets the read-only collection of review folders using the specified sort keys and a collection of property names used for categorization.
<a href="#">IEntityCollection</a>	<a href="#">getRootClasses</a> (Collection sortKeys, Collection categorizationPropertyNames) Returns the read-only collection the first level archive classes using the specified sort keys and a collection of property names used for categorization.
<a href="#">IEntityCollection</a>	<a href="#">getSubEntities</a> ( <a href="#">EntityIdKind</a> idKind, String id, <a href="#">EntityFilter</a> filter, Collection sortKeys, Collection categorizationPropertyNames) Gets the read-only collection of child entity stubs using the specified entity filter, sort keys and a collection of property names used for categorization.

String	<a href="#">getSystemPropertyName</a> ( <a href="#">SystemProperty</a> systemProperty) Gets the system property name.
<a href="#">EntitySortKey</a>	<a href="#">getSystemPropertySortKey</a> ( <a href="#">SystemProperty</a> systemProperty, <a href="#">EntitySortKeyDirection</a> direction) Gets the entity sort key for sorting entity collection by the specified system property.
com.imis.util.IReadOn lyMap	<a href="#">getTemplates</a> () Gets the collection of entity templates registered on the archive.
String	<a href="#">getUser</a> () Gets the user of the archive.
boolean	<a href="#">isClosed</a> () Gets a value indicating whether the archive is closed.
void	<a href="#">moveEntity</a> ( <a href="#">EntityIdKind</a> idKind, String id, <a href="#">EntityIdKind</a> parentIdKind, String parentId) Moves an IEntity with the specified entity identifier under the specified parent entity.
<a href="#">IClass</a>	<a href="#">openClass</a> ( <a href="#">EntityIdKind</a> idKind, String id, <a href="#">AccessMode</a> mode) Opens a IClass entity from the archive given by the class identifier with the specified access mode.
<a href="#">IDocument</a>	<a href="#">openDocument</a> ( <a href="#">EntityIdKind</a> idKind, String id, Integer sequence, <a href="#">AccessMode</a> mode) Opens a IDocument entity from the archive given by the document identifier with the specified access mode.
<a href="#">IDraft</a>	<a href="#">openDraft</a> (String id, <a href="#">AccessMode</a> mode) Opens a IDraft from the archive given by the draft identifier with the specified access mode.
<a href="#">IEntity</a>	<a href="#">openEntity</a> ( <a href="#">EntityIdKind</a> idKind, String id, Integer sequence, <a href="#">AccessMode</a> mode, <a href="#">IEntity</a> syncEntity) Opens a generic IEntity from the archive given by the entity identifier with the specified access mode.
<a href="#">IFolder</a>	<a href="#">openFolder</a> ( <a href="#">EntityIdKind</a> idKind, String id, <a href="#">AccessMode</a> mode) Opens a IFolder entity from the archive given by the folder identifier with the specified access mode.
<a href="#">IReview</a>	<a href="#">openReview</a> ( <a href="#">EntityIdKind</a> idKind, String id, <a href="#">AccessMode</a> mode) Opens the review.
void	<a href="#">removeAuthenticatingListener</a> ( <a href="#">IAuthenticatingListener</a> listener) Removes an <a href="#">IAuthenticatingListener</a> object.
void	<a href="#">removeConnectingListener</a> ( <a href="#">IConnectingListener</a> listener) Removes an <a href="#">IConnectingListener</a> object.
<a href="#">IEntityCollection</a>	<a href="#">search</a> ( <a href="#">EntityFilter</a> filter, Collection sortKeys, Set options, String searchExpression, Collection categorizationPropertyNames, Long maxElements) Searches for child entity stubs of the archive using the specified entity search filter, sort keys, search options, expression string and a collection of property names used for categorization.

<a href="#">IEntityCollection</a>	<a href="#">search</a> ( <a href="#">EntityIdKind</a> idKind, String id, <a href="#">EntityFilter</a> filter, Collection sortKeys, Set options, String searchExpression, Collection categorizationPropertyNames, Long maxElements)  Searches for child entity stubs of the archive using the specified entity search filter, sort keys, search options, expression string and a collection of property names used for categorization.
void	<a href="#">setEntityCollectionPageCount</a> (int value) Sets the entity collection page count.
void	<a href="#">setEntityCollectionPageSize</a> (int value) Sets the entity collection page size.
void	<a href="#">setEntitySecurityClass</a> ( <a href="#">EntityIdKind</a> idKind, String id, String securityClass) Sets the security class of the entity given by the entity identifier.
void	<a href="#">setEntityStatus</a> ( <a href="#">EntityIdKind</a> idKind, String id, String status) Sets the status of the entity given by the entity identifier.

Methods inherited from interface `com.imis.IAutoCloseable`

`close`

## Methods

### addConnectingListener

```
public void addConnectingListener(IConnectingListener listener)
```

Adds an [IConnectingListener](#) object.

**Parameters:**

listener - an `IConnectingListener`.

### removeConnectingListener

```
public void removeConnectingListener(IConnectingListener listener)
```

Removes an [IConnectingListener](#) object.

**Parameters:**

listener - an `IConnectingListener`.

### addAuthenticatingListener

```
public void addAuthenticatingListener(IAuthenticatingListener listener)
```

Adds an [IAuthenticatingListener](#) object.

**Parameters:**

listener - an `IAuthenticatingListener`.

### removeAuthenticatingListener

```
public void removeAuthenticatingListener(IAuthenticatingListener listener)
```

(continued from last page)

Removes an [IAuthenticatingListener](#) object.

**Parameters:**

listener - an IAuthenticatingListener.

---

## isClosed

```
public boolean isClosed()
```

Gets a value indicating whether the archive is closed.

**Returns:**

true if the archive is closed; otherwise false.

---

## getHost

```
public String getHost()
```

Gets the archive information as host name or IP address and port separated by colon (:).

**Returns:**

An archive host name or IP address and port separated by colon (:).

---

## getUser

```
public String getUser()
```

Gets the user of the archive.

**Returns:**

The user of the archive.

---

## getProperties

```
public com.imis.util.IReadOnlyMap getProperties()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the read-only map of archive properties.

**Returns:**

The read-only map of archive properties.

**Throws:**

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting archive properties.

IOException - if an I/O error occurs.

(continued from last page)

---

## getDirectory

```
public IDirectory getDirectory()  
    throws StorageConnectorException
```

Gets the archive IDirectory.

This property gives the caller access to the archive IDirectory services.

**Returns:**

A IDirectory instance.

---

## getAuditLog

```
public AuditLog getAuditLog()
```

Gets the archive AuditLog object.

This property provides access to the AuditLog interface which provides a facility for notifying archiving system about the cause for operations on entities executed from this IArchive instance.

**Returns:**

An AuditLog instance.

---

## getEntityCollectionPageSize

```
public int getEntityCollectionPageSize()
```

Gets the entity collection page size.

**Returns:**

The entity collection page size.

---

## setEntityCollectionPageSize

```
public void setEntityCollectionPageSize(int value)
```

Sets the entity collection page size.

**Parameters:**

value - the entity collection page size.

**Throws:**

IllegalArgumentException - if value is less than or equal to 0.

---

## getEntityCollectionPageCount

```
public int getEntityCollectionPageCount()
```

Gets the entity collection page count.

**Returns:**

The entity collection page count.

---

## setEntityCollectionPageCount

```
public void setEntityCollectionPageCount(int value)
```

Sets the entity collection page count.

---

(continued from last page)

**Parameters:**

value - the entity collection page count.

**Throws:**

`IllegalArgumentException` - if value is less than or equal to 0.

---

## getTemplates

```
public com.imis.util.IReadOnlyMap getTemplates()  
    throws StorageConnectorException
```

Gets the collection of entity templates registered on the archive.

**Returns:**

A read-only map of `ITemplate` objects sorted by their unique identifiers.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting templates on archive.

---

## getRetentionPolicies

```
public com.imis.util.IReadOnlyMap getRetentionPolicies()  
    throws StorageConnectorException
```

Gets the collection of retention policies sorted by their internal identifiers.

**Returns:**

A read-only map of `IRetentionPolicy` objects sorted by their identifiers.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Error occurred trying to access `Policies` class.
- Error occurred while getting entity child entities on archive.

---

## getDispositionHolds

```
public com.imis.util.IReadOnlyMap getDispositionHolds()  
    throws StorageConnectorException
```

Gets the collection of disposition holds sorted by their internal identifiers.

**Returns:**

A read-only map of `IDispositionHold` objects sorted by their identifiers.

(continued from last page)

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.

---

## authenticate

```
public void authenticate()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Authenticates user on archive.

**Throws:**

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.

IOException - if an I/O error occurs.

---

## getEntityReference

```
public IReadOnlyEntityReference getEntityReference(String referenceId)  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the entity reference using the specified entity reference identifier.

**Parameters:**

referenceId - the entity reference identifier.

**Returns:**

The IReadOnlyEntityReference object.

**Throws:**

NullPointerException - if referenceId is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.

IOException - if an I/O error occurs.

(continued from last page)

## createEntityReference

```
public IEntityReference createEntityReference()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Creates a new IEntityReference.

**Returns:**

The IEntityReference object.

**Throws:**

[CancellationException](#) - if an operation was canceled by the user.  
[LoginException](#) - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating entity on archive.

[IOException](#) - if an I/O error occurs.

---

## createEntity

```
public IEntity createEntity(String templateId,  
    EntityIdKind parentIdKind,  
    String parentId)  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Creates a new IEntity or sub-entity entity instance.

**Note:** This operation allows user to pass a reason for creating entity. Reason can be passed through [AuditLog](#) message.

**Note:** The instance is committed to the archive only if [IEntity.save\(\)](#) method is used on the new instance, otherwise it is discarded.

**Parameters:**

templateId - the entity template identifier from the templates inventory.  
parentIdKind - the parent entity identifier kind.  
parentId - the parent identifier.

**Returns:**

A new IEntity instance.

**Throws:**

[CancellationException](#) - if an operation was canceled by the user.  
[LoginException](#) - if a login error occurs.  
[NullPointerException](#) - if templateId is a null reference.



(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Template identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Error occurred while creating entity on archive.

IOException - if an I/O error occurs.

---

## createClass

```
public IClass createClass(String templateId)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Creates a new [IClass](#) entity on the first level of entity hierarchy based on the template identifier which defines its metadata scheme and default ACL state.

**Note:** This operation allows user to pass a reason for creating class. Reason can be passed through [AuditLog](#) message.

**Note:** The instance is committed to the archive only if [IEntity.save\(\)](#) method is used on the new instance, otherwise it is discarded.

### Parameters:

templateId - the class template identifier from the templates inventory.

### Returns:

A new [IClass](#) instance.

### Throws:

[NullPointerException](#) - if templateId is a null reference.

[CancellationException](#) - if an operation was canceled by the user.

[LoginException](#) - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Template identifier is required and cannot be empty.
- Operation on closed entity is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating entity on archive.

IOException - if an I/O error occurs.

---

## createClass

```
public IClass createClass(String templateId,
                        EntityIdKind parentIdKind,
                        String parentId)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

(continued from last page)

Creates a new `IClass` or sub-class entity instance.

**Note:** This operation allows user to pass a reason for creating class. Reason can be passed through [AuditLog](#) message.

**Note:** The instance is committed to the archive only if [IEntity.save\(\)](#) method is used on the new instance, otherwise it is discarded.

**Parameters:**

`templateId` - the class template identifier from the templates inventory.

`parentIdKind` - the parent entity identifier kind.

`parentId` - the parent identifier.

**Returns:**

A new `IClass` instance.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

`NullPointerException` - if `templateId` is a null reference.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Template identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Error occurred while creating entity on archive.

`IOException` - if an I/O error occurs.

---

## createFolder

```
public IFolder createFolder(String templateId,  
    EntityIdKind parentIdKind,  
    String parentId)  
    throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Creates a new `IFolder` or sub-folder entity instance.

**Note:** This operation allows user to pass a reason for creating folder. Reason can be passed through [AuditLog](#) message.

**Note:** The instance is committed to the archive only if [IEntity.save\(\)](#) method is used on the new instance, otherwise it is discarded.

**Parameters:**

`templateId` - the folder template identifier from the templates inventory.

`parentIdKind` - the parent identifier kind.

`parentId` - the parent identifier.

**Returns:**

A new `IFolder` instance.

**Throws:**

`NullPointerException` - if `templateId` is a null reference.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Template identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Error occurred while creating entity on archive.

IOException - if an I/O error occurs.

---

## createDocument

```
public IDocument createDocument(String templateId,
    EntityIdKind parentIdKind,
    String parentId)
    throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Creates a new IDocument entity instance.

**Note:** This operation allows user to pass a reason for creating document. Reason can be passed through [AuditLog](#) message.

**Note:** The instance is committed to the archive only if [IEntity.save\(\)](#) method is used on the new instance, otherwise it is discarded.

### Parameters:

templateId - the document template identifier from the templates inventory.  
 parentIdKind - the parent identifier kind.  
 parentId - the parent identifier.

### Returns:

A new IDocument instance.

### Throws:

NullPointerException - if templateId is a null reference.  
 CancellationException - if an operation was canceled by the user.  
 LoginException - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Template identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Error occurred while creating entity on archive.

IOException - if an I/O error occurs.

---

(continued from last page)

## openEntity

```
public IEntity openEntity(EntityIdKind idKind,  
    String id,  
    Integer sequence,  
    AccessMode mode,  
    IEntity syncEntity)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Opens a generic IEntity from the archive given by the entity identifier with the specified access mode.

The entity identifier can be either internal identifier, externally assigned or classification code.

**Note:** This operation allows user to pass a reason for opening entity. Reason can be passed through [AuditLog](#) message.

### Parameters:

idKind - the entity identifier kind.  
id - the entity identifier.  
sequence - the entity version sequence.  
mode - the entity access mode.  
syncEntity - the entity with which opened entity should be synchronized when used in sync dependent operations.

### Returns:

An IEntity instance.

### Throws:

NullPointerException - if id is a null reference.  
CancellationException - if an operation was canceled by the user.  
LoginException - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening entity on archive.

IOException - if an I/O error occurs.

---

## openClass

```
public IClass openClass(EntityIdKind idKind,  
    String id,  
    AccessMode mode)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Opens a IClass entity from the archive given by the class identifier with the specified access mode.

The class identifier can be either internal identifier, externally assigned or classification code.

**Note:** This operation allows user to pass a reason for opening class. Reason can be passed through [AuditLog](#) message.

### Parameters:

idKind - the class identifier kind.  
id - the class identifier.  
mode - the class access mode.

(continued from last page)

**Returns:**

A `IClass` instance.

**Throws:**

`NullPointerException` - if `id` is a null reference.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening entity on archive.
- Entity identifier does not represent a `IClass` entity.

`IOException` - if an I/O error occurs.

---

## openFolder

```
public IFolder openFolder(EntityIdKind idKind,  
                        String id,  
                        AccessMode mode)  
throws javax.security.auth.login.LoginException,  
      StorageConnectorException,  
      IOException
```

Opens a `IFolder` entity from the archive given by the folder identifier with the specified access mode.

The folder identifier can be either internal identifier, externally assigned or classification code.

**Note:** This operation allows user to pass a reason for opening folder. Reason can be passed through [AuditLog](#) message.

**Parameters:**

`idKind` - the folder identifier kind.

`id` - the folder identifier.

`mode` - the folder access mode.

**Returns:**

A `IFolder` instance.

**Throws:**

`NullPointerException` - if `id` is a null reference.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening entity on archive.
- Entity identifier does not represent a `IFolder` entity.

`IOException` - if an I/O error occurs.

(continued from last page)

## openDocument

```
public IDocument openDocument(EntityIdKind idKind,  
    String id,  
    Integer sequence,  
    AccessMode mode)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Opens a `IDocument` entity from the archive given by the document identifier with the specified access mode.

**Note:** This operation allows user to pass a reason for opening document. Reason can be passed through [AuditLog](#) message.

The document identifier can be either internal identifier, externally assigned or classification code.

### Parameters:

`idKind` - the document identifier kind.  
`id` - the document identifier.  
`sequence` - the document version sequence.  
`mode` - the document access mode.

### Returns:

A `IDocument` instance.

### Throws:

`NullPointerException` - if `id` is a null reference.  
`CancellationException` - if an operation was canceled by the user.  
`LoginException` - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening entity on archive.
- Entity identifier does not represent a `IDocument` entity.

`IOException` - if an I/O error occurs.

---

## changeEntityTemplate

```
public IEntity changeEntityTemplate(EntityIdKind idKind,  
    String id,  
    String templateId,  
    IEntity syncEntity)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Opens a generic `IEntity` from the archive given by the entity identifier in read-write mode and changes its template to the specified value. Entity needs to be saved in order for template switch to occur. Attribute values present in original and new template are preserved, others are cleared. All mutable entity operations are allowed on returned entity instances. **Note:** This operation is not supported on versioned entities, drafts or system entities (entities which are based on "sys:\*" templates). Using system templates ("sys:\*") as designated new templates is not permitted. **Note:** This operation allows user to pass a reason for opening entity. Reason can be passed through [AuditLog](#) message.

### Parameters:

`idKind` - the entity identifier kind.  
`id` - the entity identifier.

(continued from last page)

templateId - the entity's new template identifier from the templates inventory.

syncEntity - the entity with which opened entity should be synchronized when used in sync dependent operations.

**Returns:**

An IEntity instance.

**Throws:**

NullPointerException - if id is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening entity on archive.

IOException - if an I/O error occurs.

---

## moveEntity

```
public void moveEntity(EntityIdKind idKind,  
    String id,  
    EntityIdKind parentIdKind,  
    String parentId)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Moves an IEntity with the specified entity identifier under the specified parent entity.

The entity identifier can be either internal identifier, externally assigned or classification code.

**Note:** This operation requires user to pass a reason for moving entity. Reason must be passed through [AuditLog](#) message.

**Parameters:**

idKind - the entity identifier kind.

id - the entity identifier of the entity to move.

parentIdKind - the parent entity identifier kind.

parentId - the entity identifier of the new parent entity.

**Throws:**

NullPointerException - if id is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Parent entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while moving entity on archive.

IOException - if an I/O error occurs.

(continued from last page)

## deleteEntity

```
public void deleteEntity(EntityIdKind idKind,  
    String id)  
    throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Deletes the IEntity and associated resources.

Use this method to remove all resources associated with the entity and the entity itself. Delete right is required for this operation to succeed.

**Note:** This operation requires user to pass reason for deleting entity. Reason must be passed through [AuditLog](#) message.

**Note:** This operation requires description for deleted entity. Use this method only if the entity has its own description that will be used as deleted entity description, otherwise this method will fail.

**Note:** This operation is irreversible. Entity will not be accessible after a successful call is completed.

### Parameters:

idKind - the entity identifier kind.  
id - the entity identifier.

### Throws:

NullPointerException - if id is a null reference.  
CancellationException - if an operation was canceled by the user.  
LoginException - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while removing entity on archive.

IOException - if an I/O error occurs.

---

## checkOutEntity

```
public String checkOutEntity(EntityIdKind idKind,  
    String id,  
    Integer sequence)  
    throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Checks out an [IEntity](#).

**Note:** This operation allows user to pass reason for checking out entity. Reason can be passed through [AuditLog](#) message.

### Parameters:

idKind - the entity identifier kind.  
id - the entity identifier.  
sequence - the entity version sequence.

### Returns:

An [IEntityStub](#) instance of the entity private working copy.

### Throws:

NullPointerException - if id is a null reference.



(continued from last page)

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while checking out entity on archive.

IOException - if an I/O error occurs.

---

## checkInDraft

```
public String checkInDraft(String id,
    Short position,
    String comment)
    throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Checks in an entity draft.

**Note:** This operation allows user to pass reason for checking in entity draft. Reason can be passed through [AuditLog](#) message.

### Parameters:

id - the entity draft identifier.

position - the entity draft check in position in the versioning tree.

comment - the entity draft check in comment recorded in version history or a null reference if not specified.

### Returns:

An [IEntityStub](#) instance of the checked in entity.

### Throws:

NullPointerException - if id is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while checking out entity on archive.

IOException - if an I/O error occurs.

---

## discardDraft

```
public void discardDraft(String id)
    throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Discards an entity draft.

**Note:** This operation allows user to pass reason for discarding entity draft. Reason can be passed through [AuditLog](#) message.

(continued from last page)

**Parameters:**

id - the entity draft identifier.

**Throws:**

NullPointerException - if id is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while checking out entity on archive.

IOException - if an I/O error occurs.

---

## deleteEntity

```
public void deleteEntity(EntityIdKind idKind,  
    String id,  
    String description,  
    Integer sequence)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Deletes the IEntity and associated resources.

Use this method to remove all resources associated with the entity and the entity itself. Delete right is required for this operation to succeed.

**Note:** This operation requires user to pass reason for deleting entity. Reason must be passed through [AuditLog](#) message.

**Note:** This operation requires description for deleted entity. If one is not provided via description parameter, this method will fail if the entity does not have its own description.

**Note:** This operation is irreversible. Entity will not be accessible after a successful call is completed.

**Parameters:**

idKind - the entity identifier kind.

id - the entity identifier.

description - a custom description for deleted entity in case delete agent wants to specify it or a null reference to use existing entity description.

sequence - the entity version sequence.

**Throws:**

NullPointerException - if id is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Entity description is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while removing entity on archive.

IOException - if an I/O error occurs.

---

## getEntityInfo

```
public IEntityStub getEntityInfo(EntityIdKind idKind,
    String id)
    throws javax.security.auth.login.LoginException,
        StorageConnectorException,
        IOException
```

Queries archiving server for entity public properties (metadata attributes) and other specifics which can be accessed without opening an entity.

This method enables queries on single object, that is one entity identifier.

The entity identifier can be either internal identifier, externally assigned or classification code.

The returned IEntityStub object describe the public information about the entity which was queried.

### Parameters:

- idKind - the entity identifier kind.
- id - the entity identifier to be queried for their public information.

### Returns:

An IEntityStub object.

### Throws:

- NullPointerException - if id is a null reference.
- CancellationException - if an operation was canceled by the user.
- LoginException - if a login error occurs.
- [StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Identifier does not represent an entity.

IOException - if an I/O error occurs.

---

## getEntityInfo

```
public IEntityStub getEntityInfo(EntityIdKind idKind,
    String id,
    Set scope)
    throws javax.security.auth.login.LoginException,
        StorageConnectorException,
        IOException
```

(continued from last page)

Queries archiving server in the context of the specified user for entity public properties (metadata attributes) and other specifics which can be accessed without opening an entity and are specified by entity query scope.

This method enables queries on single object, that is one entity identifier.

The entity identifier can be either internal identifier, externally assigned or classification code.

The returned `IEntityStub` object describe the public information about the entity which was queried.

**Parameters:**

- `idKind` - the entity identifier kind.
- `id` - the entity identifier to be queried for their public information.
- `scope` - the entity query scope.

**Returns:**

An `IEntityStub` object.

**Throws:**

- `NullPointerException` - if `id` is a null reference.
- `CancellationException` - if an operation was canceled by the user.
- `LoginException` - if a login error occurs.
- [`StorageConnectorException`](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Identifier does not represent an entity.

`IOException` - if an I/O error occurs.

---

## getEntityInfo

```
public Map getEntityInfo(EntityIdKind idKind,
    Collection ids)
    throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Queries archiving server for entities public properties (metadata attributes) and other specifics which can be accessed without opening entities.

The entity identifier can be either internal identifier, externally assigned or classification code.

The returned `IEntityStub` objects describe the public information about each entity which was queried. The result is organized in a tree where the returning collection represents the collection of leaf stubs with their super stubs referenced as parent stubs.

**Parameters:**

- `idKind` - the entity identifier kind.
- `ids` - the collection of entity identifiers as `String` objects to be queried for their public information.

**Returns:**

A map of `IEntityStub` objects sorted by their unique identifiers.

**Throws:**

- `NullPointerException` - if `ids` is a null reference or contains a null reference.
- `CancellationException` - if an operation was canceled by the user.
- `LoginException` - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Collection of entity identifiers is required and cannot be empty.
- Entity identifier is required and cannot be empty.
- Entity identifiers are not of the same format.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Identifier does not represent an entity.

IOException - if an I/O error occurs.

**See Also:**[IEntityStub](#)

---

## getEntityInfo

```
public Map getEntityInfo(EntityIdKind idKind,
                        Collection ids,
                        Set scope)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Queries archiving server in the context of the specified user for the entities public properties (metadata attributes) and other specifics which can be accessed without opening a collection of entities and are specified by entity query scope.

The entity identifier can be either internal identifier, externally assigned or classification code.

The returned [IEntityStub](#) objects describe the public information about each entity which was queried. The result is organized in a tree where the returning collection represents the collection of leaf stubs with their super stubs referenced as parent stubs.

**Parameters:**

idKind - the entity identifier kind.

ids - the collection of entity identifiers as [String](#) objects to be queried for their public information.

scope - the entity query scope.

**Returns:**

A map of [IEntityStub](#) objects sorted by their unique identifiers.

**Throws:**

[NullPointerException](#) - if ids is a null reference or contains a null reference.

[CancellationException](#) - if an operation was canceled by the user.

[LoginException](#) - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Collection of entity identifiers is required and cannot be empty.
- Entity identifier is required and cannot be empty.
- Entity identifiers are not of the same format.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Identifier does not represent an entity.

IOException - if an I/O error occurs.

(continued from last page)

**See Also:**[IEntityStub](#)

---

## setEntityStatus

```
public void setEntityStatus(EntityIdKind idKind,  
    String id,  
    String status)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Sets the status of the entity given by the entity identifier.

**Note:** This operation allows user to pass a reason for changing entity status. Reason can be passed through [AuditLog](#) message.

**Parameters:**

idKind - the entity identifier kind.

id - the entity identifier.

status - the entity status or a null reference to reset status to inherited value.

**Throws:**

NullPointerException - if id is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while getting entities info on archive.
- Property is not present in template properties.
- Property value is not allowed based on the current template settings.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while setting status to value.

IOException - if an I/O error occurs.

---

## setEntitySecurityClass

```
public void setEntitySecurityClass(EntityIdKind idKind,  
    String id,  
    String securityClass)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Sets the security class of the entity given by the entity identifier.

**Note:** This operation requires user to pass a reason for changing entity security class. Reason must be passed through [AuditLog](#) message.

**Parameters:**

idKind - the entity identifier kind.

id - the entity identifier.

securityClass - the entity security class or a null reference to reset security class to inherited value.

**Throws:**

NullPointerException - if id is a null reference.

(continued from last page)

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while getting entities info on archive.
- Property is not present in template properties.
- Property value is not allowed based on the current template settings.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while setting security class to value.

IOException - if an I/O error occurs.

---

## getRootClasses

```
public IEntityCollection getRootClasses(Collection sortKeys,  
    Collection categorizationPropertyNames)  
    throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Returns the read-only collection the first level archive classes using the specified sort keys and a collection of property names used for categorization.

### Parameters:

sortKeys - the collection of EntitySortKey to be used for sorting the resulting entity collection or null if the collection does not need to be sorted.

categorizationPropertyNames - the collection of property names used for categorization.

### Returns:

The read-only collection of IEntityStub objects representing the first level archive classes.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting root classes on archive.

IOException - if an I/O error occurs.

### See Also:

[IEntityStub](#)

[EntitySortKey](#)

---

(continued from last page)

## getSubEntities

```
public ICollection getSubEntities(EntityIdKind idKind,
    String id,
    EntityFilter filter,
    Collection sortKeys,
    Collection categorizationPropertyNames)
throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Gets the read-only collection of child entity stubs using the specified entity filter, sort keys and a collection of property names used for categorization.

### Parameters:

idKind - the entity identifier kind.  
 id - the entity identifier.  
 filter - the entity filter or a null reference.  
 sortKeys - the collection of [EntitySortKey](#) objects.  
 categorizationPropertyNames - the collection of property names used for categorization.

### Returns:

The read-only collection of [IEntityStub](#) objects representing the entity children.

### Throws:

[CancellationException](#) - if an operation was canceled by the user.  
[LoginException](#) - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting root classes on archive.

[IOException](#) - if an I/O error occurs.

### See Also:

[IEntityStub](#)  
[EntitySortKey](#)

## getDeletedEntity

```
public IDeletionStub getDeletedEntity(EntityIdKind idKind,
    String id)
throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Queries archiving server for deleted entity public properties.

This method enables queries on single object, that is one deleted entity identifier.

The returned [IDeletionStub](#) object describe the public information about the deleted entity which was queried.

### Parameters:

idKind - the entity identifier kind.  
 id - the deleted entity identifier to be queried for their public information.

### Returns:

An [IDeletionStub](#) object.



(continued from last page)

**Throws:**

`NullPointerException` - if `id` is a null reference.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Identifier does not represent a deleted entity.

`IOException` - if an I/O error occurs.

---

## getDeletedEntities

```
public IEntityCollection getDeletedEntities(DeletionKind kind,  
                                           Collection sortKeys)  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Returns the read-only collection of deleted entities using the specified sort keys and a collection of property names used for categorization.

**Parameters:**

`kind` - the deletion kind of deleted entities.

`sortKeys` - the collection of `EntitySortKey` objects to be used for sorting the resulting deleted entity collection or null if the collection does not need to be sorted.

**Returns:**

The read-only collection of `IDeletionStub` objects representing the deleted entities.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Error occurred trying to access `Deleted` or `Disposed` class.

`IOException` - if an I/O error occurs.

**See Also:**

[IDeletionStub](#)

[EntitySortKey](#)

---

(continued from last page)

## getDrafts

```
public IEntityCollection getDrafts(Collection sortKeys,  
    Collection categorizationPropertyNames)  
    throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Returns the read-only collection of entity drafts using the specified sort keys and a collection of property names used for categorization.

### Parameters:

`sortKeys` - the collection of `EntitySortKey` objects to be used for sorting the resulting collection of entity drafts or null if the collection does not need to be sorted.  
`categorizationPropertyNames` - the collection of property names used for categorization.

### Returns:

The read-only collection of `IDraftStub` objects representing the drafts.

### Throws:

`CancellationException` - if an operation was canceled by the user.  
`LoginException` - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Error occurred trying to access `Drafts` class.
- Error occurred while getting entity child entities on archive.

`IOException` - if an I/O error occurs.

---

## getLogEntities

```
public IEntityCollection getLogEntities(LogType type,  
    Collection sortKeys,  
    Collection categorizationPropertyNames)  
    throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Returns the read-only collection of logged entities using the specified sort keys and a collection of property names used for categorization.

### Parameters:

`type` - the log type.  
`sortKeys` - the collection of `EntitySortKey` objects to be used for sorting the resulting logged entity collection or null if the collection does not need to be sorted.  
`categorizationPropertyNames` - the collection of property names used for categorization.

### Returns:

The read-only collection of `IEntityStub` objects representing the logged entities.

### Throws:

`IllegalArgumentException` - if invalid log type.  
`CancellationException` - if an operation was canceled by the user.  
`LoginException` - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Error occurred trying to access Import, Export or Transfer class.
- Error occurred while getting entity child entities on archive.

IOException - if an I/O error occurs.

**See Also:**[IEntityStub](#)[EntitySortKey](#)

---

## search

```
public IEntityCollection search(EntityFilter filter,
    Collection sortKeys,
    Set options,
    String searchExpression,
    Collection categorizationPropertyNames,
    Long maxElements)
throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Searches for child entity stubs of the archive using the specified entity search filter, sort keys, search options, expression string and a collection of property names used for categorization.

**Parameters:**

filter - the entity search filter or a null reference.  
 sortKeys - the collection of [EntitySortKey](#) objects.  
 options - the search options.  
 searchExpression - the search expression string.  
 categorizationPropertyNames - the collection of property names used for categorization.  
 maxElements - the number of elements the resulting collection should be truncated to.

**Returns:**

The read-only collection of [ISearchedEntityStub](#) objects representing the search results.

**Throws:**

CancellationException - if an operation was canceled by the user.  
 LoginException - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while performing search on archive.

IOException - if an I/O error occurs.

**See Also:**[ISearchedEntityStub](#)[EntityFilter](#)[EntitySortKey](#)[SearchOptions](#)

---

## search

```
public ICollection search(EntityIdKind idKind,  
    String id,  
    EntityFilter filter,  
    Collection sortKeys,  
    Set options,  
    String searchExpression,  
    Collection categorizationPropertyNames,  
    Long maxElements)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Searches for child entity stubs of the archive using the specified entity search filter, sort keys, search options, expression string and a collection of property names used for categorization.

### Parameters:

idKind - the entity identifier kind.  
id - the entity identifier.  
filter - the entity search filter or a null reference.  
sortKeys - the collection of [EntitySortKey](#) objects.  
options - the search options.  
searchExpression - the search expression string.  
categorizationPropertyNames - the collection of property names used for categorization.  
maxElements - the number of elements the resulting collection should be truncated to.

### Returns:

The read-only collection of [ISearchedEntityStub](#) objects representing the search results.

### Throws:

[CancellationException](#) - if an operation was canceled by the user.  
[LoginException](#) - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while performing search on archive.

[IOException](#) - if an I/O error occurs.

### See Also:

[ISearchedEntityStub](#)  
[EntityFilter](#)  
[EntitySortKey](#)  
[SearchOptions](#)

---

## createReview

```
public IReview createReview(IEntityStub scope,  
    Collection policies)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Creates the review given the list of retention policies and scope entity.

**Note:** This operation allows user to pass a reason for creating review. Reason can be passed through [AuditLog](#) message.

(continued from last page)

**Parameters:**

scope - the scope entity.  
policies - a collection of retention policies.

**Returns:**

An IReview instance.

**Throws:**

NullPointerException - if policies is a null reference.  
CancellationException - if an operation was canceled by the user.  
LoginException - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating entity on archive.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

IOException - if an I/O error occurs.

**See Also:**

[IReview](#)  
[IEntityStub](#)  
[IRetentionPolicy](#)

---

## createReview

```
public IReview createReview(IEntityStub scope,  
                           String queryExpression)  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Creates the review given the query expression and scope entity.

**Note:** This operation allows user to pass a reason for creating review. Reason can be passed through [AuditLog](#) message.

**Parameters:**

scope - the scope entity.  
queryExpression - the query expression.

**Returns:**

An IReview instance.

**Throws:**

NullPointerException - queryExpression is a null reference.  
CancellationException - if an operation was canceled by the user.  
LoginException - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating entity on archive.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

IOException - if an I/O error occurs.

**See Also:**[IReview](#)[IEntityStub](#)

---

## getReviews

```
public IEntityCollection getReviews(Collection sortKeys,
    Collection categorizationPropertyNames)
    throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Gets the read-only collection of review folders using the specified sort keys and a collection of property names used for categorization.

**Parameters:**

sortKeys - the collection of EntitySortKey objects.

categorizationPropertyNames - the collection of property names used for categorization.

**Returns:**

The read-only collection of IReviewStub objects representing the review folders.

**Throws:**

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entity child entities on archive.

IOException - if an I/O error occurs.

**See Also:**[IReviewStub](#)[EntitySortKey](#)

---

## getReviewInfo

```
public IReviewStub getReviewInfo(EntityIdKind idKind,
    String id)
    throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

(continued from last page)

Gets the review document info.

**Parameters:**

idKind - the review document identifier kind.  
id - the review document identifier.

**Returns:**

An IReviewStub stub.

**Throws:**

NullPointerException - if id is a null reference.  
CancellationException - if an operation was canceled by the user.  
LoginException - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Identifier does not represent a review.

IOException - if an I/O error occurs.

**See Also:**

[IReviewStub](#)

---

## getEffectiveRights

```
public IReadOnlyEntityRights getEffectiveRights(EntityIdKind idKind,
        String id,
        IDirectoryEntity subject)
throws javax.security.auth.login.LoginException,
        StorageConnectorException,
        IOException
```

Gets the users effective rights.

**Parameters:**

idKind - the entity identifier kind.  
id - the entity identifier.  
subject - the directory entity of the user to impersonate or a null reference for the current user.

**Returns:**

The read-only set of rights describing users effective rights.

**Throws:**

NullPointerException - if id is a null reference.  
CancellationException - if an operation was canceled by the user.  
LoginException - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.

IOException - if an I/O error occurs.

## openReview

```
public IReview openReview(EntityIdKind idKind,  
                          String id,  
                          AccessMode mode)  
throws javax.security.auth.login.LoginException,  
       StorageConnectorException,  
       IOException
```

Opens the review.

**Note:** This operation allows user to pass a reason for opening review. Reason can be passed through [AuditLog](#) message.

### Parameters:

idKind - the review document identifier kind.

id - the review document identifier.

mode - the review document access mode.

### Returns:

The [IReview](#) instance.

### Throws:

[CancellationException](#) - if an operation was canceled by the user.

[LoginException](#) - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening entity on archive.

[IOException](#) - if an I/O error occurs.

### See Also:

[IReview](#)

---

## auditLogQuery

```
public InputStream auditLogQuery(AuditQuery parameters)  
throws javax.security.auth.login.LoginException,  
       StorageConnectorException,  
       IOException
```

Gets the audit log query results data input stream.

### Parameters:

parameters - the audit log query parameters.

### Returns:

The audit log query results data input stream.

### Throws:

[NullPointerException](#) - if parameters is a null reference.

[CancellationException](#) - if an operation was canceled by the user.

[LoginException](#) - if a login error occurs.



(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error getting audit query results data stream.

[IOException](#) - if an I/O error occurs.

**See Also:**

[AuditQuery](#)

---

## getReport

```
public InputStream getReport(ReportType type)
    throws StorageConnectorException,
           IOException
```

Gets the audit log query results data input stream.

**Parameters:**

`type` - the audit log query parameters.

**Returns:**

The report input data stream or a null reference if there is no report for the specified report type.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while getting archive report.

[IOException](#) - if an I/O error occurs.

**See Also:**

[ReportType](#)

---

## getSystemPropertyName

```
public String getSystemPropertyName(SystemProperty systemProperty)
```

Gets the system property name.

**Parameters:**

`systemProperty` - the system property.

**Returns:**

The system property name.

**Throws:**

[IllegalArgumentException](#) - if invalid system property.

**See Also:**

[SystemProperty](#)

---

(continued from last page)

## getPublicClassificationCode

```
public String getPublicClassificationCode(String classificationCode)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Converts the specified classification code in canonical format to the public classification code.

### Parameters:

classificationCode - the classification code in canonical format (e.g. "C=001^C=001^C="+...+"^F=2004^F="+...+"^D=0001", where "C=", "F=" and "D=" denote class, folder and document respectively and "^" is the classification hierarchy delimiter).

### Returns:

The public classification code (e.g. "001.001."+...+"-2004-"+...+"/0001", where ".", "-" and "/" are class, folder and document entity type hierarchy delimiters) or a null reference if classificationCode is null.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred converting classification code to public format.

IOException - if an I/O error occurs.

---

## getSystemPropertySortKey

```
public EntitySortKey getSystemPropertySortKey(SystemProperty systemProperty,
      EntitySortKeyDirection direction)
```

Gets the entity sort key for sorting entity collection by the specified system property.

### Parameters:

systemProperty - the system property.

direction - the sorting direction.

### Returns:

An EntitySortKey object used for sorting entity collection by the specified system property.

### See Also:

[SystemProperty](#)

[EntitySortKey](#)

[EntitySortKeyDirection](#)

---

## getInternalId

```
public String getInternalId(String externalId)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Resolves the specified external identifier to the entity identifier that external identifier is associated with.

### Parameters:

externalId - an external entity identifier.

(continued from last page)

**Returns:**

The entity identifier that `externalId` is associated with.

**Throws:**

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while getting internal identifier.

`IOException` - if an I/O error occurs.

---

## getDraftInfo

```
public IDraftStub getDraftInfo(String id)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Queries archiving server for draft public properties (metadata attributes) and other specifics which can be accessed without opening a draft. The returned `IDraftStub` object describes the public information about the draft which was queried.

**Parameters:**

`id` - the draft identifier to be queried for their public information.

**Returns:**

An `IDraftStub` object.

**Throws:**

`NullPointerException` - if `id` is a null reference.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Draft identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Identifier does not represent a draft.

`IOException` - if an I/O error occurs.

---

## openDraft

```
public IDraft openDraft(String id,
    AccessMode mode)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Opens a `IDraft` from the archive given by the draft identifier with the specified access mode. **Note:** This operation allows user to pass a reason for opening draft. Reason can be passed through [AuditLog](#) message.

**Parameters:**

`id` - the draft identifier.

`mode` - the draft access mode.

(continued from last page)

**Returns:**

An `IDraft` instance.

**Throws:**

`NullPointerException` - if `id` is a null reference.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[`StorageConnectorException`](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening draft on archive.

`IOException` - if an I/O error occurs.

---

**close**

```
public void close()
```

Closes all active sessions to an archive.

## com.imis.storageconnector

### Class IASessionPool

java.lang.Object

└─com.imis.storageconnector.IASessionPool

public class **IASessionPool**  
extends Object

Represents a session pool with one storage server node.

#### Constructor Summary

public	<a href="#"><u>IASessionPool</u></a> (String host, com.imis.imisarc.client.IASessionOptions options, com.imis.imisarc.client.IASessionContext context, int maxCount, boolean persistent, com.imis.imisarc.client.IConnectingListener connectingListener, com.imis.imisarc.client.IAuthenticatingListener authenticatingListener) Initializes a new instance of the <a href="#"><u>IASessionPool</u></a> class.
--------	---

#### Method Summary

int	<a href="#"><u>acquire</u></a> () Acquires the reference to the session pool.
void	<a href="#"><u>acquireSession</u></a> ( <a href="#"><u>IASession</u></a> session) Acquires the specified session and removes it from the session pool, if the session reference count equals 1.
boolean	<a href="#"><u>addSession</u></a> ( <a href="#"><u>IASession</u></a> session) Adds the session to the session pool.
Date	<a href="#"><u>cleanSessions</u></a> (Date nextCheck, int period) Cleans sessions that were not used for a specified period or more from the session pool.
void	<a href="#"><u>closeSessions</u></a> () Closes all sessions in the session pool.
com.imis.imisarc.client.IASessionContext	<a href="#"><u>getContext</u></a> () Gets the session pool context.
boolean	<a href="#"><u>getPersistent</u></a> () Gets a value indicating whether session pool is persistent or not.
<a href="#"><u>IASession</u></a>	<a href="#"><u>getSession</u></a> () Gets an authenticated session from the session pool.
int	<a href="#"><u>release</u></a> () Releases the reference to the session pool.
void	<a href="#"><u>releaseSession</u></a> ( <a href="#"><u>IASession</u></a> session) Releases the specified session and returns it to the session pool, if the session reference count equals 0.
void	<a href="#"><u>setPersistent</u></a> (boolean value) Sets a value indicating whether session pool is persistent or not.

**Methods inherited from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructors

### IASessionPool

```
public IASessionPool(String host,
                     com.imis.imisarc.client.IASessionOptions options,
                     com.imis.imisarc.client.IASessionContext context,
                     int maxCount,
                     boolean persistent,
                     com.imis.imisarc.client.IConnectingListener connectingListener,
                     com.imis.imisarc.client.IAuthenticatingListener
                     authenticatingListener)
```

Initializes a new instance of the [IASessionPool](#) class.

Sessions in a persistent session pool is not removed from the list of session context session pools.

**Parameters:**

`host` - a host of the IMiS/ARC Storage Server.

`options` - a IMiS/ARC Client session options.

`context` - an IMiS/ARC Client authenticated session context.

`maxCount` - the maximum number of sessions in the session pool.

`persistent` - true to indicate the session pool is persistent; otherwise, false.

`connectingListener` - the IMiS/ARC Client session Connecting event listener.

`authenticatingListener` - the IMiS/ARC Client session Authenticating event listener.

## Methods

### getContext

```
public com.imis.imisarc.client.IASessionContext getContext()
```

Gets the session pool context.

**Returns:**

The session pool context.

### getPersistent

```
public boolean getPersistent()
```

Gets a value indicating whether session pool is persistent or not.

**Returns:**

true if session pool is persistent; otherwise, false.

### setPersistent

```
public void setPersistent(boolean value)
```

Sets a value indicating whether session pool is persistent or not.

---

(continued from last page)

**Parameters:**

value - true if session pool is persistent; otherwise, false.

---

**acquire**

```
public int acquire()
```

Acquires the reference to the session pool.

This method increments and returns the session pool reference count.

**Returns:**

The session pool reference count.

---

**release**

```
public int release()
```

Releases the reference to the session pool.

This method decrements and returns the session pool reference count.

**Returns:**

The session pool reference count.

---

**closeSessions**

```
public void closeSessions()
```

Closes all sessions in the session pool.

---

**cleanSessions**

```
public Date cleanSessions(Date nextCheck,  
                           int period)
```

Cleans sessions that were not used for a specified period or more from the session pool.

**Parameters:**

nextCheck - the time the next session clean is scheduled.

period - the period for default session clean check in seconds.

**Returns:**

The time the next clean should be performed. Default return value, used when there are no sessions in the session pool, is set to Now + period.

---

**addSession**

```
public boolean addSession(IASession session)
```

Adds the session to the session pool.

**Parameters:**

session - the IMiS/ARC Client session to add.

**Returns:**

true if the session was added to the session pool; otherwise false.

---

(continued from last page)

## getSession

```
public IASession getSession()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets an authenticated session from the session pool.

This method either retrieves and removes an existing session from the session pool or creates a new session. Whatever the case, the resulting session must be passed as a parameter to the [release\(\)](#) method to place the session back in the session pool after use.

### Returns:

An IMiS/ARC Client session.

### Throws:

[LoginException](#) - if a login error occurs.  
[StorageConnectorException](#) - if

- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.

[IOException](#) - if an I/O error occurs.

---

## acquireSession

```
public void acquireSession(IASession session)
```

Acquires the specified session and removes it from the session pool, if the session reference count equals 1.

### Parameters:

`session` - the IMiS/ARC Client session to acquire.

---

## releaseSession

```
public void releaseSession(IASession session)
```

Releases the specified session and returns it to the session pool, if the session reference count equals 0.

### Parameters:

`session` - the IMiS/ARC Client session to release.



com.imis.storageconnector

Class IASessionPoolManager

```
java.lang.Object
└--com.imis.storageconnector.IASessionPoolManager
```

```
public class IASessionPoolManager
extends Object
```

Represents the IMiS/Storage Connector session pool manager.

IASessionPoolManager is a singleton class that allows a single instance of this class to be created.

Method Summary	
void	<a href="#">close()</a> Closes the session pool manager thread.
static <a href="#">IASessionPoolManager</a>	<a href="#">getInstance()</a> Gets a singleton instance of the <a href="#">IASessionPoolManager</a> class.
<a href="#">IASessionPool</a>	<a href="#">getSessionPool</a> (String host, Map options, com.imis.imisarc.client.IConnectingListener connectingListener, com.imis.imisarc.client.IAuthenticatingListener authenticatingListener) Gets the session pool for context based on the specified host and storage/archive options.
void	<a href="#">releaseSessionPool</a> ( <a href="#">IASessionPool</a> sessionPool) Releases the session pool and removes it from the list of context session pools if the reference count value after release equals to 0.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods

```
getInstance

public static IASessionPoolManager getInstance()

Gets a singleton instance of the IASessionPoolManager class.

Returns:
    An IASessionPoolManager instance.
```

```
close

public void close()

Closes the session pool manager thread.
```

(continued from last page)

## getSessionPool

```
public IASessionPool getSessionPool(String host,  
    Map options,  
    com.imis.imisarc.client.IConnectingListener connectingListener,  
    com.imis.imisarc.client.IAuthenticatingListener authenticatingListener)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Gets the session pool for context based on the specified host and storage/archive options.

This method is used to get the context session pool for the `IArchive` and `IStorage` instances.

### Parameters:

host - a storage/archive host name or IP address and port.

options - a map of storage/archive options.

connectingListener - the IMiS/ARC Client session Connecting event listener.

authenticatingListener - the IMiS/ARC Client session Authenticating event listener.

### Returns:

The context session pool.

### Throws:

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.

`IOException` - if an I/O error occurs.

---

## releaseSessionPool

```
public void releaseSessionPool(IASessionPool sessionPool)
```

Releases the session pool and removes it from the list of context session pools if the reference count value after release equals to 0.

### Parameters:

sessionPool - the context session pool.

# com.imis.storageconnector

## Interface IAuthenticatingListener

public interface **IAuthenticatingListener**  
extends `EventListener`

The listener interface for receiving [AuthenticatingEvent](#).

Method Summary	
void	<a href="#">authenticating</a> ( <a href="#">AuthenticatingEvent</a> e) This method gets called when data for authentication is needed.

### Methods

#### authenticating

public void **authenticating**([AuthenticatingEvent](#) e)

This method gets called when data for authentication is needed.

**Parameters:**

e - the `AuthenticatingEvent` object.

# com.imis.storageconnector

## Interface IConnectingListener

public interface **IConnectingListener**  
extends

The listener interface for receiving [ConnectingEvent](#).

Method Summary	
void	<a href="#">connecting</a> ( <a href="#">ConnectingEvent</a> e) This method gets called when session is attempting to connect.

### Methods

#### connecting

public void **connecting**([ConnectingEvent](#) e)

This method gets called when session is attempting to connect.

**Parameters:**

e - the [ConnectingEvent](#) object.

## com.imis.storageconnector Interface IContentTypeResolver

public interface **IContentTypeResolver**  
extends

Defines the methods used by external content type and extension resolver.

### Method Summary

String	<a href="#"><code>getContentType</code></a> (String extension) Gets the first available content type for a specified extension.
List	<a href="#"><code>getContentTypes</code></a> (String extension) Gets the content types for a specified extension.
String	<a href="#"><code>getExtension</code></a> (String contentType) Gets the first available extension for a specified content type.
List	<a href="#"><code>getExtensions</code></a> (String contentType) Gets the extensions for a specified content type.

### Methods

#### **getExtension**

public String **getExtension**(String contentType)

Gets the first available extension for a specified content type.

**Parameters:**

contentType - the file format content type (case sensitive).

**Returns:**

The first available extension (including the leading period) for the specified content type or a null reference if no extensions are defined.

**Throws:**

NullPointerException - if contentType is a null reference.

#### **getExtensions**

public List **getExtensions**(String contentType)

Gets the extensions for a specified content type.

**Parameters:**

contentType - the file format content type (case sensitive).

**Returns:**

A list of extensions (including the leading period) for the specified content type.

**Throws:**

(continued from last page)

NullPointerException - if contentType is a null reference.

---

## getContentType

```
public String getContentType(String extension)
```

Gets the first available content type for a specified extension.

**Parameters:**

extension - a file format extension (with or without a leading period).

**Returns:**

The first available content type for a specified extension or a null reference if no content types are defined.

**Throws:**

NullPointerException - if extension is a null reference.

---

## getContentTypes

```
public List getContentTypes(String extension)
```

Gets the content types for a specified extension.

**Parameters:**

extension - a file format extension (with or without a leading period).

**Returns:**

A list of content types for a specified extension.

**Throws:**

NullPointerException - if extension is a null reference.

# com.imis.storageconnector

## Class ObjectIdEncoding



All Implemented Interfaces:  
Serializable, Comparable

public final class **ObjectIdEncoding**  
extends Enum

Defines constants that represent the encoding of object identifiers used when creating new objects.

Constants are used as values for the [StorageConnector.OPTION\\_OBJECT\\_IDENCODING](#) IMIS\_ARC storage object identifier encoding disposition option.

See Also:  
[StorageConnector.openIMiSARCStorage\(String, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, String, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, int, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, int, String, Map\)](#)

Field Summary	
public static final	<a href="#">BASE16</a> Base 16 encoding.
public static final	<a href="#">BASE64</a> Base 64 encoding.
public static final	<a href="#">BASE85</a> Base 85 encoding.

Method Summary	
static <a href="#">ObjectIdEncoding</a>	<a href="#">valueOf</a> (String name)
static <a href="#">ObjectIdEncoding[]</a>	<a href="#">values</a> ()

Methods inherited from class java.lang.Enum	
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf	

Methods inherited from class java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

Methods inherited from interface java.lang.Comparable	
compareTo	

## Fields

### BASE16

```
public static final com.imis.storageconnector.ObjectIdEncoding BASE16
```

Base 16 encoding.

---

### BASE64

```
public static final com.imis.storageconnector.ObjectIdEncoding BASE64
```

Base 64 encoding.

---

### BASE85

```
public static final com.imis.storageconnector.ObjectIdEncoding BASE85
```

Base 85 encoding.

## Methods

### values

```
public final static ObjectIdEncoding\[\] values()
```

---

### valueOf

```
public static ObjectIdEncoding valueOf(String name)
```



## com.imis.storageconnector Class ObjectIdKind

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.ObjectIdKind
  
```

**All Implemented Interfaces:**  
Serializable, Comparable

```

public final class ObjectIdKind
extends Enum
  
```

Defines constants that represent the kind of object identifiers used when opening objects.

Constants are used as values for the [StorageConnector.OPTION\\_OBJECT\\_IDKIND](#) IMIS\_ARC storage object identifier kind disposition option.

**See Also:**

[StorageConnector.openIMiSARCStorage\(String, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, String, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, int, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, int, String, Map\)](#)

### Field Summary

public static final	<a href="#">EXTERNAL</a> Object identifier is external.
public static final	<a href="#">INTERNAL</a> Object identifier is internal.
public static final	<a href="#">UNKNOWN</a> Object identifier type is not known.

### Method Summary

static <a href="#">ObjectIdKind</a>	<a href="#">valueOf</a> (String name)
static <a href="#">ObjectIdKind[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

## Fields

### UNKNOWN

```
public static final com.imis.storageconnector.ObjectIdKind UNKNOWN
```

Object identifier type is not known.

Object identifier type can be either internal or external.

---

### INTERNAL

```
public static final com.imis.storageconnector.ObjectIdKind INTERNAL
```

Object identifier is internal.

[Document.getId\(\)](#) represents an internal object identifier.

---

### EXTERNAL

```
public static final com.imis.storageconnector.ObjectIdKind EXTERNAL
```

Object identifier is external.

External object identifiers can be associated with an object using the [Storage.setExternalObjectId\(String, String\)](#) method.

---

## Methods

### values

```
public final static ObjectIdKind\[\] values()
```

---

### valueOf

```
public static ObjectIdKind valueOf(String name)
```

## com.imis.storageconnector Class ObjectIdType

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.ObjectIdType
  
```

**All Implemented Interfaces:**  
Serializable, Comparable

public final class **ObjectIdType**  
extends Enum

Defines constants that represent the types of object identifiers used when creating new objects.

Constants are used as values for the [StorageConnector.OPTION\\_OBJECT\\_IDTYPE](#) IMIS\_ARC storage object identifier type disposition option.

**See Also:**

[StorageConnector.openIMiSARCStorage\(String, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, String, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, int, Map\)](#),  
[StorageConnector.openIMiSARCStorage\(String, int, String, Map\)](#)

### Field Summary

public static final	<a href="#">ID192</a> 192 bit long object identifier.
public static final	<a href="#">ID256</a> 256 bit long object identifier.

### Method Summary

static <a href="#">ObjectIdType</a>	<a href="#">valueOf</a> (String name)
static <a href="#">ObjectIdType[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

(continued from last page)

## Fields

### ID192

```
public static final com.imis.storageconnector.ObjectIdType ID192
```

192 bit long object identifier.

---

### ID256

```
public static final com.imis.storageconnector.ObjectIdType ID256
```

256 bit long object identifier.

---

## Methods

### values

```
public final static ObjectIdType\[\] values()
```

---

### valueOf

```
public static ObjectIdType valueOf(String name)
```

## com.imis.storageconnector Class Storage

java.lang.Object

└-com.imis.storageconnector.Storage

### All Implemented Interfaces:

com.imis.IAutoCloseable

public abstract class **Storage**  
extends Object  
implements com.imis.IAutoCloseable

Abstract base class for storage supported by IMiS/Storage Connector.

**Note:** Only IMIS\_ARC storage is currently implemented by IMiS/Storage Connector. Use one of the `StorageConnector.openIMiSARCStorage` methods to create an IMIS\_ARC storage.

### See Also:

[StorageConnector.openIMiSARCStorage\(String\)](#), [StorageConnector.openIMiSARCStorage\(String, int\)](#), [StorageConnector.openIMiSARCStorage\(String, Map\)](#), [StorageConnector.openIMiSARCStorage\(String, String\)](#), [StorageConnector.openIMiSARCStorage\(String, int, Map\)](#), [StorageConnector.openIMiSARCStorage\(String, int, String\)](#), [StorageConnector.openIMiSARCStorage\(String, String, Map\)](#), [StorageConnector.openIMiSARCStorage\(String, int, String, Map\)](#)

## Constructor Summary

protected	<a href="#">Storage(StorageType storeType)</a> Initializes a new instance of the Storage class.
-----------	--

## Method Summary

void	<a href="#">addAuthenticatingListener(IAuthenticatingListener listener)</a> Adds an <a href="#">IAuthenticatingListener</a> object.
void	<a href="#">addConnectingListener(IConnectingListener listener)</a> Adds an <a href="#">IConnectingListener</a> object.
abstract InputStream	<a href="#">auditLogQuery(AuditQuery parameters)</a> Gets the audit log query results data input stream.
abstract InputStream	<a href="#">auditLogQuery(AuditQuery parameters, Map options)</a> Gets the audit log query results data input stream using the specified object options.
abstract void	<a href="#">close()</a> Closes the storage.
abstract <a href="#">Document</a>	<a href="#">createObject(String profile, String mime)</a> Creates a document in the storage profile using the specified MIME type.
abstract <a href="#">Document</a>	<a href="#">createObject(String profile, String mime, Map options)</a> Creates a document in the storage profile using the specified MIME type and options.
abstract void	<a href="#">deleteObject(String id)</a> Deletes a document from the storage.

abstract void	<a href="#"><u>deleteObject</u></a> (String id, Map options) Deletes a document from the storage using the specified options.
abstract <a href="#"><u>AuditLog</u></a>	<a href="#"><u>getAuditLog</u></a> () Gets the AuditLog object from the thread safe local store.
abstract long	<a href="#"><u>getCapacity</u></a> () Gets the storage capacity.
abstract long	<a href="#"><u>getProfileCapacity</u></a> (String profile) Gets the storage profile capacity, in bytes.
abstract int	<a href="#"><u>getProfileObjectCount</u></a> (String profile) Gets the storage profile object count.
abstract long	<a href="#"><u>getProfileSpaceAvailable</u></a> (String profile) Gets the storage profile available space, in bytes.
abstract long	<a href="#"><u>getProfileSpaceUsed</u></a> (String profile) Gets the storage profile used space, in bytes.
abstract long	<a href="#"><u>getSpaceAvailable</u></a> () Gets the storage available space.
abstract long	<a href="#"><u>getSpaceUsed</u></a> () Gets the storage used space.
abstract String	<a href="#"><u>getStoreInfo</u></a> () Gets the storage information.
<a href="#"><u>StorageType</u></a>	<a href="#"><u>getStoreType</u></a> () Gets the storage type.
boolean	<a href="#"><u>isClosed</u></a> () Gets a value indicating whether the storage is closed.
abstract void	<a href="#"><u>moveObject</u></a> (String id, String profile) Moves a document to the storage profile.
abstract <a href="#"><u>Document</u></a>	<a href="#"><u>moveObject</u></a> (String id, String profile, <a href="#"><u>DocumentAccess</u></a> mode) Moves a document to the storage profile and opens the document with the specified access mode.
abstract int	<a href="#"><u>objectCount</u></a> () Gets the number of objects on storage.
abstract <a href="#"><u>Document</u></a>	<a href="#"><u>openObject</u></a> (String id, <a href="#"><u>DocumentAccess</u></a> mode) Opens a document on the storage using the specified access mode.
abstract <a href="#"><u>Document</u></a>	<a href="#"><u>openObject</u></a> (String id, <a href="#"><u>DocumentAccess</u></a> mode, Map options) Opens a document on the storage using the specified access mode and options.
void	<a href="#"><u>removeAuthenticatingListener</u></a> ( <a href="#"><u>IAuthenticatingListener</u></a> listener) Removes an <a href="#"><u>IAuthenticatingListener</u></a> object.
void	<a href="#"><u>removeConnectingListener</u></a> ( <a href="#"><u>IConnectingListener</u></a> listener) Removes an <a href="#"><u>IConnectingListener</u></a> object.

abstract String	<a href="#"><u>resolveExternalObjectId</u></a> (String externalId) Resolves the specified external identifier to the object identifier that external identifier is associated with using the object options.
abstract String	<a href="#"><u>resolveExternalObjectId</u></a> (String externalId, Map options) Resolves the specified external identifier to the object identifier that external identifier is associated with using the specified options.
abstract String	<a href="#"><u>retrieveObject</u></a> (String id) Retrieves a document from the storage to a temporary file.
abstract String	<a href="#"><u>retrieveObject</u></a> (String id, OutputStream target) Retrieves a document from the storage and copies the contents to the specified OutputStream.
abstract String	<a href="#"><u>retrieveObject</u></a> (String id, String target) Retrieves a document from the storage to a specified local file.
abstract void	<a href="#"><u>setExternalObjectId</u></a> (String id, String externalId) Sets the association of the specified external identifier with an object on the IMiS/ARC Storage Server.
void	<a href="#"><u>setIsClosed</u></a> (boolean value) Sets a value indicating whether the storage is closed.
<a href="#"><u>Document</u></a>	<a href="#"><u>storeObject</u></a> ( <a href="#"><u>Document</u></a> source, String profile) Stores a source document on the storage profile with the source document default MIME type and returns a document opened with the source document access mode.
abstract String	<a href="#"><u>storeObject</u></a> ( <a href="#"><u>Document</u></a> source, String profile, String mime) Stores a source document on the storage profile with the specified MIME type and returns a document identifier.
abstract <a href="#"><u>Document</u></a>	<a href="#"><u>storeObject</u></a> ( <a href="#"><u>Document</u></a> source, String profile, String mime, <a href="#"><u>DocumentAccess</u></a> mode) Stores a source document on the storage profile with the specified MIME type and returns a document opened with the specified access mode.
abstract String	<a href="#"><u>storeObject</u></a> (InputStream source, String profile, String mime) Stores InputStream contents on the storage profile with the specified MIME type and returns a document identifier.
abstract <a href="#"><u>Document</u></a>	<a href="#"><u>storeObject</u></a> (InputStream source, String profile, String mime, <a href="#"><u>DocumentAccess</u></a> mode) Stores InputStream contents on the storage profile with the specified MIME type and returns a document opened with the specified access mode.
abstract String	<a href="#"><u>storeObject</u></a> (String source, String profile, String mime) Stores a local file on the storage profile with the specified MIME type and returns a document identifier.
abstract <a href="#"><u>Document</u></a>	<a href="#"><u>storeObject</u></a> (String source, String profile, String mime, <a href="#"><u>DocumentAccess</u></a> mode) Stores a local file on the storage profile with the specified MIME type and returns a document opened with the specified access mode.

**Methods inherited from class** java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods inherited from interface** com.imis.IAutoCloseable

---

close

---

---

## Constructors

### Storage

protected **Storage**([StorageType](#) storeType)

Initializes a new instance of the Storage class.

**Parameters:**

storeType - a storage type.

**Throws:**

[StorageConnectorException](#) - if demo license expired.

---

## Methods

### addConnectingListener

public void **addConnectingListener**([IConnectingListener](#) listener)

Adds an [IConnectingListener](#) object.

**Parameters:**

listener - an IConnectingListener.

---

### removeConnectingListener

public void **removeConnectingListener**([IConnectingListener](#) listener)

Removes an [IConnectingListener](#) object.

**Parameters:**

listener - an IConnectingListener.

---

### addAuthenticatingListener

public void **addAuthenticatingListener**([IAuthenticatingListener](#) listener)

Adds an [IAuthenticatingListener](#) object.

**Parameters:**

listener - an IAuthenticatingListener.

---

### removeAuthenticatingListener

public void **removeAuthenticatingListener**([IAuthenticatingListener](#) listener)

Removes an [IAuthenticatingListener](#) object.

**Parameters:**

listener - an IAuthenticatingListener.

---



(continued from last page)

## isClosed

```
public boolean isClosed()
```

Gets a value indicating whether the storage is closed.

**Returns:**

true if the storage is closed; otherwise, false.

---

## setIsClosed

```
protected void setIsClosed(boolean value)
```

Sets a value indicating whether the storage is closed.

**Parameters:**

value - true if the storage is closed; otherwise, false.

---

## getStoreType

```
public StorageType getStoreType()
```

Gets the storage type.

**Returns:**

A storage type.

---

## getStoreInfo

```
public abstract String getStoreInfo()  
throws StorageConnectorException
```

Gets the storage information.

**Returns:**

A storage information.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

---

## getCapacity

```
public abstract long getCapacity()  
throws StorageConnectorException,  
IOException
```

Gets the storage capacity.

**Note:** This feature is currently not supported on IMIS\_ARC storage.

**Returns:**

Storage capacity, in bytes.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IOException - if an I/O error occurs.

(continued from last page)

## getSpaceAvailable

```
public abstract long getSpaceAvailable()  
    throws StorageConnectorException,  
           IOException
```

Gets the storage available space.

**Note:** This feature is currently not supported on IMIS\_ARC storage.

**Returns:**

Available space on the storage, in bytes.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IOException - if an I/O error occurs.

---

## getSpaceUsed

```
public abstract long getSpaceUsed()  
    throws StorageConnectorException,  
           IOException
```

Gets the storage used space.

**Note:** This feature is currently not supported on IMIS\_ARC storage.

**Returns:**

Used space on the storage, in bytes.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IOException - if an I/O error occurs.

---

## objectCount

```
public abstract int objectCount()  
    throws StorageConnectorException,  
           IOException
```

Gets the number of objects on storage.

**Note:** This feature is currently not supported on IMIS\_ARC storage.

**Returns:**

The number of objects on storage.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IOException - if an I/O error occurs.

---

## getAuditLog

```
public abstract AuditLog getAuditLog()
```

Gets the AuditLog object from the thread safe local store.

**Returns:**

An AuditLog instance.

---

## createObject

```
public abstract Document createObject(String profile,
                                       String mime)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Creates a document in the storage profile using the specified MIME type.

### Parameters:

profile - a name of the profile where the new document is created.  
mime - a MIME type of the new document.

### Returns:

A Document instance.

### Throws:

NullPointerException - if mime is a null reference.  
CancellationException - if an operation was canceled by the user.  
LoginException - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating an object.

IOException - if an I/O error occurs.

---

## createObject

```
public abstract Document createObject(String profile,
                                       String mime,
                                       Map options)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Creates a document in the storage profile using the specified MIME type and options.

Applicable options are [StorageConnector.OPTION\\_OBJECT\\_IDTYPE](#) and [StorageConnector.OPTION\\_OBJECT\\_IDENCODING](#). IMIS\_ARC storage object identifier encoding disposition option.

### Parameters:

profile - a name of the profile where the new document is created.  
mime - a MIME type of the new document.  
options - a map of object options from which desired object identifier type and encoding can be obtained.

### Returns:

A Document instance.

### Throws:

NullPointerException - if mime is a null reference.  
CancellationException - if an operation was canceled by the user.  
LoginException - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
 IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating an object.

[IOException](#) - if an I/O error occurs.

---

## openObject

```
public abstract Document openObject(String id,
    DocumentAccess mode)
    throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Opens a document on the storage using the specified access mode.

### Parameters:

id - a document identifier.  
 mode - a document access mode.

### Returns:

A [Document](#) instance.

### Throws:

[NullPointerException](#) - if id is a null reference.  
[CancellationException](#) - if an operation was canceled by the user.  
[LoginException](#) - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
 IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening an object.

[IOException](#) - if an I/O error occurs.

---

## openObject

```
public abstract Document openObject(String id,
    DocumentAccess mode,
    Map options)
    throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Opens a document on the storage using the specified access mode and options.

Applicable option is [StorageConnector.OPTION\\_OBJECT\\_IDKIND](#).

### Parameters:

id - a document identifier.  
 mode - a document access mode.  
 options - a map of object options from which desired object identifier kind can be obtained.

(continued from last page)

**Returns:**

A Document instance.

**Throws:**

NullPointerException - if id is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening an object.

IOException - if an I/O error occurs.

---

## moveObject

```
public abstract void moveObject(String id,  
    String profile)  
    throws javax.security.auth.login.LoginException,  
        StorageConnectorException,  
        IOException
```

Moves a document to the storage profile.

**Parameters:**

id - a document identifier.

profile - a name of the profile where the document is moved.

**Throws:**

NullPointerException - if id is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while moving an object.

IOException - if an I/O error occurs.

---

## moveObject

```
public abstract Document moveObject(String id,  
    String profile,  
    DocumentAccess mode)  
    throws javax.security.auth.login.LoginException,  
        StorageConnectorException,  
        IOException
```

Moves a document to the storage profile and opens the document with the specified access mode.

**Parameters:**

(continued from last page)

id - a document identifier.

profile - a name of the profile where the document is moved.

mode - a document access mode.

**Returns:**

A Document instance.

**Throws:**

NullPointerException - if id is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while moving an object.
- Error occurred while opening an object.

IOException - if an I/O error occurs.

---

## deleteObject

```
public abstract void deleteObject(String id)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Deletes a document from the storage.

**Parameters:**

id - a document identifier.

**Throws:**

NullPointerException - if id is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while removing an object.

IOException - if an I/O error occurs.

---

## deleteObject

```
public abstract void deleteObject(String id,
    Map options)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

(continued from last page)

Deletes a document from the storage using the specified options.

Applicable option is [StorageConnector.OPTION\\_OBJECT\\_IDKIND](#).

#### Parameters:

id - a document identifier.

options - a map of object options from which desired object identifier kind can be obtained.

#### Throws:

`NullPointerException` - if id is a null reference.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while removing an object.

`IOException` - if an I/O error occurs.

---

## storeObject

```
public Document storeObject(Document source,
    String profile)
    throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Stores a source document on the storage profile with the source document default MIME type and returns a document opened with the source document access mode.

#### Parameters:

source - a source Document.

profile - a name of the profile where the document is stored.

#### Returns:

A Document opened in the source document access mode.

#### Throws:

`NullPointerException` - if source is a null reference.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating an object.
- Error copying stream contents.
- Error occurred while committing object changes.
- Error occurred while closing object.
- Error occurred while opening object.

`IOException` - if an I/O error occurs.

## storeObject

```
public abstract Document storeObject(Document source,
    String profile,
    String mime,
    DocumentAccess mode)
throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Stores a source document on the storage profile with the specified MIME type and returns a document opened with the specified access mode.

### Parameters:

source - a source Document.  
 profile - a name of the profile where the document is stored.  
 mime - a document MIME type or a null reference to use source default MIME type.  
 mode - a document access mode.

### Returns:

A Document opened in the specified access mode.

### Throws:

[NullPointerException](#) - if source is a null reference.  
[CancellationException](#) - if an operation was canceled by the user.  
[LoginException](#) - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
 IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Default MIME type is not available.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating an object.
- Error copying stream contents.
- Error occurred while committing object changes.
- Error occurred while closing object.
- Error occurred while opening object.

[IOException](#) - if an I/O error occurs.

## storeObject

```
public abstract Document storeObject(String source,
    String profile,
    String mime,
    DocumentAccess mode)
throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Stores a local file on the storage profile with the specified MIME type and returns a document opened with the specified access mode.

### Parameters:

source - a path to the source file.  
 profile - a name of the profile where the document is stored.  
 mime - a document MIME type or a null reference to use default MIME type resolved from file extension.  
 mode - a document access mode.



(continued from last page)

**Returns:**

A Document opened in the specified access mode.

**Throws:**

`NullPointerException` - if source is a null reference.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- File has no extension which is required if MIME type is not provided!
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating an object.
- Error copying stream contents.
- Error occurred while committing object changes.
- Error occurred while closing object.
- Error occurred while opening object.

`SecurityException` - if a security error is detected.

`IOException` - if an I/O error occurs.

---

## storeObject

```
public abstract Document storeObject(InputStream source,  
    String profile,  
    String mime,  
    DocumentAccess mode)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Stores `InputStream` contents on the storage profile with the specified MIME type and returns a document opened with the specified access mode.

**Parameters:**

source - a source `InputStream`.

profile - a name of the profile where the document is stored.

mime - a document MIME type.

mode - a document access mode.

**Returns:**

A Document opened in the specified access mode.

**Throws:**

`NullPointerException` - if source or mime is a null reference.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating an object.
- Error copying stream contents.
- Error occurred while committing object changes.
- Error occurred while closing object.
- Error occurred while opening object.

IOException - if an I/O error occurs.

---

## storeObject

```
public abstract String storeObject(Document source,  
    String profile,  
    String mime)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Stores a source document on the storage profile with the specified MIME type and returns a document identifier.

### Parameters:

source - a source Document.

profile - a name of the profile where the document is stored.

mime - a document MIME type or a null reference to use source default MIME type.

### Returns:

A document identifier.

### Throws:

NullPointerException - if source is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.  
IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Default MIME type is not available.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating an object.
- Error copying stream contents.
- Error occurred while committing object changes.
- Error occurred while closing object.

IOException - if an I/O error occurs.

---

(continued from last page)

## storeObject

```
public abstract String storeObject(String source,  
    String profile,  
    String mime)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Stores a local file on the storage profile with the specified MIME type and returns a document identifier.

### Parameters:

source - a path to the source file.

profile - a name of the profile where the document is stored.

mime - a document MIME type or a null reference to use default MIME type resolved from file extension.

### Returns:

A document identifier.

### Throws:

NullPointerException - if source is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- File has no extension which is required if MIME type is not provided.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating an object.
- Error copying stream contents.
- Error occurred while committing object changes.
- Error occurred while closing object.

SecurityException - if a security error is detected.

IOException - if an I/O error occurs.

---

## storeObject

```
public abstract String storeObject(InputStream source,  
    String profile,  
    String mime)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Stores InputStream contents on the storage profile with the specified MIME type and returns a document identifier.

### Parameters:

source - a source InputStream.

profile - a name of the profile where the document is stored.

mime - a document MIME type.

### Returns:

A document identifier.

### Throws:

NullPointerException - if source or mime is a null reference.

CancellationException - if an operation was canceled by the user.

(continued from last page)

`LoginException` - if a login error occurs.

[`StorageConnectorException`](#) - if an IMiS/Storage Connector error occurs.

IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating an object.
- Error copying stream contents.
- Error occurred while committing object changes.
- Error occurred while closing object.

`IOException` - if an I/O error occurs.

---

## retrieveObject

```
public abstract String retrieveObject(String id)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Retrieves a document from the storage to a temporary file.

### Parameters:

`id` - a document identifier.

### Returns:

A string containing the full path of the retrieved object.

### Throws:

`NullPointerException` - if `id` is a null reference.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[`StorageConnectorException`](#) - if an IMiS/Storage Connector error occurs.

IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening an object.
- Error copying stream contents.
- Error occurred while closing object.

`SecurityException` - if a security error is detected.

`IOException` - if an I/O error occurs.

---

## retrieveObject

```
public abstract String retrieveObject(String id,
    String target)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Retrieves a document from the storage to a specified local file.

### Parameters:

`id` - a document identifier.

(continued from last page)

target - a path of the retrieved document.

**Returns:**

A string containing [Document.getDefaultMime\(\)](#) of the retrieved document.

**Throws:**

NullPointerException - if id or target is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening an object.
- Error copying stream contents.
- Error occurred while closing object.

IOException - if an I/O error occurs.

---

## retrieveObject

```
public abstract String retrieveObject(String id,  
    OutputStream target)  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Retrieves a document from the storage and copies the contents to the specified OutputStream.

**Parameters:**

id - a document identifier.

target - a target OutputStream where the retrieved document contents are copied.

**Returns:**

A string containing [Document.getDefaultMime\(\)](#) of the retrieved document.

**Throws:**

NullPointerException - if id or target is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IMIS\_ARC storage errors:

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening an object.
- Error copying stream contents.
- Error occurred while closing object.

IOException - if an I/O error occurs.

(continued from last page)

## setExternalObjectId

```
public abstract void setExternalObjectId(String id,  
    String externalId)  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Sets the association of the specified external identifier with an object on the IMiS/ARC Storage Server.

### Parameters:

id - an object identifier.  
externalId - an external object identifier.

### Throws:

[NullPointerException](#) - if id or externalId is a null reference.  
[CancellationException](#) - if an operation was canceled by the user.  
[LoginException](#) - if a login error occurs.  
[StorageConnectorException](#) - if

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while associating external identifier with object.

[IOException](#) - if an I/O error occurs.

---

## resolveExternalObjectId

```
public abstract String resolveExternalObjectId(String externalId)  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Resolves the specified external identifier to the object identifier that external identifier is associated with using the object options.

### Parameters:

externalId - an external object identifier.

### Returns:

The object identifier that externalId is associated with.

### Throws:

[NullPointerException](#) - if externalId is a null reference.  
[CancellationException](#) - if an operation was canceled by the user.  
[LoginException](#) - if a login error occurs.  
[StorageConnectorException](#) - if

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while searching for external identifier association with an object.

[IOException](#) - if an I/O error occurs.

(continued from last page)

## resolveExternalObjectId

```
public abstract String resolveExternalObjectId(String externalId,  
    Map options)  
    throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Resolves the specified external identifier to the object identifier that external identifier is associated with using the specified options.

### Parameters:

`externalId` - an external object identifier.

`options` - a map of object options from which desired object identifier type and encoding can be obtained.

### Returns:

The object identifier that `externalId` is associated with.

### Throws:

`NullPointerException` - if `externalId` is a null reference.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if

- Operation on closed storage is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while searching for external identifier association with an object.

`IOException` - if an I/O error occurs.

---

## getProfileCapacity

```
public abstract long getProfileCapacity(String profile)  
    throws StorageConnectorException,  
    IOException
```

Gets the storage profile capacity, in bytes.

**Note:** This feature is currently not supported on IMIS\_ARC storage.

### Parameters:

`profile` - a name of the storage profile.

### Returns:

A storage profile capacity, in bytes.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

`IOException` - if an I/O error occurs.

---

## getProfileSpaceAvailable

```
public abstract long getProfileSpaceAvailable(String profile)  
    throws StorageConnectorException,  
    IOException
```

Gets the storage profile available space, in bytes.

**Note:** This feature is currently not supported on IMIS\_ARC storage.

(continued from last page)

**Parameters:**

profile - a name of the storage profile.

**Returns:**

A storage profile available space, in bytes.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

[IOException](#) - if an I/O error occurs.

---

## getProfileSpaceUsed

```
public abstract long getProfileSpaceUsed(String profile)
    throws StorageConnectorException,
           IOException
```

Gets the storage profile used space, in bytes.

**Note:** This feature is currently not supported on IMIS\_ARC storage.

**Parameters:**

profile - a name of the storage profile.

**Returns:**

A storage profile used space, in bytes.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

[IOException](#) - if an I/O error occurs.

---

## getProfileObjectCount

```
public abstract int getProfileObjectCount(String profile)
    throws StorageConnectorException,
           IOException
```

Gets the storage profile object count.

**Note:** This feature is currently not supported on IMIS\_ARC storage.

**Parameters:**

profile - a name of the storage profile.

**Returns:**

A number of documents on the storage profile.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

[IOException](#) - if an I/O error occurs.

---

## auditLogQuery

```
public abstract InputStream auditLogQuery(AuditQuery parameters)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Gets the audit log query results data input stream.



(continued from last page)

**Parameters:**

parameters - the audit log query parameters.

**Returns:**

The audit log query results data input stream.

**Throws:**

NullPointerException - if parameters is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed storage is not allowed.
- Error getting audit query results data input stream.

IOException - if an I/O error occurs.

---

## auditLogQuery

```
public abstract InputStream auditLogQuery(AuditQuery parameters,  
                                           Map options)  
throws javax.security.auth.login.LoginException,  
       StorageConnectorException,  
       IOException
```

Gets the audit log query results data input stream using the specified object options.

**Parameters:**

parameters - the audit log query parameters.

options - a map of object options from which desired object identifier type and encoding can be obtained.

**Returns:**

The audit log query results data input stream.

**Throws:**

NullPointerException - if parameters is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed storage is not allowed.
- Error getting audit query results data input stream.

IOException - if an I/O error occurs.

---

## close

```
public abstract void close()  
throws StorageConnectorException,  
       IOException
```

Closes the storage.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

IOException - if an I/O error occurs.

## com.imis.storageconnector

### Class StorageConnector

java.lang.Object

└-com.imis.storageconnector.StorageConnector

public class **StorageConnector**  
extends Object

Defines the class with basic IMiS/Storage Connector methods.

StorageConnector is a singleton class that allows only one instance of this class to be created and accessed.

A singleton instance can be acquired with static [getInstance\(\)](#) property. When the StorageConnector singleton is no longer in use call [freeInstance\(\)](#) method to insure proper IMiS/Storage Connector shutdown.

### Field Summary

public static final	<a href="#">DEFAULT_PAGE_COUNT</a> Default entity collection page count. Value: <b>8</b>
public static final	<a href="#">DEFAULT_PAGE_SIZE</a> Default entity collection page size. Value: <b>64</b>
public static final	<a href="#">MAX_OPTION_NAME_LENGTH</a> Maximum length of the authentication name options. Value: <b>1024</b>
public static final	<a href="#">OPTION_APPLICATION_NAME</a> IMIS_ARCHIVE archive and IMIS_ARC storage audit trail application name option. Value: <b>ApplicationName</b>
public static final	<a href="#">OPTION_ARCHIVE_ID</a> IMIS_ARCHIVE archive identifier option. Value: <b>ArchiveId</b>
public static final	<a href="#">OPTION_AUTH_CRYPTO_ALGORITHM</a> IMIS_ARCHIVE archive and IMIS_ARC storage authentication cryptographic algorithm option. Value: <b>AuthCryptoAlgorithm</b>
public static final	<a href="#">OPTION_AUTH_CRYPTO_CIPHERMODE</a> IMIS_ARCHIVE archive and IMIS_ARC storage authentication block cipher mode option. Value: <b>AuthCryptoCipherMode</b>
public static final	<a href="#">OPTION_AUTH_CRYPTO_KEYSIZE</a> IMIS_ARCHIVE archive and IMIS_ARC storage authentication key size option. Value: <b>AuthCryptoKeySize</b>
public static final	<a href="#">OPTION_AUTH_KEY</a> IMIS_ARC storage authentication key data option. Value: <b>AuthKey</b>

public static final	<a href="#">OPTION_AUTH_TYPE</a> IMIS_ARCHIVE archive and IMIS_ARC storage authentication key type option. Value: <b>AuthType</b>
public static final	<a href="#">OPTION_AUTH_TYPE_DISPOSITION</a> IMIS_ARCHIVE archive authentication type disposition option. Value: <b>AuthTypeDisposition</b>
public static final	<a href="#">OPTION_COMPUTER_NAME</a> IMIS_ARCHIVE archive and IMIS_ARC storage audit trail computer name option. Value: <b>ComputerName</b>
public static final	<a href="#">OPTION_CONTENT_PART_IDENCODING</a> IMIS_ARCHIVE archive content part identifier encoding disposition option. Value: <b>ContentPartIdEncoding</b>
public static final	<a href="#">OPTION_CONTENT_PART_IDTYPE</a> IMIS_ARCHIVE archive content part identifier type disposition option. Value: <b>ContentPartIdType</b>
public static final	<a href="#">OPTION_DEFERRED_COLLECTION_FILTERING</a> IMIS_ARCHIVE collection filtering disposition. Value: <b>DeferredCollectionFiltering</b>
public static final	<a href="#">OPTION_DISCOVERY_ARCHIVE_TYPES</a> IMIS_ARCHIVE archive discovery archive types option. Value: <b>DiscoveryArchiveTypes</b>
public static final	<a href="#">OPTION_DISCOVERY_HINT</a> IMIS_ARCHIVE archive discovery hint option. Value: <b>DiscoveryHint</b>
public static final	<a href="#">OPTION_ENTITY_IDENCODING</a> IMIS_ARCHIVE archive entity identifier encoding disposition option. Value: <b>EntityIdEncoding</b>
public static final	<a href="#">OPTION_ENTITY_IDTYPE</a> IMIS_ARCHIVE archive entity identifier type disposition option. Value: <b>EntityIdType</b>
public static final	<a href="#">OPTION_ENTITYCOLLECTION_PAGECOUNT</a> IMIS_ARCHIVE archive entity collection page count option. Value: <b>EntityCollectionPageCount</b>
public static final	<a href="#">OPTION_ENTITYCOLLECTION_PAGESIZE</a> IMIS_ARCHIVE archive entity collection page size option. Value: <b>EntityCollectionPageSize</b>
public static final	<a href="#">OPTION_IMPERSONALIZATION_USER_NAME</a> IMIS_ARCHIVE archive authentication impersonalization user name option. Value: <b>ImpersonalizationUserName</b>
public static final	<a href="#">OPTION_KERBEROS_KDC</a> IMIS_ARCHIVE archive Kerberos Key Distribution Center (KDC) option. Value: <b>KerberosKDC</b>

public static final	<a href="#">OPTION_KERBEROS_REALM</a> IMIS_ARCHIVE archive Kerberos Realm option. Value: <b>KerberosRealm</b>
public static final	<a href="#">OPTION_KERBEROS_SPN</a> IMIS_ARCHIVE archive Kerberos Service Principal Name (SPN) option. Value: <b>KerberosSPN</b>
public static final	<a href="#">OPTION_KERBEROS_TICKET</a> IMIS_ARCHIVE archive Kerberos Service Ticket option. Value: <b>KerberosTicket</b>
public static final	<a href="#">OPTION_LOCAL_ADDRESS</a> IMIS_ARCHIVE archive client local address option. Value: <b>LocalAddress</b>
public static final	<a href="#">OPTION_LOCAL_CERTIFICATE</a> IMIS_ARCHIVE archive local certificate option. Value: <b>LocalCertificate</b>
public static final	<a href="#">OPTION_LOCAL_CERTIFICATE_KEY</a> IMIS_ARCHIVE archive local certificate key option. Value: <b>LocalCertificate</b>
public static final	<a href="#">OPTION_MAX_IO_PAYLOAD_SIZE</a> IMIS_ARCHIVE archive maximum I/O payload size. Value: <b>MaxIOPayloadSize</b>
public static final	<a href="#">OPTION_MAX_SESSIONS_PER_USER</a> IMIS_ARCHIVE archive and IMIS_ARC storage maximum number of sessions per user option. Value: <b>MaxSessionsPerUser</b>
public static final	<a href="#">OPTION_NODES</a> IMIS_ARCHIVE archive and IMIS_ARC storage additional cluster nodes option. Value: <b>Nodes</b>
public static final	<a href="#">OPTION_OBJECT_IDENCODING</a> IMIS_ARC storage object identifier encoding disposition option. Value: <b>ObjectIdEncoding</b>
public static final	<a href="#">OPTION_OBJECT_IDKIND</a> IMIS_ARC storage object identifier kind disposition option. Value: <b>ObjectIdKind</b>
public static final	<a href="#">OPTION_OBJECT_IDTYPE</a> IMIS_ARC storage object identifier type disposition option. Value: <b>ObjectIdType</b>
public static final	<a href="#">OPTION_OUTBOUND_ADDRESS</a> IMIS_ARCHIVE archive and IMIS_ARC storage client outbound IPv4 address option. Value: <b>OutboundAddress</b>
public static final	<a href="#">OPTION_OUTBOUND_ADDRESS_V6</a> IMIS_ARCHIVE archive and IMIS_ARC storage client outbound IPv6 address option. Value: <b>OutboundAddressV6</b>

public static final	<a href="#">OPTION_PERSISTENT_SESSIONS</a> IMIS_ARCHIVE archive and IMIS_ARC storage persistent sessions option. Value: <b>PersistentSessions</b>
public static final	<a href="#">OPTION_USER_NAME</a> IMIS_ARCHIVE archive and IMIS_ARC storage audit trail user name option. Value: <b>UserName</b>
public static final	<a href="#">OPTION_USER_PASSWORD</a> IMIS_ARCHIVE archive user password option. Value: <b>UserPassword</b>
public static final	<a href="#">PROPERTY_AUDIT_LOG_REQUIRED</a> Specifies an IMIS_ARCHIVE archive property key for a boolean value indicating whether audit log message must be specified for operations that support audit log. Value: <b>AuditLogRequired</b>
public static final	<a href="#">PROPERTY_AUDIT_LOG_SUPPORTED</a> Specifies an IMIS_ARCHIVE archive property key for a boolean value indicating whether audit log is supported. Value: <b>AuditLogSupported</b>
public static final	<a href="#">PROPERTY_AUTHENTICATED_USER_ACCOUNT</a> Specifies an IMIS_ARCHIVE archive property key for a String value representing authenticated user account. Value: <b>AuthenticatedUserAccount</b>
public static final	<a href="#">PROPERTY_CAPABILITIES</a> Specifies an IMIS_ARCHIVE archive property key for a set of <a href="#">ArchiveCapabilities</a> values representing archive capabilities. Value: <b>Capabilities</b>
public static final	<a href="#">PROPERTY_CLASS_HIERARCHY_DELIMITER</a> Specifies an IMIS_ARCHIVE archive property key for a String value representing class entity classification code hierarchy delimiter. Value: <b>ClassHierarchyDelimiter</b>
public static final	<a href="#">PROPERTY_DOCUMENT_HIERARCHY_DELIMITER</a> Specifies an IMIS_ARCHIVE archive property key for a String value representing document entity classification code hierarchy delimiter. Value: <b>DocumentHierarchyDelimiter</b>
public static final	<a href="#">PROPERTY_FOLDER_HIERARCHY_DELIMITER</a> Specifies an IMIS_ARCHIVE archive property key for a String value representing folder entity classification code hierarchy delimiter. Value: <b>FolderHierarchyDelimiter</b>
public static final	<a href="#">PROPERTY_ROOT_EFFECTIVE_RIGHTS</a> Specifies an IMIS_ARCHIVE archive property key for a <a href="#">IReadOnlyEntityRights</a> value representing the archive root effective rights. Value: <b>RootEffectiveRights</b>
public static final	<a href="#">PROPERTY_SERVICE_NAME</a> Specifies an IMIS_ARCHIVE archive property key for a String value representing the name of the service. Value: <b>ServiceName</b>

public static final	<a href="#"><u>PROPERTY_SERVICE_VERSION</u></a> Specifies an IMIS_ARCHIVE archive property key for a long value representing the version number of the service. Value: <b>ServiceVersion</b>
public static final	<a href="#"><u>PROPERTY_TRAFFIC_ENCRYPTED</u></a> Specifies an IMIS_ARCHIVE archive property key for a boolean value indicating whether traffic between client and server is encrypted. Value: <b>TrafficEncrypted</b>

## Method Summary

void	<a href="#"><u>addObjectRef</u></a> (Object obj) Adds an object to the list of objects referenced by IMiS/Storage Connector.
void	<a href="#"><u>finalize</u></a> () Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.
static void	<a href="#"><u>freeInstance</u></a> () Terminates the singleton instance of the StorageConnector class.
static void	<a href="#"><u>fromBackEndAuthenticationOptions</u></a> (Map source, Map target) Converts the specified IMiS/ARC Client authentication options to the IMiS/Storage Connector authentication options.
static <a href="#"><u>AuthType</u></a>	<a href="#"><u>fromBackEndAuthenticationType</u></a> (com.imis.imisarc.client.AuthenticationType type) Converts the specified IMiS/ARC Client authentication type to the IMiS/Storage Connector authentication type.
<a href="#"><u>IContentTypeResolver</u></a>	<a href="#"><u>getContentTypeResolver</u></a> () Gets the content type resolver.
<a href="#"><u>IContentTypeResolver</u></a>	<a href="#"><u>getCustomContentTypeResolver</u></a> () Gets the custom content type resolver.
static <a href="#"><u>StorageConnector</u></a>	<a href="#"><u>getInstance</u></a> () Gets a singleton instance of the StorageConnector class.
String	<a href="#"><u>getVersionInfo</u></a> () Gets the version information as string value.
void	<a href="#"><u>logAddHandler</u></a> (Handler handler) Adds a log handler to the storage object.
Level	<a href="#"><u>logGetLevel</u></a> () Gets the logging level of the logger.
void	<a href="#"><u>logInternal</u></a> (boolean enabled) Enables or disables the IMiS/Storage Connector internal logging.
boolean	<a href="#"><u>logIsInternal</u></a> () Gets a value indicating whether the internal logging of the IMiS/Storage Connector is enabled or not.
void	<a href="#"><u>logRemoveHandler</u></a> (Handler handler) Removes a log handler associated with the storage object.

<a href="#">void</a>	<a href="#">logSetLevel</a> (Level level) Sets the logging level of the logger and associated log handlers.
<a href="#">void</a>	<a href="#">objectTransfer</a> ( <a href="#">IArchive</a> archive, <a href="#">EntityIdKind</a> sourceIdKind, List sourceIds, <a href="#">EntityIdKind</a> targetIdKind, String targetId, String targetContentName, List descriptions) Transfers source legacy objects to the specified content on the target entity.
<a href="#">IArchive</a>	<a href="#">openArchive</a> ( <a href="#">ArchiveType</a> type, String host, int port, Map options) Opens an archive from the host, specified by the type, host name or IP address and TCP port.
<a href="#">IArchive</a>	<a href="#">openArchive</a> ( <a href="#">ArchiveType</a> type, String host, Map options) Opens an archive from the host, specified by the type and host name or IP address and default port 16807.
<a href="#">Storage</a>	<a href="#">openContentManagerStorage</a> (String name, String username, String password) Opens a storage on IBM DB2 Content Manager specified by the name using a specified username and a password.
<a href="#">Storage</a>	<a href="#">openDocumentManagerStorage</a> () Opens a storage on IBM Lotus Domino Document Manager or Domino.Doc compliant Storage Server.
<a href="#">Storage</a>	<a href="#">openFileSystemStorage</a> (String rootDirectory) Opens a storage on Local/Remote File System specified by the root directory.
<a href="#">Storage</a>	<a href="#">openIMiSARCStorage</a> (String host) Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address and default port 16807.
<a href="#">Storage</a>	<a href="#">openIMiSARCStorage</a> (String host, int port) Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address and port.
<a href="#">Storage</a>	<a href="#">openIMiSARCStorage</a> (String host, int port, Map options) Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address, port and storage options.
<a href="#">Storage</a>	<a href="#">openIMiSARCStorage</a> (String host, int port, String profile) Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address, port and profile.
<a href="#">Storage</a>	<a href="#">openIMiSARCStorage</a> (String host, int port, String profile, Map options) Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address, port, profile and storage options.
<a href="#">Storage</a>	<a href="#">openIMiSARCStorage</a> (String host, Map options) Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address, default port 16807 and storage options.
<a href="#">Storage</a>	<a href="#">openIMiSARCStorage</a> (String host, String profile) Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address, default port 16807 and profile.
<a href="#">Storage</a>	<a href="#">openIMiSARCStorage</a> (String host, String profile, Map options) Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address, default port 16807, profile and storage options.

static String	<a href="#"><code>optionsToString</code></a> (Map options) Returns a string representation of an IMIS_ARC storage options, excluding authentication sensitive data.
static Set	<a href="#"><code>readOptionDiscoveryArchiveTypes</code></a> (Map options) Reads the <a href="#"><code>OPTION_DISCOVERY_ARCHIVETYPES</code></a> value from the specified options.
static String	<a href="#"><code>readOptionDiscoveryHint</code></a> (Map options) Reads the <a href="#"><code>OPTION_DISCOVERY_HINT</code></a> value from the specified options.
static int	<a href="#"><code>readOptionEntityCollectionPageCount</code></a> (Map options) Reads the <a href="#"><code>OPTION_ENTITYCOLLECTION_PAGECOUNT</code></a> value from the specified options.
static int	<a href="#"><code>readOptionEntityCollectionPageSize</code></a> (Map options) Reads the <a href="#"><code>OPTION_ENTITYCOLLECTION_PAGESIZE</code></a> value from the specified options.
void	<a href="#"><code>setCustomContentTypeResolver</code></a> ( <a href="#"><code>IContentTypeResolver</code></a> value) Sets the custom content type resolver.
static void	<a href="#"><code>toBackendAuthenticationOptions</code></a> (Map source, Map target) Converts the specified IMiS/Storage Connector authentication options to the IMiS/ARC Client authentication options.
static com.imis.imisarc.client.AuthenticationType	<a href="#"><code>toBackendAuthenticationType</code></a> ( <a href="#"><code>AuthType</code></a> type) Converts the specified IMiS/Storage Connector authentication type to the IMiS/ARC Client authentication type.

#### Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Fields

### OPTION\_PERSISTENT\_SESSIONS

```
public static final java.lang.String OPTION_PERSISTENT_SESSIONS
```



(continued from last page)

IMIS\_ARCHIVE archive and IMIS\_ARC storage persistent sessions option.

Specifies whether established sessions are to be preserved for subsequent usage or if they should be closed after each transaction.

The value of the persistent sessions option is specified as Boolean.

Default value is set to true.

The following code example demonstrates how to make storage sessions persistent.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_PERSISTENT_SESSIONS, Boolean.TRUE);

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **PersistentSessions**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)  
[openArchive\(ArchiveType, String, int, Map\)](#)  
[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_MAX\_SESSIONS\_PER\_USER

```
public static final java.lang.String OPTION_MAX_SESSIONS_PER_USER
```

(continued from last page)

IMIS\_ARCHIVE archive and IMIS\_ARC storage maximum number of sessions per user option.

Specifies the maximum number of sessions per user (i.e. number of sessions in context session pool).

The value of the maximum number of sessions per user option is specified as Integer.

Default value is set to 1.

The following code example demonstrates how to set more than one storage sessions per user.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_MAX_SESSIONS_PER_USER, new Integer(5));

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **MaxSessionsPerUser**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)  
[openArchive\(ArchiveType, String, int, Map\)](#)  
[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_NODES

```
public static final java.lang.String OPTION_NODES
```

(continued from last page)

IMIS\_ARCHIVE archive and IMIS\_ARC storage additional cluster nodes option.

Specifies the list of additional server nodes in the cluster of IMiS/ARC Storage Servers.

The value of the storage additional cluster nodes option can be specified in one of the following forms:

- String of hosts in hostname\_or\_ipaddress:port form separated by the semicolon (;),
- List of String hosts in hostname\_or\_ipaddress:port form or
- List of InetAddress hosts.

The following code example demonstrates how to specify additional server node.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_NODES, "iarc2.acme.com:16807");

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **Nodes**

#### See Also:

[openArchive\(ArchiveType, String, Map\)](#)  
[openArchive\(ArchiveType, String, int, Map\)](#)  
[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## MAX\_OPTION\_NAME\_LENGTH

public static final int **MAX\_OPTION\_NAME\_LENGTH**

Maximum length of the authentication name options.

Constant value: **1024**

#### See Also:

[OPTION\\_USER\\_NAME](#)  
[OPTION\\_COMPUTER\\_NAME](#)  
[OPTION\\_APPLICATION\\_NAME](#)

---

## OPTION\_USER\_NAME

public static final java.lang.String **OPTION\_USER\_NAME**

(continued from last page)

IMIS\_ARCHIVE archive and IMIS\_ARC storage audit trail user name option.

Specifies the user name used when establishing a new session with IMiS/ARC Storage Server.

The value of the user name option is specified as String.

User name can not be longer than [MAX\\_OPTION\\_NAME\\_LENGTH](#).

The following code example demonstrates how to set user name used when establishing a new session to IMiS/ARC Storage Server.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_USER_NAME, "Test User");

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **UserName**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)  
[openArchive\(ArchiveType, String, int, Map\)](#)  
[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_COMPUTER\_NAME

```
public static final java.lang.String OPTION_COMPUTER_NAME
```

(continued from last page)

IMIS\_ARCHIVE archive and IMIS\_ARC storage audit trail computer name option.

Specifies the computer name used when establishing a new session with IMiS/ARC Storage Server. If no computer name is specified, it is implicitly set to `InetAddress.getLocalHost().getHostName()`.

The value of the computer name option is specified as `String`.

Computer name can not be longer than [MAX\\_OPTION\\_NAME\\_LENGTH](#).

The following code example demonstrates how to set computer name used when establishing a new session to IMiS/ARC Storage Server.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_COMPUTER_NAME, "Test Computer");

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **ComputerName**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)  
[openArchive\(ArchiveType, String, int, Map\)](#)  
[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_APPLICATION\_NAME

```
public static final java.lang.String OPTION_APPLICATION_NAME
```

(continued from last page)

IMIS\_ARCHIVE archive and IMIS\_ARC storage audit trail application name option.

Specifies the application name used when establishing a new session with IMiS/ARC Storage Server. If no computer name is specified, it is implicitly set to "IMiS/Storage Connector + version".

The value of the application name option is specified as String.

Authentication application name can not be longer than [MAX\\_OPTION\\_NAME\\_LENGTH](#).

The following code example demonstrates how to set audit trail application name used when establishing a new session to IMiS/ARC Storage Server.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_APPLICATION_NAME, "Test Application");

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **ApplicationName**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)  
[openArchive\(ArchiveType, String, int, Map\)](#)  
[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_LOCAL\_ADDRESS

```
public static final java.lang.String OPTION_LOCAL_ADDRESS
```

(continued from last page)

IMIS\_ARCHIVE archive client local address option.

Specifies the local address from where the user is establishing a new session with IMiS/ARC Storage Server.

The value of the archive client local address option can be specified as `InetAddress` or `String`.

When the value of the archive client local address option equals to any local address (e.g. "0.0.0.0" for IPv4 or "0:0:0:0:0:0:0:0" for IPv6) session uses `Socket.getLocalAddress()` as client local address.

The following code example demonstrates how to set client local address used when establishing a new session to IMiS/ARC Storage Server.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_LOCAL_ADDRESS,
InetAddress.getByName("127.0.0.1"));

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **LocalAddress**

**See Also:**

[`openArchive\(ArchiveType, String, Map\)`](#)

[`openArchive\(ArchiveType, String, int, Map\)`](#)

---

## OPTION\_AUTH\_TYPE

```
public static final java.lang.String OPTION_AUTH_TYPE
```

(continued from last page)

IMIS\_ARCHIVE archive and IMIS\_ARC storage authentication key type option.

Specifies the type of the cryptographic key used when establishing a connection to IMiS/ARC Storage Server.

The value of the storage authentication key type option is specified as one of [AuthType](#) enum value.

The following code example demonstrates how to set pre-shared key authentication type to be used when establishing a connection to IMiS/ARC Storage Server.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_AUTH_TYPE, AuthType.PRE_SHARED_KEY);

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **AuthType**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)  
[openArchive\(ArchiveType, String, int, Map\)](#)  
[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_AUTH\_KEY

```
public static final java.lang.String OPTION_AUTH_KEY
```



(continued from last page)

IMIS\_ARC storage authentication key data option.

Specifies the authentication key used when establishing a connection to IMiS/ARC Storage Server.

The value of the storage authentication key data option can be specified as `String` or `byte[]`.

Used only when [OPTION\\_AUTH\\_TYPE](#) is specified as `AuthType.PRE_SHARED_KEY`.

The following code example demonstrates how to set an authentication key used with pre-shared key authentication.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_AUTH_TYPE, AuthType.PRE_SHARED_KEY);
options.put(StorageConnector.OPTION_AUTH_KEY, "psk");

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **AuthKey**

**See Also:**

[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_USER\_PASSWORD

```
public static final java.lang.String OPTION_USER_PASSWORD
```

(continued from last page)

IMIS\_ARCHIVE archive user password option.

Specifies the authentication user password used when establishing a new session with IMiS/ARC Storage Server.

The value of the archive user password option is specified as `string`.

Used only when [OPTION\\_AUTH\\_TYPE](#) is specified as `AuthType.LOCAL_CREDENTIALS`.

The following code example demonstrates how to set user password.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_AUTH_TYPE, AuthType.LOCAL_CREDENTIALS);
options.put(StorageConnector.OPTION_USER_PASSWORD, "password");

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **UserPassword**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_AUTH\_CRYPTO\_KEYSIZE

```
public static final java.lang.String OPTION_AUTH_CRYPTO_KEYSIZE
```

(continued from last page)

IMIS\_ARCHIVE archive and IMIS\_ARC storage authentication key size option.

Specifies the size of the cryptographic key used when establishing a connection to IMiS/ARC Storage Server. If no cryptographic key size is specified, it is implicitly set to `AuthCryptoKeySize.KS128`.

The value of the storage authentication key size option is specified as one of [AuthCryptoKeySize](#) enum value.

Used only when [OPTION\\_AUTH\\_TYPE](#) is specified.

The following code example demonstrates how to set 256 bit cryptographic key used with pre-shared key authentication.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_AUTH_TYPE, AuthType.PRE_SHARED_KEY);
options.put(StorageConnector.OPTION_AUTH_CRYPTOKEYSIZE, AuthCryptoKeySize.KS256);

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **AuthCryptoKeySize**

#### See Also:

[openArchive\(ArchiveType, String, Map\)](#)  
[openArchive\(ArchiveType, String, int, Map\)](#)  
[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_AUTH\_CRYPTOKEYSIZE

```
public static final java.lang.String OPTION_AUTH_CRYPTOKEYSIZE
```

(continued from last page)

IMIS\_ARCHIVE archive and IMIS\_ARC storage authentication cryptographic algorithm option.

Specifies the type of the cryptographic algorithm used when establishing a connection to IMiS/ARC Storage Server. If no cryptographic algorithm is specified, it is implicitly set to `AuthCryptoAlgorithm.AES`.

The value of the storage authentication cryptographic algorithm option is specified as one of [AuthCryptoAlgorithm](#) enum value.

Used only when [OPTION\\_AUTH\\_TYPE](#) is specified.

The following code example demonstrates how to set AES cryptographic algorithm used with pre-shared key authentication.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_AUTH_TYPE, AuthType.PRE_SHARED_KEY);
options.put(StorageConnector.OPTION_AUTH_CRYPTO_ALGORITHM, AuthCryptoAlgorithm.AES);

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **AuthCryptoAlgorithm**

#### See Also:

[openArchive\(ArchiveType, String, Map\)](#)  
[openArchive\(ArchiveType, String, int, Map\)](#)  
[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_AUTH\_CRYPTO\_CIPHERMODE

```
public static final java.lang.String OPTION_AUTH_CRYPTO_CIPHERMODE
```

(continued from last page)

IMIS\_ARCHIVE archive and IMIS\_ARC storage authentication block cipher mode option.

Specifies the block cipher mode used when establishing a connection to IMiS/ARC Storage Server. If no block cipher is specified, it is implicitly set to `AuthCryptoCipherMode.CBC`.

The value of the storage authentication block cipher mode option is specified as one of [AuthCryptoCipherMode](#) enum value.

Used only when [OPTION\\_AUTH\\_TYPE](#) is specified.

The following code example demonstrates how to set CBC block cipher mode used with pre-shared key authentication.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_AUTH_TYPE, AuthType.PRE_SHARED_KEY);
options.put(StorageConnector.OPTION_AUTH_CRYPTOCIPHERMODE,
AuthCryptoCipherMode.CBC);

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **AuthCryptoCipherMode**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)  
[openArchive\(ArchiveType, String, int, Map\)](#)  
[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_OBJECT\_IDTYPE

```
public static final java.lang.String OPTION_OBJECT_IDTYPE
```

(continued from last page)

IMIS\_ARC storage object identifier type disposition option.

Specifies the object identifier type disposition for the newly created objects. If no object identifier type disposition is specified, it is implicitly set to `ObjectIdType.ID256`.

The value of the storage object identifier type disposition option is specified as one of [ObjectIdType](#) enum value.

The following code example demonstrates how to set 256 bit long object identifiers when creating new objects.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_OBJECT_IDTYPE, ObjectIdType.ID256);

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **ObjectIdType**

**See Also:**

[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_OBJECT\_IDENCODING

```
public static final java.lang.String OPTION_OBJECT_IDENCODING
```

(continued from last page)

IMIS\_ARC storage object identifier encoding disposition option.

Specifies the object identifier encoding disposition for the newly created objects. If no object identifier encoding disposition is specified, it is implicitly set to `ObjectIdEncoding.BASE16`.

The value of the storage object identifier encoding disposition option is specified as one of [ObjectIdEncoding](#) enum value.

The following code example demonstrates how to set base 64 encoding for object identifiers when creating new objects.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_OBJECT_IDENCODING, ObjectIdEncoding.BASE16);

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **ObjectIdEncoding**

**See Also:**

[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_OBJECT\_IDKIND

```
public static final java.lang.String OPTION_OBJECT_IDKIND
```

(continued from last page)

IMIS\_ARC storage object identifier kind disposition option.

Specifies the object identifier kind disposition for handling existing objects. If no object identifier kind disposition is specified, it is implicitly set to `ObjectIdKind.INTERNAL`.

The value of the storage object identifier kind disposition option is specified as one of [ObjectIdKind](#) enum value.

The following code example demonstrates how to set external object identifier kind disposition when opening new objects.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for storage on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_OBJECT_IDKIND, ObjectIdKind.EXTERNAL);

// Open storage on IMiS/ARC Storage Server
Storage stg = sc.openIMiSARCStorage(host, port, options);
```

Constant value: **ObjectIdKind**

**See Also:**

[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_ENTITY\_IDTYPE

```
public static final java.lang.String OPTION_ENTITY_IDTYPE
```



(continued from last page)

IMIS\_ARCHIVE archive entity identifier type disposition option.

Specifies the entity identifier type disposition for the newly created entities. If no entity identifier type disposition is specified, it is implicitly set to `EntityIdType.ID256`.

The value of the archive entity identifier type disposition option is specified as one of [EntityIdType](#) enum value.

The following code example demonstrates how to set 256 bit long entity identifiers when creating new entities.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_ENTITY_IDTYPE, EntityIdType.ID256);

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **EntityIdType**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_ENTITY\_IDENCODING

```
public static final java.lang.String OPTION_ENTITY_IDENCODING
```

(continued from last page)

IMIS\_ARCHIVE archive entity identifier encoding disposition option.

Specifies the entity identifier encoding disposition for the newly created entities. If no entity identifier encoding disposition is specified, it is implicitly set to `EntityIdEncoding.BASE16`.

The value of the archive entity identifier encoding disposition option is specified as one of [EntityIdEncoding](#) enum value.

The following code example demonstrates how to set base 64 encoding for entity identifiers when creating new entities.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_ENTITY_IDENCODING, EntityIdEncoding.BASE16);

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **EntityIdEncoding**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_CONTENT\_PART\_IDTYPE

```
public static final java.lang.String OPTION_CONTENT_PART_IDTYPE
```

(continued from last page)

IMIS\_ARCHIVE archive content part identifier type disposition option.

Specifies the content part identifier type disposition for the newly created content parts. If no content part identifier type disposition is specified, it is implicitly set to `ContentPartIdType.ID256`.

The value of the archive content part identifier type disposition option is specified as one of [ContentPartIdType](#) enum value.

The following code example demonstrates how to set 256 bit long content part identifiers when creating new content parts.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_CONTENT_PART_IDTYPE, ContentPartIdType.ID256);

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **ContentPartIdType**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_CONTENT\_PART\_IDENCODING

```
public static final java.lang.String OPTION_CONTENT_PART_IDENCODING
```

(continued from last page)

IMIS\_ARCHIVE archive content part identifier encoding disposition option.

Specifies the content part identifier encoding disposition for the newly created content parts. If no content part identifier encoding disposition is specified, it is implicitly set to `ContentPartIdEncoding.BASE16`.

The value of the archive content part identifier encoding disposition option is specified as one of [ContentPartIdEncoding](#) enum value.

The following code example demonstrates how to set base 64 encoding for content part identifiers when creating new content parts.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_CONTENT_PART_IDENCODING,
ContentPartIdEncoding.BASE16);

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **ContentPartIdEncoding**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_ENTITYCOLLECTION\_PAGESIZE

```
public static final java.lang.String OPTION_ENTITYCOLLECTION_PAGESIZE
```

(continued from last page)

IMIS\_ARCHIVE archive entity collection page size option.

Specifies the page size of entity collection. If no entity collection page size is specified, it is implicitly set to 64.

The value of the archive entity collection page size option is specified as positive int.

The following code example demonstrates how to set entity collection page size.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_ENTITYCOLLECTION_PAGESIZE, 64);

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **EntityCollectionPageSize**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## DEFAULT\_PAGE\_SIZE

```
public static final int DEFAULT_PAGE_SIZE
```

Default entity collection page size.

Constant value: **64**

---

## OPTION\_ENTITYCOLLECTION\_PAGECOUNT

```
public static final java.lang.String OPTION_ENTITYCOLLECTION_PAGECOUNT
```

(continued from last page)

IMIS\_ARCHIVE archive entity collection page count option.

Specifies the page count of entity collection. If no entity collection page count is specified, it is implicitly set to 8.

The value of the archive entity collection page count option is specified as positive int.

The following code example demonstrates how to set entity collection page count.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_ENTITYCOLLECTION_PAGECOUNT, 8);

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **EntityCollectionPageCount**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## DEFAULT\_PAGE\_COUNT

```
public static final int DEFAULT_PAGE_COUNT
```

Default entity collection page count.

Constant value: **8**

---

## OPTION\_ARCHIVE\_ID

```
public static final java.lang.String OPTION_ARCHIVE_ID
```

(continued from last page)

IMIS\_ARCHIVE archive identifier option.

Specifies the archive identifier. If no archive identifier is specified, it is implicitly set to 0.

The value of the archive identifier option is specified as `int` or `String`.

The following code example demonstrates how to set archive identifier.

```
Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarc.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_ARCHIVE_ID, 1);

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **ArchiveId**

**See Also:**

[`openArchive\(ArchiveType, String, Map\)`](#)

[`openArchive\(ArchiveType, String, int, Map\)`](#)

---

## OPTION\_DISCOVERY\_HINT

```
public static final java.lang.String OPTION_DISCOVERY_HINT
```

IMIS\_ARCHIVE archive discovery hint option.

Specifies the discovery hint representing IMiS/ARC Storage Server host string. If none specified, archive discovery is disabled.

The value of the archive discovery hint option is specified as `String`.

The following code example demonstrates how to set discovery hint.

```
// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_DISCOVERY_HINT, "iarc.acme.com:16807");

// Find all archives on IMiS/ARC Storage Server
Collection<IArchiveDescriptor> archives = Discovery.findArchives(options);
```

Constant value: **DiscoveryHint**

(continued from last page)

**See Also:**[openArchive\(ArchiveType, String, Map\)](#)[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_DISCOVERY\_ARCHIVETYPES

public static final java.lang.String **OPTION\_DISCOVERY\_ARCHIVETYPES**

IMIS\_ARCHIVE archive discovery archive types option.

Specifies the discovery archive types. If no archive type is specified, it is implicitly set to `EnumSet.allOf(ArchiveType.class)`.

The value of the archive discovery archive types option is specified as one of [ArchiveType](#) enum value.

The following code example demonstrates how to set discovery hint.

```
// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_DISCOVERY_ARCHIVETYPES,
ArchiveType.IMIS_ARCHIVE);

// Find all archives on IMiS/ARC Storage Server
Collection<IArchiveDescriptor> archives = Discovery.findArchives(options);
```

Constant value: **DiscoveryArchiveTypes**

**See Also:**[openArchive\(ArchiveType, String, Map\)](#)[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_AUTH\_TYPE\_DISPOSITION

public static final java.lang.String **OPTION\_AUTH\_TYPE\_DISPOSITION**



(continued from last page)

IMIS\_ARCHIVE archive authentication type disposition option.

Specifies the authentication type disposition. If no authentication type disposition is specified, it is implicitly set to:

- `AuthType.KERBEROS5_SERVICE_TICKET`
- `AuthType.LDAP_CREDENTIALS`
- `AuthType.LOCAL_CREDENTIALS`

The value of the authentication type disposition option is specified as a collection of distinct [AuthType](#) enum values.

The following code example demonstrates how to set authentication type disposition.

```
// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
List<AuthType> disposition = new ArrayList<AuthType>(new AuthType[] {
    AuthType.LDAP_CREDENTIALS,
    AuthType.LOCAL_CREDENTIALS
});
options.put(StorageConnector.OPTION_AUTH_TYPE_DISPOSITION, disposition);

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **AuthTypeDisposition**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_KERBEROS\_KDC

```
public static final java.lang.String OPTION_KERBEROS_KDC
```

(continued from last page)

IMIS\_ARCHIVE archive Kerberos Key Distribution Center (KDC) option.

Specifies the Kerberos KDC used when establishing a new session with IMiS/ARC Storage Server.

The value of the archive Kerberos KDC option is specified as `String`.

**Note:** Used only when [OPTION\\_AUTH\\_TYPE](#) is specified as `AuthType.KERBEROS5_SERVICE_TICKET`.

The following code example demonstrates how to set Kerberos KDC.

```
// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_KERBEROS_KDC, "kdc.acme.com");

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **KerberosKDC**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_KERBEROS\_SPN

```
public static final java.lang.String OPTION_KERBEROS_SPN
```

IMIS\_ARCHIVE archive Kerberos Service Principal Name (SPN) option.

Specifies the Kerberos SPN used when establishing a new session with IMiS/ARC Storage Server.

The value of the archive Kerberos SPN option is specified as `String`.

**Note:** Used only when [OPTION\\_AUTH\\_TYPE](#) is specified as `AuthType.KERBEROS5_SERVICE_TICKET`.

The following code example demonstrates how to set Kerberos SPN.

```
// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_KERBEROS_SPN, "IARC/iarc.acme.com@ACME.COM");

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **KerberosSPN**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

## OPTION\_KERBEROS\_REALM

```
public static final java.lang.String OPTION_KERBEROS_REALM
```

IMIS\_ARCHIVE archive Kerberos Realm option.

Specifies the Kerberos Realm used when establishing a new session with IMiS/ARC Storage Server.

The value of the archive Kerberos Realm option is specified as `String`.

**Note:** Used only when [OPTION\\_AUTH\\_TYPE](#) is specified as `AuthType.KERBEROS5_SERVICE_TICKET`.

The following code example demonstrates how to set Kerberos Realm.

```
// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.Add(StorageConnector.OPTION_KERBEROS_REALM, "ACME.COM");

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **KerberosRealm**

### See Also:

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_KERBEROS\_TICKET

```
public static final java.lang.String OPTION_KERBEROS_TICKET
```

(continued from last page)

IMIS\_ARCHIVE archive Kerberos Service Ticket option.

Specifies the Kerberos Service Ticket used when establishing a new session with IMiS/ARC Storage Server.

The value of the Kerberos Service Ticket option is specified as `byte[]`.

**Note:** Used only when [OPTION\\_AUTH\\_TYPE](#) is specified as `AuthType.KERBEROS5_SERVICE_TICKET`.

The following code example demonstrates how to set Kerberos Service Ticket.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarc.acme.com";
int port = 16807;

byte[] ticket = new byte[] { ... };

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_KERBEROS_TICKET, ticket);

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **KerberosTicket**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_LOCAL\_CERTIFICATE

```
public static final java.lang.String OPTION_LOCAL_CERTIFICATE
```

(continued from last page)

IMIS\_ARCHIVE archive local certificate option.

Specifies the local certificate used when establishing a new session with IMiS/ARC Storage Server.

The value of the local certificate option is specified as X509Certificate.

The following code example demonstrates how to set local certificate.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarc.acme.com";
int port = 16807;

X509Certificate cert = ...;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_LOCAL_CERTIFICATE, cert);

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **LocalCertificate**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_LOCAL\_CERTIFICATE\_KEY

```
public static final java.lang.String OPTION_LOCAL_CERTIFICATE_KEY
```

(continued from last page)

IMIS\_ARCHIVE archive local certificate key option.

Specifies the local certificate key used when establishing a new session with IMiS/ARC Storage Server.

The value of the local certificate key option is specified as `PrivateKey`.

The following code example demonstrates how to set local certificate key.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarc.acme.com";
int port = 16807;

PrivateKey key = ...;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_LOCAL_CERTIFICATE_KEY, key);

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **LocalCertificate**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_OUTBOUND\_ADDRESS

```
public static final java.lang.String OPTION_OUTBOUND_ADDRESS
```

(continued from last page)

IMIS\_ARCHIVE archive and IMIS\_ARC storage client outbound IPv4 address option.

Specifies the outbound IPv4 address used when establishing a new session with IMiS/ARC Storage Server.

The value of the archive client outbound IPv4 address option can be specified as `Inet4Address` or `String`.

The following code example demonstrates how to set client outbound IPv4 address used when establishing a new session to IMiS/ARC Storage Server.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_OUTBOUND_ADDRESS,
InetAddress.getByName("192.168.2.1"));

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **OutboundAddress**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)  
[openArchive\(ArchiveType, String, int, Map\)](#)  
[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_OUTBOUND\_ADDRESS\_V6

```
public static final java.lang.String OPTION_OUTBOUND_ADDRESS_V6
```

(continued from last page)

IMIS\_ARCHIVE archive and IMIS\_ARC storage client outbound IPv6 address option.

Specifies the outbound IPv6 address used when establishing a new session with IMiS/ARC Storage Server.

The value of the archive client outbound IPv6 address option can be specified as `Inet6Address` or `String`.

The following code example demonstrates how to set client outbound IPv6 address used when establishing a new session to IMiS/ARC Storage Server.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_OUTBOUND_ADDRESS_V6,
InetAddress.getByName("0000:0000:0000:0000:0000:FFFF:C0A8:0201"));

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **OutboundAddressV6**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)  
[openArchive\(ArchiveType, String, int, Map\)](#)  
[openIMiSARCStorage\(String, Map\)](#)  
[openIMiSARCStorage\(String, String, Map\)](#)  
[openIMiSARCStorage\(String, int, Map\)](#)  
[openIMiSARCStorage\(String, int, String, Map\)](#)

---

## OPTION\_IMPERSONALIZATION\_USER\_NAME

```
public static final java.lang.String OPTION_IMPERSONALIZATION_USER_NAME
```



(continued from last page)

IMIS\_ARCHIVE archive authentication impersonalization user name option.

Specifies the impersonalization user name used when establishing a new session with IMiS/ARC Storage Server.

The value of the impersonalization user name option is specified as `String`.

Impersonalization user name can not be longer than [MAX\\_OPTION\\_NAME\\_LENGTH](#).

This option is applicable on server version 9.9 or above.

The following code example demonstrates how to set impersonalization user name used when establishing a new session to IMiS/ARC Storage Server.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_IMPERSONALIZATION_USER_NAME, "Test Impersonate
User");

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **ImpersonalizationUserName**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_MAX\_IO\_PAYLOAD\_SIZE

```
public static final java.lang.String OPTION_MAX_IO_PAYLOAD_SIZE
```

(continued from last page)

IMIS\_ARCHIVE archive maximum I/O payload size.

Specifies the requested maximum I/O usable payload size used in communicating with IMiS/ARC Storage Server. This value can be adjusted by the server in case server doesn't allow values as high as requested.

This option is applicable on server version 9.8 or above.

The following code example demonstrates how to set the maximum size of I/O packet used in communicating with IMiS/ARC Storage Server.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_MAX_IO_PAYLOAD_SIZE, 262144);

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **MaxIOPayloadSize**

**See Also:**

[openArchive\(ArchiveType, String, Map\)](#)

[openArchive\(ArchiveType, String, int, Map\)](#)

---

## OPTION\_DEFERRED\_COLLECTION\_FILTERING

```
public static final java.lang.String OPTION_DEFERRED_COLLECTION_FILTERING
```

(continued from last page)

IMIS\_ARCHIVE collection filtering disposition.

Specifies how the IMiS/ARC Storage Server should filter its entity collections. Entity collections can be pre-filtered (filtering is done at collection creation stage) or elements can be filtered while they are being read. This option significantly impacts how fast large entity collections can be accessed. While it's good to have filtered collections available as soon as possible, the caller might wait longer than necessary to be able to access the first element since filtering also processes elements which might never be accessed. Deferred filtering instructs the server to delay filtering to element read stage and filter elements while they are being read. These collections are more responsive and caller can access first elements faster. The down side of this is that collection element count might not be the same as accessible element count so the code must be adapted to compensate this behaviour. In general collection iteration stops at last available element while accessing elements by their indexes might return null results if the index is beyond accessible element count.

The value of deferred collection filtering option is specified as Boolean.

Default value is set to false.

This option is applicable on server version 9.10 or above.

The following code example demonstrates how to enable deferred collection filtering.

```
// Get an instance of the IMiS/Storage Connector
StorageConnector sc = StorageConnector.getInstance();

// Set IMiS/ARC Storage Server host & port
String host = "iarcl.acme.com";
int port = 16807;

// Set options for archive on IMiS/ARC Storage Server
Map options = new TreeMap();
options.put(StorageConnector.OPTION_DEFERRED_COLLECTION_FILTERING, Boolean.TRUE);

// Open archive on IMiS/ARC Storage Server
IArchive archive = sc.openArchive(ArchiveType.IMIS_ARCHIVE, host, port, options);
```

Constant value: **DeferredCollectionFiltering**

See Also:

[openArchive\(ArchiveType, String, Map\)](#)  
[openArchive\(ArchiveType, String, int, Map\)](#)

---

## PROPERTY\_AUDIT\_LOG\_REQUIRED

```
public static final java.lang.String PROPERTY_AUDIT_LOG_REQUIRED
```

Specifies an IMIS\_ARCHIVE archive property key for a boolean value indicating whether audit log message must be specified for operations that support audit log.

Constant value: **AuditLogRequired**

---

## PROPERTY\_AUDIT\_LOG\_SUPPORTED

```
public static final java.lang.String PROPERTY_AUDIT_LOG_SUPPORTED
```

Specifies an IMIS\_ARCHIVE archive property key for a boolean value indicating whether audit log is supported.

Constant value: **AuditLogSupported**

---

(continued from last page)

---

## PROPERTY\_CAPABILITIES

```
public static final java.lang.String PROPERTY_CAPABILITIES
```

Specifies an IMIS\_ARCHIVE archive property key for a set of [ArchiveCapabilities](#) values representing archive capabilities.

Constant value: **Capabilities**

---

## PROPERTY\_CLASS\_HIERARCHY\_DELIMITER

```
public static final java.lang.String PROPERTY_CLASS_HIERARCHY_DELIMITER
```

Specifies an IMIS\_ARCHIVE archive property key for a String value representing class entity classification code hierarchy delimiter.

Constant value: **ClassHierarchyDelimiter**

---

## PROPERTY\_DOCUMENT\_HIERARCHY\_DELIMITER

```
public static final java.lang.String PROPERTY_DOCUMENT_HIERARCHY_DELIMITER
```

Specifies an IMIS\_ARCHIVE archive property key for a String value representing document entity classification code hierarchy delimiter.

Constant value: **DocumentHierarchyDelimiter**

---

## PROPERTY\_FOLDER\_HIERARCHY\_DELIMITER

```
public static final java.lang.String PROPERTY_FOLDER_HIERARCHY_DELIMITER
```

Specifies an IMIS\_ARCHIVE archive property key for a String value representing folder entity classification code hierarchy delimiter.

Constant value: **FolderHierarchyDelimiter**

---

## PROPERTY\_ROOT\_EFFECTIVE\_RIGHTS

```
public static final java.lang.String PROPERTY_ROOT_EFFECTIVE_RIGHTS
```

Specifies an IMIS\_ARCHIVE archive property key for a [IReadOnlyEntityRights](#) value representing the archive root effective rights.

Constant value: **RootEffectiveRights**

---

## PROPERTY\_SERVICE\_NAME

```
public static final java.lang.String PROPERTY_SERVICE_NAME
```

Specifies an IMIS\_ARCHIVE archive property key for a String value representing the name of the service.

Constant value: **ServiceName**

See Also:

[PROPERTY\\_SERVICE\\_VERSION](#)

---

## PROPERTY\_SERVICE\_VERSION

```
public static final java.lang.String PROPERTY_SERVICE_VERSION
```

(continued from last page)

Specifies an IMIS\_ARCHIVE archive property key for a long value representing the version number of the service. Version number equals IMiS/ARChive Storage Server build number from server version 9.9 or above.

Version	Name
460	IMiS/ARChive Storage Server 6.x
530	IMiS/ARChive Storage Server 7.x
580	IMiS/ARChive Storage Server 8.x
590	IMiS/ARChive Storage Server 9.x
670	IMiS/ARChive Storage Server 9.7.x
685	IMiS/ARChive Storage Server 9.8.x
690	IMiS/ARChive Storage Server 9.9.1810.690

Constant value: **ServiceVersion**

## PROPERTY\_TRAFFIC\_ENCRYPTED

```
public static final java.lang.String PROPERTY_TRAFFIC_ENCRYPTED
```

Specifies an IMIS\_ARCHIVE archive property key for a boolean value indicating whether traffic between client and server is encrypted.

Constant value: **TrafficEncrypted**

## PROPERTY\_AUTHENTICATED\_USER\_ACCOUNT

```
public static final java.lang.String PROPERTY_AUTHENTICATED_USER_ACCOUNT
```

Specifies an IMIS\_ARCHIVE archive property key for a String value representing authenticated user account.

Constant value: **AuthenticatedUserAccount**

## Methods

### getInstance

```
public static StorageConnector getInstance()
```

Gets a singleton instance of the StorageConnector class.

#### Returns:

A StorageConnector instance.

### freeInstance

```
public static void freeInstance()
```

Terminates the singleton instance of the StorageConnector class.

This method must be called when StorageConnector singleton instance is not used any more.

#### See Also:

[getInstance\(\)](#)

### finalize

```
protected void finalize()
    throws Throwable
```

(continued from last page)

Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.

Any exception thrown by the finalize method causes the finalization of this object to be halted, but is otherwise ignored.

**Throws:**

Throwable - if an error or exception occurs.

---

## optionsToString

```
public static String optionsToString(Map options)
```

Returns a string representation of an IMIS\_ARC storage options, excluding authentication sensitive data.

**Parameters:**

options - an IMIS\_ARC storage options.

**Returns:**

a string containing the key-value pairs.

---

## readOptionEntityCollectionPageSize

```
public static int readOptionEntityCollectionPageSize(Map options)
```

Reads the [OPTION\\_ENTITYCOLLECTION\\_PAGESIZE](#) value from the specified options.

**Parameters:**

options - the map of options.

**Returns:**

The entity collection page size.

---

## readOptionEntityCollectionPageCount

```
public static int readOptionEntityCollectionPageCount(Map options)
```

Reads the [OPTION\\_ENTITYCOLLECTION\\_PAGECOUNT](#) value from the specified options.

**Parameters:**

options - the map of options.

**Returns:**

The entity collection page size.

---

## readOptionDiscoveryHint

```
public static String readOptionDiscoveryHint(Map options)
```

Reads the [OPTION\\_DISCOVERY\\_HINT](#) value from the specified options.

**Parameters:**

options - the map of options.

**Returns:**

The archive discovery hint.

---

## readOptionDiscoveryArchiveTypes

```
public static Set readOptionDiscoveryArchiveTypes(Map options)
```

(continued from last page)

Reads the [OPTION\\_DISCOVERY\\_ARCHIVETYPES](#) value from the specified options.

**Parameters:**

options - the map of options.

**Returns:**

The discovery archive types.

---

## fromBackendAuthenticationType

```
public static AuthType  
fromBackendAuthenticationType(com.imis.imisarc.client.AuthenticationType type)
```

Converts the specified IMiS/ARC Client authentication type to the IMiS/Storage Connector authentication type.

**Parameters:**

type - the IMiS/ARC Client authentication type.

**Returns:**

The authentication type.

**Throws:**

`IllegalArgumentException` - if invalid internal authentication type.

---

## toBackendAuthenticationType

```
public static com.imis.imisarc.client.AuthenticationType  
toBackendAuthenticationType(AuthType type)
```

Converts the specified IMiS/Storage Connector authentication type to the IMiS/ARC Client authentication type.

**Parameters:**

type - the authentication type.

**Returns:**

The IMiS/ARC Client authentication type.

**Throws:**

`IllegalArgumentException` - if invalid authentication type.

---

## fromBackendAuthenticationOptions

```
public static void fromBackendAuthenticationOptions(Map source,  
                                                    Map target)
```

Converts the specified IMiS/ARC Client authentication options to the IMiS/Storage Connector authentication options.

**Parameters:**

source - the IMiS/ARC Client authentication options.

target - the authentication options.

---

## toBackendAuthenticationOptions

```
public static void toBackendAuthenticationOptions(Map source,  
                                                    Map target)
```

Converts the specified IMiS/Storage Connector authentication options to the IMiS/ARC Client authentication options.

**Parameters:**

(continued from last page)

source - the authentication options.

target - the IMiS/ARC Client authentication options.

---

## addObjectRef

```
public void addObjectRef(Object obj)
    throws StorageConnectorException
```

Adds an object to the list of objects referenced by IMiS/Storage Connector.

**Parameters:**

obj - an Object instance.

**Throws:**

[StorageConnectorException](#) - if demo license expired.

---

## getContentTypeResolver

```
public IContentTypeResolver getContentTypeResolver()
```

Gets the content type resolver.

**Returns:**

An IContentTypeResolver instance.

---

## getCustomContentTypeResolver

```
public IContentTypeResolver getCustomContentTypeResolver()
```

Gets the custom content type resolver.

**Returns:**

An IContentTypeResolver instance.

---

## setCustomContentTypeResolver

```
public void setCustomContentTypeResolver(IContentTypeResolver value)
```

Sets the custom content type resolver.

**Parameters:**

value - an IContentTypeResolver instance.

---

## logAddHandler

```
public void logAddHandler(Handler handler)
```

Adds a log handler to the storage object.

**Parameters:**

handler - a Handler instance.

**Throws:**

SecurityException - if a security manager exists and if the caller does not have LoggingPermission("control").

---

## logRemoveHandler

```
public void logRemoveHandler(Handler handler)
```



(continued from last page)

Removes a log handler associated with the storage object.

**Parameters:**

handler - a Handler instance.

**Throws:**

SecurityException - if a security manager exists and if the caller does not have LoggingPermission("control").

---

## logIsInternal

```
public boolean logIsInternal()
```

Gets a value indicating whether the internal logging of the IMiS/Storage Connector is enabled or not.

**Returns:**

true to enable internal logging; otherwise, false.

**See Also:**

[logInternal\(boolean\)](#)

---

## logInternal

```
public void logInternal(boolean enabled)
    throws IOException
```

Enables or disables the IMiS/Storage Connector internal logging.

When internal logging is enabled, IMiS/Storage Connector writes to the "IMiS.StorageConnector.Java.%g.log" rotating log files located in

\* the system temporary folder on Windows systems (i.e. System.getProperty("java.io.tmpdir")) or

\* the user home folder on other systems (i.e. System.getProperty("user.home"))

where %g escape sequence is a generation number used to distinguish rotated log files.

**Parameters:**

enabled - true to enable internal logging; otherwise, false.

**Throws:**

IOException - if there are IO problems opening the files.

SecurityException - if a security manager exists and if the caller does not have LoggingPermission("control").

**See Also:**

[logIsInternal\(\)](#)

---

## logGetLevel

```
public Level logGetLevel()
```

Gets the logging level of the logger.

**Returns:**

The logging level.

---

## logSetLevel

```
public void logSetLevel(Level level)
```

Sets the logging level of the logger and associated log handlers.

**Parameters:**

(continued from last page)

level - the logging level.

**Throws:**

SecurityException - if a security manager exists and if the caller does not have LoggingPermission("control").

---

## getVersionInfo

```
public String getVersionInfo()
```

Gets the version information as string value.

**Returns:**

A string containing version information.

---

## openIMiSARCStorage

```
public Storage openIMiSARCStorage(String host)  
    throws StorageConnectorException
```

Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address and default port 16807.

**Parameters:**

host - a host name or IP address of the IMiS/ARC Storage Server.

**Returns:**

A Storage instance.

**Throws:**

NullPointerException - if host is a null reference.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

---

## openIMiSARCStorage

```
public Storage openIMiSARCStorage(String host,  
    int port)  
    throws StorageConnectorException
```

Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address and port.

**Parameters:**

host - a host name or IP address of the IMiS/ARC Storage Server.

port - a TCP port number of the IMiS/ARC Storage Server.

**Returns:**

A Storage instance.

**Throws:**

NullPointerException - if host is a null reference.

IllegalArgumentException - if port is outside the range of valid port values.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

---

## openIMiSARCStorage

```
public Storage openIMiSARCStorage(String host,  
    String profile)  
    throws StorageConnectorException
```

Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address, default port 16807 and profile.

(continued from last page)

**Parameters:**

host - a host name or IP address of the IMiS/ARC Storage Server.  
profile - a name of the profile on the IMiS/ARC Storage Server.

**Returns:**

A Storage instance.

**Throws:**

NullPointerException - if host is a null reference.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

---

## openIMiSARCStorage

```
public Storage openIMiSARCStorage(String host,  
    int port,  
    String profile)  
throws StorageConnectorException
```

Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address, port and profile.

**Parameters:**

host - a host name or IP address of the IMiS/ARC Storage Server.  
port - a TCP port number of the IMiS/ARC Storage Server.  
profile - a name of the profile on the IMiS/ARC Storage Server.

**Returns:**

A Storage instance.

**Throws:**

NullPointerException - if host is a null reference.  
IllegalArgumentException - if port is outside the range of valid port values.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

---

## openIMiSARCStorage

```
public Storage openIMiSARCStorage(String host,  
    Map options)  
throws StorageConnectorException
```

Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address, default port 16807 and storage options.

**Parameters:**

host - a host name or IP address of the IMiS/ARC Storage Server.  
options - an IMIS\_ARC storage options.

**Returns:**

A Storage instance.

**Throws:**

NullPointerException - if host is a null reference.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

---

## openIMiSARCStorage

```
public Storage openIMiSARCStorage(String host,  
    int port,  
    Map options)  
throws StorageConnectorException
```

(continued from last page)

Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address, port and storage options.

**Parameters:**

host - a host name or IP address of the IMiS/ARC Storage Server.  
port - a TCP port number of the IMiS/ARC Storage Server.  
options - an IMIS\_ARC storage options.

**Returns:**

A Storage instance.

**Throws:**

NullPointerException - if host is a null reference.  
IllegalArgumentException - if port is outside the range of valid port values.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

---

## openIMiSARCStorage

```
public Storage openIMiSARCStorage(String host,  
    String profile,  
    Map options)  
throws StorageConnectorException
```

Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address, default port 16807, profile and storage options.

**Parameters:**

host - a host name or IP address of the IMiS/ARC Storage Server.  
profile - a name of the profile on the IMiS/ARC Storage Server.  
options - an IMIS\_ARC storage options.

**Returns:**

A Storage instance.

**Throws:**

NullPointerException - if host is a null reference.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

---

## openIMiSARCStorage

```
public Storage openIMiSARCStorage(String host,  
    int port,  
    String profile,  
    Map options)  
throws StorageConnectorException
```

Opens a storage on IMiS/ARC Storage Server specified by the host name or IP address, port, profile and storage options.

**Parameters:**

host - a host name or IP address of the IMiS/ARC Storage Server.  
port - a TCP port number of the IMiS/ARC Storage Server.  
profile - a name of the profile on the IMiS/ARC Storage Server.  
options - an IMIS\_ARC storage options.

**Returns:**

A Storage instance.

**Throws:**

NullPointerException - if host is a null reference.  
IllegalArgumentException - if port is outside the range of valid port values.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

## openFileSystemStorage

```
public Storage openFileSystemStorage(String rootDirectory)
    throws StorageConnectorException,
           java.net.UnknownHostException,
           IOException
```

Opens a storage on Local/Remote File System specified by the root directory.

**Note:** FILE\_SYSTEM storage is currently not supported by IMiS/Storage Connector.

**Parameters:**

rootDirectory - a root directory path.

**Returns:**

A Storage instance.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

UnknownHostException - if an unknown host error occurs.

IOException - if an I/O error occurs.

---

## openContentManagerStorage

```
public Storage openContentManagerStorage(String name,
    String username,
    String password)
    throws StorageConnectorException,
           java.net.UnknownHostException,
           IOException
```

Opens a storage on IBM DB2 Content Manager specified by the name using a specified username and a password.

**Note:** CONTENT\_MANAGER storage is currently not supported by IMiS/Storage Connector.

**Parameters:**

name - a name of the datastore.

username - a user name.

password - a password.

**Returns:**

A Storage instance.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

UnknownHostException - if an unknown host error occurs.

IOException - if an I/O error occurs.

---

## openDocumentManagerStorage

```
public Storage openDocumentManagerStorage( )
    throws StorageConnectorException,
           java.net.UnknownHostException,
           IOException
```

Opens a storage on IBM Lotus Domino Document Manager or Domino.Doc compliant Storage Server.

**Note:** DOMINO\_DOC storage is currently not supported by IMiS/Storage Connector.

**Returns:**

(continued from last page)

A Storage instance.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

[UnknownHostException](#) - if an unknown host error occurs.

[IOException](#) - if an I/O error occurs.

---

## openArchive

```
public IArchive openArchive(ArchiveType type,  
    String host,  
    Map options)  
throws StorageConnectorException
```

Opens an archive from the host, specified by the type and host name or IP address and default port 16807.

**Parameters:**

type - a type of archive to which Storage Connector should connect to.

host - a host name or IP address of the archiving system.

options - an archive login and session options.

**Returns:**

An [IArchive](#) instance.

**Throws:**

[NullPointerException](#) - if host is a null reference.

[IllegalArgumentException](#) - if type is invalid archive type.

[StorageConnectorException](#) - if demo license expired.

---

## openArchive

```
public IArchive openArchive(ArchiveType type,  
    String host,  
    int port,  
    Map options)  
throws StorageConnectorException
```

Opens an archive from the host, specified by the type, host name or IP address and TCP port.

Additional options are used to further determine the user credentials which should be used when connecting to the server, archive identifier if service is hosted on multi-archive system, crypto context information, etc...

**Parameters:**

type - a type of archive to which Storage Connector should connect to.

host - a host name or IP address of the archiving system.

port - a TCP port number of the archiving system.

options - an archive login and session options.

**Returns:**

An [IArchive](#) instance.

**Throws:**

[NullPointerException](#) - if host is a null reference.

[IllegalArgumentException](#) - if type is invalid archive type or port is outside the range of valid port values.

[StorageConnectorException](#) - if demo license expired.

(continued from last page)

## objectTransfer

```
public void objectTransfer(IArchive archive,  
    EntityIdKind sourceIdKind,  
    List sourceIds,  
    EntityIdKind targetIdKind,  
    String targetId,  
    String targetContentName,  
    List descriptions)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Transfers source legacy objects to the specified content on the target entity.

### Parameters:

archive - the archive.  
sourceIdKind - the source legacy object identifiers kind.  
sourceIds - the source legacy object identifiers.  
targetIdKind - the target entity identifier kind.  
targetId - the target entity identifier.  
targetContentName - the target content name.  
descriptions - the collection of object descriptions to be used when transferring legacy objects to its target or a null reference if none specified. If object descriptions are specified collection must contain same number of elements as sourceIds collection and their values can be null.

### Throws:

[LoginException](#) - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Collection of legacy object identifiers is required and cannot be empty.
- Number of descriptions must equal number of legacy object identifiers.
- Error getting session from the session pool.
- Error occurred while transferring legacy objects on archive.

[IOException](#) - if an I/O error occurs.

## com.imis.storageconnector Class StorageConnectorException

```

java.lang.Object
  |-- java.lang.Throwable
    |-- java.lang.Exception
      |-- com.imis.GlobalizedException
        |-- com.imis.storageconnector.StorageConnectorException

```

**All Implemented Interfaces:**  
Serializable

```

public class StorageConnectorException
extends com.imis.GlobalizedException

```

Represents errors that occur during IMiS/Storage Connector execution.

### Constructor Summary

public	<a href="#">StorageConnectorException</a> (String resourceName) Initializes a new instance of the StorageConnectorException class with a specified name of the string resource that describes the current exception.
public	<a href="#">StorageConnectorException</a> (String resourceName, Object arg0) Initializes a new instance of the StorageConnectorException class with a specified name of the formatted string resource that describes the current exception.
public	<a href="#">StorageConnectorException</a> (String resourceName, Object[] args) Initializes a new instance of the StorageConnectorException class with a specified name of the formatted string resource that describes the current exception.
public	<a href="#">StorageConnectorException</a> (String resourceName, Throwable cause) Initializes a new instance of the StorageConnectorException class with a specified name of the string resource that describes the current exception and a reference to the inner exception that is the cause of this exception.
public	<a href="#">StorageConnectorException</a> (String resourceName, Throwable cause, Object arg0) Initializes a new instance of the StorageConnectorException class with a specified name of the formatted string resource that describes the current exception and a reference to the inner exception that is the cause of this exception.
public	<a href="#">StorageConnectorException</a> (String resourceName, Throwable cause, Object[] args) Initializes a new instance of the StorageConnectorException class with a specified name of the formatted string resource that describes the current exception and a reference to the inner exception that is the cause of this exception.

### Methods inherited from class com.imis.GlobalizedException

getMessage

### Methods inherited from class java.lang.Throwable



```
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace,
initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString
```

#### Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

## Constructors

### StorageConnectorException

```
public StorageConnectorException(String resourceName)
```

Initializes a new instance of the `StorageConnectorException` class with a specified name of the string resource that describes the current exception.

The `Throwable.getMessage()` of the new instance is initialized with the specified string resource or the resource name, if the string resource is not found in the resources.

#### Parameters:

`resourceName` - the name of the string resource that describes the current exception.

### StorageConnectorException

```
public StorageConnectorException(String resourceName,
                                Object arg0)
```

Initializes a new instance of the `StorageConnectorException` class with a specified name of the formatted string resource that describes the current exception.

The `Throwable.getMessage()` of the new instance is initialized with the specified formatted string resource or the resource name, if the string resource is not found in the resources.

#### Parameters:

`resourceName` - the name of the string resource containing one format item that describes the current exception.  
`arg0` - an object to format.

### StorageConnectorException

```
public StorageConnectorException(String resourceName,
                                Object[] args)
```

Initializes a new instance of the `StorageConnectorException` class with a specified name of the formatted string resource that describes the current exception.

The `Throwable.getMessage()` of the new instance is initialized with the specified formatted string resource or the resource name, if the string resource is not found in the resources.

#### Parameters:

`resourceName` - the name of the string resource containing zero or more format items that describes the current exception.  
`args` - an object array containing zero or more objects to format.

### StorageConnectorException

```
public StorageConnectorException(String resourceName,
                                Throwable cause)
```

(continued from last page)

Initializes a new instance of the `StorageConnectorException` class with a specified name of the string resource that describes the current exception and a reference to the inner exception that is the cause of this exception.

The `Throwable.getMessage()` of the new instance is initialized with the specified string resource or the resource name, if the string resource is not found in the resources.

Note that the detail message associated with cause is not automatically incorporated in this exception's detail message.

**Parameters:**

`resourceName` - the name of the string resource that describes the current exception.

`cause` - the cause (which is saved for later retrieval by the `getCause()` method). (A `null` value is permitted, and indicates that the cause is nonexistent or unknown.)

---

## StorageConnectorException

```
public StorageConnectorException(String resourceName,  
                                Throwable cause,  
                                Object arg0)
```

Initializes a new instance of the `StorageConnectorException` class with a specified name of the formatted string resource that describes the current exception and a reference to the inner exception that is the cause of this exception.

The `Throwable.getMessage()` of the new instance is initialized with the specified formatted string resource or the resource name, if the string resource is not found in the resources.

**Parameters:**

`resourceName` - the name of the string resource containing one format item that describes the current exception.

`cause` - the cause (which is saved for later retrieval by the `getCause()` method). (A `null` value is permitted, and indicates that the cause is nonexistent or unknown.)

`arg0` - an object to format.

---

## StorageConnectorException

```
public StorageConnectorException(String resourceName,  
                                Throwable cause,  
                                Object[] args)
```

Initializes a new instance of the `StorageConnectorException` class with a specified name of the formatted string resource that describes the current exception and a reference to the inner exception that is the cause of this exception.

The `Throwable.getMessage()` of the new instance is initialized with the specified formatted string resource or the resource name, if the string resource is not found in the resources.

**Parameters:**

`resourceName` - the name of the string resource containing zero or more objects to format that describes the current exception.

`cause` - the cause (which is saved for later retrieval by the `getCause()` method). (A `null` value is permitted, and indicates that the cause is nonexistent or unknown.)

`args` - an object array containing zero or more objects to format.

---

## com.imis.storageconnector Class StorageConnectorLogger

```

java.lang.Object
  |
+- java.util.logging.Logger
    |
+- com.imis.util.logging.Logger
    |
+- com.imis.storageconnector.StorageConnectorLogger

```

public class **StorageConnectorLogger**  
extends com.imis.util.logging.Logger

Defines the IMiS/Storage Connector logger.

StorageConnectorLogger is a singleton class derived from the com.imis.util.logging.Logger class that allows a single instance of this class to be created.

### Fields inherited from class java.util.logging.Logger

global

### Method Summary

void	<a href="#">addHandler</a> (Handler handler) Adds the specified log handler to the IMiS/Storage Connector logger.
void	<a href="#">finalize</a> () Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.
static <a href="#">StorageConnectorLogger</a>	<a href="#">getInstance</a> () Gets a singleton instance of the StorageConnectorLogger class.
boolean	<a href="#">getInternalLogging</a> () Gets the value indicating whether the IMiS/Storage Connector logger is logging to an internal file log handler.
void	<a href="#">removeHandler</a> (Handler handler) Removes the specified log handler to the IMiS/Storage Connector logger.
void	<a href="#">setInternalLogging</a> (boolean value) Sets a value indicating whether the IMiS/Storage Connector logger enables or disables logging to an internal file log handler.
void	<a href="#">setLevel</a> (Level level) Gets the effective logging level.

### Methods inherited from class com.imis.util.logging.Logger

addHandler, debug, debug, entering, entering, entering, entering, entering, entering, error, error, error, exiting, exiting, exiting, exiting, info, info, log, logEntry, logEntry, logEntry, logError, removeHandler, warn, warn, warn

### Methods inherited from class java.util.logging.Logger

```
addHandler, config, entering, entering, entering, exiting, exiting, fine, finer,
finest, getAnonymousLogger, getAnonymousLogger, getFilter, getHandlers, getLevel,
getLogger, getLogger, getName, getParent, getResourceBundle, getResourceBundleName,
getUseParentHandlers, info, isLoggable, log, log, log, log, log, logp, logp, logp,
logp, logrb, logrb, logrb, logrb, logrb, removeHandler, setFilter, setLevel, setParent,
setUseParentHandlers, severe, throwing, warning
```

#### Methods inherited from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait
```

## Methods

### finalize

```
protected void finalize()
    throws Throwable
```

Called by the garbage collector on an object when garbage collection determines that there are no more references to the object.

This method disables internal logging and clears logger session list.

Any exception thrown by the finalize method causes the finalization of this object to be halted, but is otherwise ignored.

#### Throws:

`Throwable` - if an error or exception occurs.

### getInstance

```
public static StorageConnectorLogger getInstance()
```

Gets a singleton instance of the `StorageConnectorLogger` class.

#### Returns:

A `StorageConnectorLogger` instance.

### getInternalLogging

```
public boolean getInternalLogging()
```

Gets the value indicating whether the IMiS/Storage Connector logger is logging to an internal file log handler.

#### Returns:

`true` to enable logging to the internal file log handler; otherwise, `false`.

### setInternalLogging

```
public void setInternalLogging(boolean value)
    throws IOException
```

(continued from last page)

Sets a value indicating whether the IMiS/Storage Connector logger enables or disables logging to an internal file log handler.

When this method is called with parameter `true` IMiS/Storage Connector logger starts logging to the rotating log files located in

\* the system temporary folder on Windows systems (i.e. `System.getProperty("java.io.tmpdir")`) or;

\* the user home folder on other systems (i.e. `System.getProperty("user.home")`).

Rotating log file name is set to `"IMiS.StorageConnector.Java.%g.log"`, where `%g` escape sequence is a generation number used to distinguish rotated log files.

**Parameters:**

value - `true` to enable logging to the internal file log handler; otherwise, `false`.

**Throws:**

`IOException` - if there are IO problems opening the files.

`SecurityException` - if a security manager exists and if the caller does not have

`LoggingPermission("control")`.

---

## addHandler

```
public void addHandler(Handler handler)
```

Adds the specified log handler to the IMiS/Storage Connector logger.

**Parameters:**

handler - the `java.util.logging.Handler` object to add.

---

## removeHandler

```
public void removeHandler(Handler handler)
```

Removes the specified log handler to the IMiS/Storage Connector logger.

**Parameters:**

handler - the `java.util.logging.Handler` object to remove.

---

## setLevel

```
public void setLevel(Level level)
```

Gets the effective logging level.

**Parameters:**

level - the logging level for the logger. Default value is `Level.INFO`

**Throws:**

`SecurityException` - if a security manager exists and if the caller does not have

`LoggingPermission("control")`.

---

## com.imis.storageconnector Class StorageType

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.StorageType
  
```

**All Implemented Interfaces:**  
Serializable, Comparable

```

public final class StorageType
extends Enum
  
```

Defines storage types supported by IMiS/Storage Connector.

**Note:** Only IMIS\_ARC storage is currently implemented by IMiS/Storage Connector.

### Field Summary

public static final	<a href="#">CONTENT_MANAGER</a> IBM DB2 Content Manager Server.
public static final	<a href="#">DOMINO_DOC</a> IBM Lotus Domino Document Manager or Domino.Doc compliant Storage Server.
public static final	<a href="#">FILE_SYSTEM</a> Local/Remote File System.
public static final	<a href="#">IMIS_ARC</a> IMiS/ARChive Storage Server.
public static final	<a href="#">LNDI_MSS</a> Lotus Notes Document Imaging Mass Storage Server (LN:DI MSS).
public static final	<a href="#">ODMA</a> ODMA compliant Storage Server.

### Method Summary

static <a href="#">StorageType</a>	<a href="#">valueOf</a> (String name)
static <a href="#">StorageType[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods inherited from interface** `java.lang.Comparable``compareTo`

## Fields

### FILE\_SYSTEM

```
public static final com.imis.storageconnector.StorageType FILE_SYSTEM
```

Local/Remote File System.

**Note:** `FILE_SYSTEM` storage is currently not supported by IMiS/Storage Connector.

### IMIS\_ARC

```
public static final com.imis.storageconnector.StorageType IMIS_ARC
```

IMiS/ARChive Storage Server.

### LNDI\_MSS

```
public static final com.imis.storageconnector.StorageType LNDI_MSS
```

Lotus Notes Document Imaging Mass Storage Server (LN:DI MSS).

**Note:** `LNDI_MSS` storage is currently not supported by IMiS/Storage Connector.

### ODMA

```
public static final com.imis.storageconnector.StorageType ODMA
```

ODMA compliant Storage Server.

**Note:** `ODMA` storage is currently not supported by IMiS/Storage Connector.

### DOMINO\_DOC

```
public static final com.imis.storageconnector.StorageType DOMINO_DOC
```

IBM Lotus Domino Document Manager or Domino.Doc compliant Storage Server.

**Note:** `DOMINO_DOC` storage is currently not supported by IMiS/Storage Connector.

### CONTENT\_MANAGER

```
public static final com.imis.storageconnector.StorageType CONTENT_MANAGER
```

IBM DB2 Content Manager Server.

**Note:** `CONTENT_MANAGER` storage is currently not supported by IMiS/Storage Connector.

## Methods

(continued from last page)

## values

```
public final static StorageType\[\] values()
```

---

## valueOf

```
public static StorageType valueOf(String name)
```



---

**Package**

**com.imis.storageconnector.entitymodel**

## com.imis.storageconnector.entitymodel

### Class AccessMode

java.lang.Object

└─ java.lang.Enum

└─ com.imis.storageconnector.entitymodel.AccessMode

#### All Implemented Interfaces:

Serializable, Comparable

public final class **AccessMode**  
extends Enum

Represents the type of access.

### Field Summary

public static final	<a href="#">READ</a> Read access.
public static final	<a href="#">READ_WRITE</a> Read and write access.
public static final	<a href="#">WRITE</a> Write access.

### Method Summary

static <a href="#">AccessMode</a>	<a href="#">valueOf</a> (String name)
static <a href="#">AccessMode[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields

(continued from last page)

## READ

```
public static final com.imis.storageconnector.entitymodel.AccessMode READ
```

Read access. Data can be read.

---

## WRITE

```
public static final com.imis.storageconnector.entitymodel.AccessMode WRITE
```

Write access. Data can be written.

---

## READ\_WRITE

```
public static final com.imis.storageconnector.entitymodel.AccessMode READ_WRITE
```

Read and write access. Data can be written and read.

## Methods

### values

```
public final static AccessMode[] values()
```

---

### valueOf

```
public static AccessMode valueOf(String name)
```

## com.imis.storageconnector.entitymodel

### Class ACLEntrySubjectType

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.entitymodel.ACLEntrySubjectType
  
```

#### All Implemented Interfaces:

Serializable, Comparable

```

public final class ACLEntrySubjectType
extends Enum
  
```

Defines constants for ACL entry subject type.

### Field Summary

public static final	<a href="#">ATTRIBUTE</a> Subject is attribute.
public static final	<a href="#">DIRECTORY_ENTITY</a> Subject is directory entity.
public static final	<a href="#">UNKNOWN</a> Not known.

### Method Summary

static <a href="#">ACLEntrySubjectType</a>	<a href="#">valueOf</a> (String name)
static <a href="#">ACLEntrySubjectType[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields

(continued from last page)

## UNKNOWN

```
public static final com.imis.storageconnector.entitymodel.ACLEntrySubjectType UNKNOWN
```

Not known.

## DIRECTORY\_ENTITY

```
public static final com.imis.storageconnector.entitymodel.ACLEntrySubjectType  
DIRECTORY_ENTITY
```

Subject is directory entity.

## ATTRIBUTE

```
public static final com.imis.storageconnector.entitymodel.ACLEntrySubjectType  
ATTRIBUTE
```

Subject is attribute.

## Methods

### values

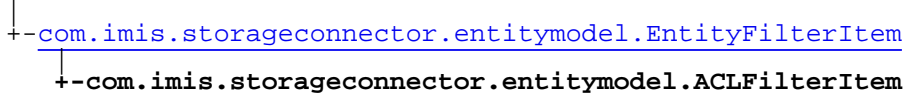
```
public final static ACLEntrySubjectType\[\] values()
```

### valueOf

```
public static ACLEntrySubjectType valueOf(String name)
```

## com.imis.storageconnector.entitymodel Class ACLFilterItem

java.lang.Object



public class **ACLFilterItem**  
extends [EntityFilterItem](#)

Represents the ACL filter item.

### Constructor Summary

public	<a href="#">ACLFilterItem</a> ( <a href="#">IDirectoryEntity</a> subject, Set filter) Initializes a new instance of the <a href="#">ACLFilterItem</a> class.
--------	---

### Method Summary

String	<a href="#">toString()</a> Returns a string that represents the entity filter item.
--------	--

Methods inherited from class [com.imis.storageconnector.entitymodel.EntityFilterItem](#)

[toString](#)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### ACLFilterItem

public **ACLFilterItem**([IDirectoryEntity](#) subject,  
Set filter)

Initializes a new instance of the [ACLFilterItem](#) class.

#### Parameters:

subject - the directory entity subject.  
filter - the entity rights filter.

#### Throws:

NullPointerException - if subject is a null reference.

## Methods

(continued from last page)

## **toString**

```
public String toString()
```

Returns a string that represents the entity filter item.

### **Returns:**

A string that represents the entity filter item.

## com.imis.storageconnector.entitymodel

### Class ContentPartIdEncoding

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.entitymodel.ContentPartIdEncoding
  
```

#### All Implemented Interfaces:

Serializable, Comparable

public final class **ContentPartIdEncoding**  
extends Enum

Defines constants for the encoding of content part identifier used when creating new content parts.

Used as values for the IMISARCHIVE archive content part identifier encoding disposition option.

#### See Also:

[StorageConnector.OPTION\\_CONTENT\\_PART\\_IDENCODING](#), [StorageConnector.openArchive\(ArchiveType, String, Map\)](#), [StorageConnector.openArchive\(ArchiveType, String, int, Map\)](#)

### Field Summary

public static final	<a href="#">BASE16</a> Base 16 encoding.
public static final	<a href="#">BASE64</a> Base 64 encoding.
public static final	<a href="#">BASE85</a> Base 85 encoding.

### Method Summary

static <a href="#">ContentPartIdEncoding</a>	<a href="#">valueOf</a> (String name)
static <a href="#">ContentPartIdEncoding</a> <a href="#">[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo



## Fields

### BASE16

```
public static final com.imis.storageconnector.entitymodel.ContentPartIdEncoding BASE16
```

Base 16 encoding.

---

### BASE64

```
public static final com.imis.storageconnector.entitymodel.ContentPartIdEncoding BASE64
```

Base 64 encoding.

---

### BASE85

```
public static final com.imis.storageconnector.entitymodel.ContentPartIdEncoding BASE85
```

Base 85 encoding.

## Methods

### values

```
public final static ContentPartIdEncoding\[\] values()
```

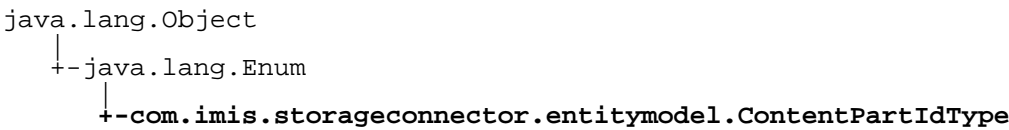
---

### valueOf

```
public static ContentPartIdEncoding valueOf(String name)
```

# com.imis.storageconnector.entitymodel

## Class ContentPartIdType



All Implemented Interfaces:  
Serializable, Comparable

public final class **ContentPartIdType**  
extends Enum

Defines constants for the type of content part identifiers used when creating new content parts.

Used as values for the IMISARCHIVE archive content part identifier type disposition option.

See Also:  
[StorageConnector.OPTION\\_CONTENT\\_PART\\_IDTYPE](#), [StorageConnector.openArchive\(ArchiveType, String, Map\)](#), [StorageConnector.openArchive\(ArchiveType, String, int, Map\)](#)

Field Summary	
public static final	<a href="#">ID192</a> 192 bit long file value identifier.
public static final	<a href="#">ID256</a> 256 bit long file value identifier.

Method Summary	
static <a href="#">ContentPartIdType</a>	<a href="#">valueOf</a> (String name)
static <a href="#">ContentPartIdType[]</a>	<a href="#">values</a> ()

Methods inherited from class java.lang.Enum	
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf	

Methods inherited from class java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

Methods inherited from interface java.lang.Comparable	
compareTo	

## Fields

(continued from last page)

## ID192

```
public static final com.imis.storageconnector.entitymodel.ContentPartIdType ID192
```

192 bit long file value identifier.

---

## ID256

```
public static final com.imis.storageconnector.entitymodel.ContentPartIdType ID256
```

256 bit long file value identifier.

## Methods

### values

```
public final static ContentPartIdType\[\] values()
```

---

### valueOf

```
public static ContentPartIdType valueOf(String name)
```

## com.imis.storageconnector.entitymodel Class DeletionKind

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.entitymodel.DeletionKind
  
```

**All Implemented Interfaces:**  
Serializable, Comparable

public final class **DeletionKind**  
extends Enum

Describes the kind of entity deletion.

### Field Summary

public static final	<a href="#">DELETED</a> Entity was deleted.
public static final	<a href="#">DISPOSED</a> Entity was disposed through review.

### Method Summary

static <a href="#">DeletionKind</a>	<a href="#">valueOf</a> (String name)
static <a href="#">DeletionKind[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields

#### DELETED

```

public static final com.imis.storageconnector.entitymodel.DeletionKind DELETED

    Entity was deleted.
  
```

---

## DISPOSED

```
public static final com.imis.storageconnector.entitymodel.DeletionKind DISPOSED
```

Entity was disposed through review.

## Methods

### values

```
public final static DeletionKind\[\] values()
```

---

### valueOf

```
public static DeletionKind valueOf(String name)
```

# com.imis.storageconnector.entitymodel

## Class EntityFilter

java.lang.Object

└-com.imis.storageconnector.entitymodel.EntityFilter

public class **EntityFilter**  
extends Object

Represents the entity filter.

### Field Summary

public static final	<a href="#">ALL</a> The entity filter that includes all entities.
public static final	<a href="#">CLASSES</a> The entity filter that includes only <a href="#">IClass</a> entities.
public static final	<a href="#">DOCUMENTS</a> The entity filter that includes only <a href="#">IDocument</a> entities.
public static final	<a href="#">FOLDERS</a> The entity filter that includes only <a href="#">IFolder</a> entities.

### Constructor Summary

public	<a href="#">EntityFilter</a> (Set filter, <a href="#">EntityFilterItem</a> item) Initializes a new instance of the <a href="#">EntityFilter</a> class.
--------	---

### Method Summary

<a href="#">EntityFilterItem</a>	<a href="#">getItem()</a> Gets the entity filter item.
Set	<a href="#">getType()</a> Gets the entity type filter.
String	<a href="#">toString()</a> Returns a string that represents the current EntityFilter.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Fields

#### ALL

public static final com.imis.storageconnector.entitymodel.EntityFilter **ALL**

(continued from last page)

The entity filter that includes all entities.

---

## CLASSES

```
public static final com.imis.storageconnector.entitymodel.EntityFilter CLASSES
```

The entity filter that includes only [IClass](#) entities.

---

## FOLDERS

```
public static final com.imis.storageconnector.entitymodel.EntityFilter FOLDERS
```

The entity filter that includes only [IFolder](#) entities.

---

## DOCUMENTS

```
public static final com.imis.storageconnector.entitymodel.EntityFilter DOCUMENTS
```

The entity filter that includes only [IDocument](#) entities.

## Constructors

### EntityFilter

```
public EntityFilter(Set filter,  
                    EntityFilterItem item)
```

Initializes a new instance of the [EntityFilter](#) class.

#### Parameters:

`filter` - the entity type filter.  
`item` - the entity filter item.

## Methods

### getType

```
public Set getType()
```

Gets the entity type filter.

#### Returns:

The entity type filter.

---

### getItem

```
public EntityFilterItem getItem()
```

Gets the entity filter item.

#### Returns:

The entity filter item.

---

### toString

```
public String toString()
```

Returns a string that represents the current `EntityFilter`.

(continued from last page)

**Returns:**

A string that represents the current `EntityFilter`.



# com.imis.storageconnector.entitymodel

## Class EntityFilterItem

java.lang.Object

└─com.imis.storageconnector.entitymodel.EntityFilterItem

Direct Known Subclasses:

[ACLFILTERItem](#)

public abstract class **EntityFilterItem**  
extends Object

Represents the base class for entity filter items.

See Also:

[EntityFilter](#)

### Constructor Summary

public	<a href="#">EntityFilterItem()</a>
--------	------------------------------------

### Method Summary

abstract String	<a href="#">toString()</a> Returns a string that represents the entity filter item.
-----------------	--

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### EntityFilterItem

public **EntityFilterItem()**

## Methods

### toString

public abstract String **toString()**

Returns a string that represents the entity filter item.

**Returns:**

A string that represents the entity filter item.

# com.imis.storageconnector.entitymodel

## Class EntityIdEncoding

```

java.lang.Object
  |
  +- java.lang.Enum
        +- com.imis.storageconnector.entitymodel.EntityIdEncoding
  
```

### All Implemented Interfaces:

Serializable, Comparable

public final class **EntityIdEncoding**  
extends Enum

Defines constants for the encoding of entity identifier used when creating new entities.

Used as values for the IMISARCHIVE archive entity identifier encoding disposition option.

### See Also:

[StorageConnector.OPTION\\_ENTITY\\_IDENCODING](#), [StorageConnector.openArchive\(ArchiveType, String, Map\)](#), [StorageConnector.openArchive\(ArchiveType, String, int, Map\)](#)

## Field Summary

public static final	<a href="#">BASE16</a> Base 16 encoding.
public static final	<a href="#">BASE64</a> Base 64 encoding.
public static final	<a href="#">BASE85</a> Base 85 encoding.

## Method Summary

static <a href="#">EntityIdEncoding</a>	<a href="#">valueOf</a> (String name)
static <a href="#">EntityIdEncoding[]</a>	<a href="#">values</a> ()

### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods inherited from interface java.lang.Comparable

compareTo

## Fields

### BASE16

```
public static final com.imis.storageconnector.entitymodel.EntityIdEncoding BASE16
```

Base 16 encoding.

---

### BASE64

```
public static final com.imis.storageconnector.entitymodel.EntityIdEncoding BASE64
```

Base 64 encoding.

---

### BASE85

```
public static final com.imis.storageconnector.entitymodel.EntityIdEncoding BASE85
```

Base 85 encoding.

## Methods

### values

```
public final static EntityIdEncoding\[\] values()
```

---

### valueOf

```
public static EntityIdEncoding valueOf(String name)
```

## com.imis.storageconnector.entitymodel Class EntityIdKind

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.entitymodel.EntityIdKind
  
```

### All Implemented Interfaces:

Serializable, Comparable

```

public final class EntityIdKind
extends Enum
  
```

Describes the kind of identifier used to access an IEntity.

### See Also:

[IEntity](#)

## Field Summary

public static final	<a href="#">CLASSIFICATION_CODE</a> Defines a classification code in canonical format which describes the entity position in the Classification tree.
public static final	<a href="#">EXTERNAL</a> Defines an external entity identifier which is generated by external system and associated with an IEntity are its creation or modification.
public static final	<a href="#">INTERNAL</a> Defines an internal entity identifier which is returned by the archive.
public static final	<a href="#">UNKNOWN</a> Defines an invalid entity identifier which should never be used for accessing an IEntity and its purpose is as an invalid value in parameter initialization.

## Method Summary

static <a href="#">EntityIdKind</a>	<a href="#">valueOf</a> (String name)
static <a href="#">EntityIdKind[]</a>	<a href="#">values</a> ()

### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods inherited from interface java.lang.Comparable

compareTo

---

## Fields

### UNKNOWN

```
public static final com.imis.storageconnector.entitymodel.EntityIdKind UNKNOWN
```

Defines an invalid entity identifier which should never be used for accessing an `IEntity` and its purpose is as an invalid value in parameter initialization.

---

### INTERNAL

```
public static final com.imis.storageconnector.entitymodel.EntityIdKind INTERNAL
```

Defines an internal entity identifier which is returned by the archive. It can be encoded in various combinations of [EntityIdType](#) and [EntityIdEncoding](#).

---

### EXTERNAL

```
public static final com.imis.storageconnector.entitymodel.EntityIdKind EXTERNAL
```

Defines an external entity identifier which is generated by external system and associated with an `IEntity` are its creation or modification. It must uniquely identify an entity in order to be used for entity identification.

---

### CLASSIFICATION\_CODE

```
public static final com.imis.storageconnector.entitymodel.EntityIdKind  
CLASSIFICATION_CODE
```

Defines a classification code in canonical format which describes the entity position in the Classification tree. Classification code canonical parts are separated by the entity hierarchy delimiter "^". Canonical parts are composed of an entity type prefix, "C=", "F=" or "D=" for class, folder or document entity respectively, and relative classification code. (e.g.: "C=1^C=1^C=1" addresses a class which is identified by classification code 1 at level 1, classification code 1 on level 2 and classification code 1 on level 3).

---

## Methods

### values

```
public final static EntityIdKind\[\] values()
```

---

### valueOf

```
public static EntityIdKind valueOf(String name)
```

# com.imis.storageconnector.entitymodel

## Class EntityIdType



All Implemented Interfaces:  
Serializable, Comparable

public final class **EntityIdType**  
extends Enum

Defines constants for the type of entity identifiers used when creating new entities.

Used as values for the IMISARCHIVE archive entity identifier type disposition option.

See Also:  
[StorageConnector.OPTION\\_ENTITY\\_IDTYPE](#), [StorageConnector.openArchive\(ArchiveType, String, Map\)](#), [StorageConnector.openArchive\(ArchiveType, String, int, Map\)](#)

Field Summary	
public static final	<a href="#">ID192</a> 192 bit long entity identifier.
public static final	<a href="#">ID256</a> 256 bit long entity identifier.

Method Summary	
static <a href="#">EntityIdType</a>	<a href="#">valueOf</a> (String name)
static <a href="#">EntityIdType[]</a>	<a href="#">values</a> ()

Methods inherited from class java.lang.Enum
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface java.lang.Comparable
compareTo

## Fields

(continued from last page)

## ID192

```
public static final com.imis.storageconnector.entitymodel.EntityIdType ID192
```

192 bit long entity identifier.

---

## ID256

```
public static final com.imis.storageconnector.entitymodel.EntityIdType ID256
```

256 bit long entity identifier.

## Methods

### values

```
public final static EntityIdType\[\] values()
```

---

### valueOf

```
public static EntityIdType valueOf(String name)
```

# com.imis.storageconnector.entitymodel

## Class EntityQueryScope

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.entitymodel.EntityQueryScope
  
```

### All Implemented Interfaces:

Serializable, Comparable

```

public final class EntityQueryScope
extends Enum
  
```

Defines the entity info query scope that server must honor when returning its result. Note that general information (i.e. internal identifier, entity type, template name, created, updated and accessed timestamps, effective rights and child entity count) is always returned.

### Field Summary

public static final	<a href="#">DRAFT</a> Information about entity draft should be returned.
public static final	<a href="#">PARENT_INFO</a> Parent information should be returned (returns the same information as for queried entity).
public static final	<a href="#">PROPERTIES</a> Properties information should be returned.
public static final	<a href="#">REFERENCES</a> Entity references should be returned.
public static final	<a href="#">TEMPLATES</a> Sub-entity templates should be returned.
public static final	<a href="#">VERSIONS</a> All entity versions should be returned.

### Method Summary

static <a href="#">EntityQueryScope</a>	<a href="#">valueOf</a> (String name)
static <a href="#">EntityQueryScope[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait



**Methods inherited from interface** `java.lang.Comparable``compareTo`

## Fields

### PROPERTIES

```
public static final com.imis.storageconnector.entitymodel.EntityQueryScope PROPERTIES
```

Properties information should be returned.

### TEMPLATES

```
public static final com.imis.storageconnector.entitymodel.EntityQueryScope TEMPLATES
```

Sub-entity templates should be returned.

### PARENT\_INFO

```
public static final com.imis.storageconnector.entitymodel.EntityQueryScope PARENT_INFO
```

Parent information should be returned (returns the same information as for queried entity).

### REFERENCES

```
public static final com.imis.storageconnector.entitymodel.EntityQueryScope REFERENCES
```

Entity references should be returned.

### DRAFT

```
public static final com.imis.storageconnector.entitymodel.EntityQueryScope DRAFT
```

Information about entity draft should be returned.

### VERSIONS

```
public static final com.imis.storageconnector.entitymodel.EntityQueryScope VERSIONS
```

All entity versions should be returned.

## Methods

#### values

```
public final static EntityQueryScope\[\] values()
```

#### valueOf

```
public static EntityQueryScope valueOf(String name)
```

## com.imis.storageconnector.entitymodel

### Class EntityRightsFilter

```

java.lang.Object
|
+- java.lang.Enum
|
+- com.imis.storageconnector.entitymodel.EntityRightsFilter

```

**All Implemented Interfaces:**  
 Serializable, Comparable

```

public final class EntityRightsFilter
extends Enum

```

Defines the filter enumeration which controls what entities in regards to rights should be included in the entity collection.

Field Summary	
public static final	<a href="#">CHANGE_RETENTION</a> Entities with ChangeRetention right should be included in the collection.
public static final	<a href="#">CHANGE_RIGHTS</a> Entities with ChangeRights right should be included in the collection.
public static final	<a href="#">CHANGE_SECURITY_CLASS</a> Entities with ChangeSecurityClass right should be included in the collection.
public static final	<a href="#">CHANGE_STATUS</a> Entities with ChangeStatus right should be included in the collection.
public static final	<a href="#">CREATE_REFERENCES</a> Entities with CreateReferences right should be included in the collection.
public static final	<a href="#">CREATE_SUB_ENTITIES</a> Entities with CreateSubEntities right should be included in the collection.
public static final	<a href="#">DELETE_ACCESS</a> Entities with DeleteAccess right should be included in the collection.
public static final	<a href="#">MOVE_ACCESS</a> Entities with MoveAccess right should be included in the collection.
public static final	<a href="#">READ_ACCESS</a> Entities with ReadAccess right should be included in the collection.
public static final	<a href="#">WRITE_ACCESS</a> Entities with WriteAccess right should be included in the collection.

Method Summary	
static <a href="#">EntityRightsFilter</a>	<a href="#">valueOf</a> (String name)
static <a href="#">EntityRightsFilter[]</a>	<a href="#">values</a> ()

**Methods inherited from class** `java.lang.Enum`

`clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

**Methods inherited from class** `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

**Methods inherited from interface** `java.lang.Comparable`

`compareTo`

## Fields

### READ\_ACCESS

```
public static final com.imis.storageconnector.entitymodel.EntityRightsFilter  
READ_ACCESS
```

Entities with `ReadAccess` right should be included in the collection.

### WRITE\_ACCESS

```
public static final com.imis.storageconnector.entitymodel.EntityRightsFilter  
WRITE_ACCESS
```

Entities with `WriteAccess` right should be included in the collection.

### MOVE\_ACCESS

```
public static final com.imis.storageconnector.entitymodel.EntityRightsFilter  
MOVE_ACCESS
```

Entities with `MoveAccess` right should be included in the collection.

### DELETE\_ACCESS

```
public static final com.imis.storageconnector.entitymodel.EntityRightsFilter  
DELETE_ACCESS
```

Entities with `DeleteAccess` right should be included in the collection.

### CHANGE\_RIGHTS

```
public static final com.imis.storageconnector.entitymodel.EntityRightsFilter  
CHANGE_RIGHTS
```

Entities with `ChangeRights` right should be included in the collection.

### CREATE\_SUB\_ENTITIES

```
public static final com.imis.storageconnector.entitymodel.EntityRightsFilter  
CREATE_SUB_ENTITIES
```

(continued from last page)

Entities with `CreateSubEntities` right should be included in the collection.

---

## CHANGE\_SECURITY\_CLASS

```
public static final com.imis.storageconnector.entitymodel.EntityRightsFilter  
CHANGE_SECURITY_CLASS
```

Entities with `ChangeSecurityClass` right should be included in the collection.

---

## CHANGE\_STATUS

```
public static final com.imis.storageconnector.entitymodel.EntityRightsFilter  
CHANGE_STATUS
```

Entities with `ChangeStatus` right should be included in the collection.

---

## CHANGE\_RETENTION

```
public static final com.imis.storageconnector.entitymodel.EntityRightsFilter  
CHANGE_RETENTION
```

Entities with `ChangeRetention` right should be included in the collection.

---

## CREATE\_REFERENCES

```
public static final com.imis.storageconnector.entitymodel.EntityRightsFilter  
CREATE_REFERENCES
```

Entities with `CreateReferences` right should be included in the collection.

## Methods

### values

```
public final static EntityRightsFilter\[\] values()
```

---

### valueOf

```
public static EntityRightsFilter valueOf(String name)
```

## com.imis.storageconnector.entitymodel

### Class EntitySortKey

java.lang.Object

└─com.imis.storageconnector.entitymodel.EntitySortKey

public class **EntitySortKey**  
extends Object

Defines the object which encapsulates the description how entity collection should be sorted.

**See Also:**

[IEntity.getSubEntities\(EntityFilter, Collection, Collection\)](#),  
[IEntityStub.getSubEntities\(EntityFilter, Collection, Collection\)](#),  
[IArchive.getRootClasses\(Collection, Collection\)](#)

### Constructor Summary

public	<a href="#">EntitySortKey</a> (String propertyName, <a href="#">EntitySortKeyDirection</a> direction) Initializes a new instance of the <a href="#">EntitySortKey</a> class.
--------	---

### Method Summary

<a href="#">EntitySortKeyDirection</a>	<a href="#">getDirection</a> () Gets the entity sorting direction with which the values of property should be sorted.
String	<a href="#">getPropertyName</a> () Gets the property unique name.
void	<a href="#">setDirection</a> ( <a href="#">EntitySortKeyDirection</a> value) Sets the entity sorting direction with which the values of property should be sorted.
void	<a href="#">setPropertyName</a> (String value) Sets the property unique name.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### EntitySortKey

public **EntitySortKey**(String propertyName,  
[EntitySortKeyDirection](#) direction)

Initializes a new instance of the [EntitySortKey](#) class.

#### Parameters:

propertyName - the property unique name which should be sorted.  
direction - the sorting direction.

#### Throws:

(continued from last page)

NullPointerException - if `propertyName` is a null reference.

## Methods

### getPropertyName

```
public String getPropertyName()
```

Gets the property unique name.

The property unique name identifies a `IProperty` or `IReadOnlyProperty` within an `IEntity` instance.

**Returns:**

The property unique name.

### setPropertyName

```
public void setPropertyName(String value)
```

Sets the property unique name.

The property unique name identifies a `IProperty` or `IReadOnlyProperty` within an `IEntity` instance.

**Parameters:**

`value` - the property unique name.

**Throws:**

NullPointerException - if `value` is a null reference.

### getDirection

```
public EntitySortKeyDirection getDirection()
```

Gets the entity sorting direction with which the values of property should be sorted.

**Returns:**

The entity sorting direction.

### setDirection

```
public void setDirection(EntitySortKeyDirection value)
```

Sets the entity sorting direction with which the values of property should be sorted.

**Parameters:**

`value` - the entity sorting direction.

## com.imis.storageconnector.entitymodel Class EntitySortKeyDirection

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.entitymodel.EntitySortKeyDirection
  
```

**All Implemented Interfaces:**  
Serializable, Comparable

public final class **EntitySortKeyDirection**  
extends Enum

Defines enumeration for sorting direction of entity collection elements.

### Field Summary

public static final	<a href="#">ASCENDING</a> Values of property, identified by <a href="#">EntitySortKey.getPropertyName()</a> is to be sorted in ascending order.
public static final	<a href="#">DESCENDING</a> Values of property, identified by <a href="#">EntitySortKey.getPropertyName()</a> is to be sorted in descending order.

### Method Summary

static <a href="#">EntitySortKeyDirection</a> n	<a href="#">valueOf</a> (String name)
static <a href="#">EntitySortKeyDirection</a> n[]	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields

(continued from last page)

## ASCENDING

```
public static final com.imis.storageconnector.entitymodel.EntitySortKeyDirection  
ASCENDING
```

Values of property, identified by [EntitySortKey.getPropertyName\(\)](#) is to be sorted in ascending order.

---

## DESCENDING

```
public static final com.imis.storageconnector.entitymodel.EntitySortKeyDirection  
DESCENDING
```

Values of property, identified by [EntitySortKey.getPropertyName\(\)](#) is to be sorted in descending order.

## Methods

### values

```
public final static EntitySortKeyDirection\[\] values()
```

---

### valueOf

```
public static EntitySortKeyDirection valueOf(String name)
```



# com.imis.storageconnector.entitymodel

## Class EntityType

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.entitymodel.EntityType
  
```

**All Implemented Interfaces:**  
 Serializable, Comparable

public final class **EntityType**  
 extends Enum

Describes the IEntity specialization in the entity tree.

**See Also:**

[IEntity](#)

### Field Summary

public static final	<a href="#">ANY</a> Represents a generic entity type.
public static final	<a href="#">CLASS</a> Class entity type.
public static final	<a href="#">DOCUMENT</a> Document entity type.
public static final	<a href="#">FOLDER</a> Folder entity type.
public static final	<a href="#">UNKNOWN</a> Unknown entity type.

### Method Summary

static <a href="#">EntityType</a>	<a href="#">valueOf</a> (String name)
static <a href="#">EntityType[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

---

## Fields

### UNKNOWN

```
public static final com.imis.storageconnector.entitymodel.EntityType UNKNOWN
```

Unknown entity type. This value defines an invalid `IEntity` type which is never returned on an coherent entity.

---

### CLASS

```
public static final com.imis.storageconnector.entitymodel.EntityType CLASS
```

Class entity type. This value defines a [IClass](#) object, instances which make classification scheme.

---

### FOLDER

```
public static final com.imis.storageconnector.entitymodel.EntityType FOLDER
```

Folder entity type. This value defines a [IFolder](#) object, which can contain additional sub-folders or documents.

---

### DOCUMENT

```
public static final com.imis.storageconnector.entitymodel.EntityType DOCUMENT
```

Document entity type. This value defines a [IDocument](#) object, which can be contained within a class, folder or sub-folder.

---

### ANY

```
public static final com.imis.storageconnector.entitymodel.EntityType ANY
```

Represents a generic entity type. This value defines a [IEntity](#) object.

---

## Methods

### values

```
public final static EntityType\[\] values()
```

---

### valueOf

```
public static EntityType valueOf(String name)
```

## com.imis.storageconnector.entitymodel

### Class EntityTypeFilter

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.entitymodel.EntityTypeFilter
  
```

#### All Implemented Interfaces:

Serializable, Comparable

```

public final class EntityTypeFilter
extends Enum
  
```

Defines the filter enumeration which controls what types of entities should be included in the entity collection.

#### See Also:

[IEntity.getSubEntities\(EntityFilter, Collection, Collection\)](#),  
[IEntityStub.getSubEntities\(EntityFilter, Collection, Collection\)](#),  
[IArchive.getRootClasses\(Collection, Collection\)](#)

### Field Summary

public static final	<a href="#">CLASSES</a> Entities of type <a href="#">EntityType.CLASS</a> should be included in the collection.
public static final	<a href="#">DOCUMENTS</a> Entities of type <a href="#">EntityType.DOCUMENT</a> should be included in the collection.
public static final	<a href="#">FOLDERS</a> Entities of type <a href="#">EntityType.FOLDER</a> should be included in the collection.

### Method Summary

static <a href="#">EntityTypeFilter</a>	<a href="#">valueOf</a> (String name)
static <a href="#">EntityTypeFilter[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

(continued from last page)

## Fields

### CLASSES

```
public static final com.imis.storageconnector.entitymodel.EntityTypeFilter CLASSES
```

Entities of type [EntityType.CLASS](#) should be included in the collection.

### FOLDERS

```
public static final com.imis.storageconnector.entitymodel.EntityTypeFilter FOLDERS
```

Entities of type [EntityType.FOLDER](#) should be included in the collection.

### DOCUMENTS

```
public static final com.imis.storageconnector.entitymodel.EntityTypeFilter DOCUMENTS
```

Entities of type [EntityType.DOCUMENT](#) should be included in the collection.

## Methods

### values

```
public final static EntityTypeFilter\[\] values()
```

### valueOf

```
public static EntityTypeFilter valueOf(String name)
```

## com.imis.storageconnector.entitymodel Interface IACL

public interface **IACL**  
extends

Represents an Access Control List, a collection of rights attached to an IEntity.

The IACL object enables the caller to access the IEntity Security Descriptor. Based on the caller set of rights, it has the option of reading and/or modifying the IEntityACLEntry or IPropertyACLEntry objects, representing security constraints on the entity and its properties. It also provides means of creating new ACL entries which can be committed to the owner entity at the commit stage using the [IEntity.save\(\)](#) method or discarded.

**See Also:**

[IEntity](#), [IEntityACLEntry](#), [IPropertyACLEntry](#)

### Method Summary

<a href="#">IEntityACLEntry</a>	<a href="#">addEntry</a> ( <a href="#">ACLEntrySubjectType</a> subjectType, String subject) Adds a new IEntityACLEntry specified with the ACL entry subject type and unique subject identifier to the owning entity IACL.
<a href="#">IPropertyACLEntry</a>	<a href="#">addPropertyEntry</a> (String propertyName, <a href="#">ACLEntrySubjectType</a> subjectType, String subject) Adds a new IPropertyACLEntry specified by the property name, ACL entry subject type and unique subject identifier to the owning entity IACL.
com.imis.util.IReadOnlyMap	<a href="#">getEntries</a> () Gets the read-only map of entity ACL entries assigned to the owner IEntity.
com.imis.util.IReadOnlyMap	<a href="#">getPropertyEntries</a> () Gets the read-only map of read-only maps of property ACL entries assigned to the owner IProperty.

### Methods

#### getEntries

public com.imis.util.IReadOnlyMap **getEntries**()

Gets the read-only map of entity ACL entries assigned to the owner IEntity.

Use this method to iterate and access different entity ACL entries.

**Returns:**

A read-only map of IEntityACLEntry objects sorted by pairs of ACL entry subject type and unique identifier.

#### getPropertyEntries

public com.imis.util.IReadOnlyMap **getPropertyEntries**()

Gets the read-only map of read-only maps of property ACL entries assigned to the owner IProperty.

Use this method to iterate and access different property ACL entries.

**Returns:**

(continued from last page)

A read-only map of `IPropertyACLEntry` objects sorted externally by property name and internally by pairs of property ACL entry subject type and unique identifier.

---

## addEntry

```
public IEntityACLEntry addEntry(ACLEntrySubjectType subjectType,  
    String subject)  
    throws StorageConnectorException
```

Adds a new `IEntityACLEntry` specified with the ACL entry subject type and unique subject identifier to the owning entity `IACL`.

**Note:** An entry added in this manner is considered temporary until the owner `IEntity` changes are committed using the [IEntity.save\(\)](#) method.

### Parameters:

subjectType - the ACL entry subject type.  
subject - the ACL entry subject.

### Returns:

An `IEntityACLEntry` instance.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- ACL entry for subject already exists.

---

## addPropertyEntry

```
public IPropertyACLEntry addPropertyEntry(String propertyName,  
    ACLEntrySubjectType subjectType,  
    String subject)  
    throws StorageConnectorException
```

Adds a new `IPropertyACLEntry` specified by the property name, ACL entry subject type and unique subject identifier to the owning entity `IACL`.

**Note:** An entry added in this manner is considered temporary until the owner `IEntity` changes are committed using the [IEntity.save\(\)](#) method.

### Parameters:

propertyName - the property name.  
subjectType - the ACL entry subject type.  
subject - the ACL entry subject.

### Returns:

A `IPropertyACLEntry` instance.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- ACL entry for property and subject already exists.

## com.imis.storageconnector.entitymodel Interface IBinaryValue

public interface **IBinaryValue**  
extends

Represents an unlimited length byte array property which has a read-write interface.

### Method Summary

String	<a href="#">getContentTypes()</a> Gets the binary value content type (MIME).
byte[]	<a href="#">getValue()</a> Gets the value of the IBinaryValue as array of bytes.
InputStream	<a href="#">openDataInputStream()</a> Opens a data input stream from which the binary value data can be read.
OutputStream	<a href="#">openDataOutputStream(boolean append)</a> Opens a data output stream to which the binary value data can be written.
void	<a href="#">setValue(byte[] value)</a> Sets the value of the IBinaryValue as array of bytes.

### Methods

#### getContentTypes

```
public String getContentTypes()
```

Gets the binary value content type (MIME).

**Returns:**

The string representing the value content type (MIME).

#### getValue

```
public byte[] getValue()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the value of the IBinaryValue as array of bytes.

**Note:** This operation allows user to pass a reason for opening binary value data input stream. Reason can be passed through [AuditLog](#) message.

**Returns:**

The array of bytes representing the value of the IBinaryValue.

**Throws:**

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening binary value data stream.

IOException - if an I/O error occurs.

---

## setValue

```
public void setValue(byte[] value)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Sets the value of the `IBinaryValue` as array of bytes.

**Note:** This operation allows user to pass a reason for opening binary value data output stream. Reason can be passed through [AuditLog](#) message.

### Parameters:

value - the array of bytes representing the value of the `IBinaryValue`.

### Throws:

NullPointerException - if value is null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening binary value data stream.

IOException - if an I/O error occurs.

---

## openDataInputStream

```
public InputStream openDataInputStream()
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Opens a data input stream from which the binary value data can be read.

**Note:** This operation allows user to pass a reason for opening binary value data input stream. Reason can be passed through [AuditLog](#) message.

### Returns:

A binary value data `InputStream`.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening property value data input stream.

IOException - if an I/O error occurs.



(continued from last page)

## openDataOutputStream

```
public OutputStream openDataOutputStream(boolean append)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Opens a data output stream to which the binary value data can be written.

**Note:** This operation allows user to pass a reason for opening binary value data output stream. Reason can be passed through [AuditLog](#) message.

### Parameters:

append - true to append data to the end of the output stream; otherwise false.

### Returns:

A binary value data OutputStream.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Cannot open property value in read-write mode on a read-only property.
- Error occurred while opening property value data output stream.

IOException - if an I/O error occurs.

# com.imis.storageconnector.entitymodel

## Interface IClass

All Superinterfaces:

[IEntity](#)

public interface **IClass**

extends [IEntity](#)

Defines a IClass entity.

A IClass entity represents a node in the document classification scheme (tree). It may contain metadata attributes, has its inherited and explicit ACL entries and can contain additional sub-classes, sub-folders and/or documents as sub-entities.

Current implementation inherits from the base IEntity and adds factory methods for creating new instances of sub-entities which can be either Classes, Folders or Documents.

See Also:

[IEntity](#), [IFolder](#), [IDocument](#)

## Method Summary

<a href="#">IClass</a>	<a href="#">createClass</a> (String templateId) Creates a new subclass entity instance.
<a href="#">IDocument</a>	<a href="#">createDocument</a> (String templateId) Creates a new IDocument entity instance.
<a href="#">IFolder</a>	<a href="#">createFolder</a> (String templateId) Creates a new IFolder or sub-folder entity instance.
boolean	<a href="#">isChildClassificationCodeGenerated</a> () Gets a value indicating whether or not child classification code is auto generated.
void	<a href="#">setChildClassificationCodeGenerated</a> (boolean value) Sets a value indicating whether or not child classification code is auto generated.

## Methods inherited from interface [com.imis.storageconnector.entitymodel.IEntity](#)

[accessed](#), [areValuesInherited](#), [close](#), [closed](#), [createBinaryValue](#), [created](#), [createStringMaxValue](#), [getACL](#), [getArchivalInformationalPackage](#), [getArchive](#), [getAuditLog](#), [getCategories](#), [getClassificationCode](#), [getCreator](#), [getDescription](#), [getDraftCreator](#), [getDraftId](#), [getEffectiveRights](#), [getEmailEntity](#), [getEvidenceRecords](#), [getExternalIds](#), [getId](#), [getKeywords](#), [getMode](#), [getModifiedBy](#), [getMoveDetails](#), [getOwner](#), [getParent](#), [getPhysicalEntity](#), [getPickListValues](#), [getProperties](#), [getPublicClassificationCode](#), [getReferences](#), [getReport](#), [getRetention](#), [getSaveLog](#), [getSecurityClass](#), [getSecurityClassChangeDetails](#), [getSignificance](#), [getStatus](#), [getStub](#), [getSubEntities](#), [getSubEntityTemplates](#), [getSystemProperties](#), [getSystemProperty](#), [getTemplate](#), [getTitle](#), [getTransferDetails](#), [getType](#), [getVersionComment](#), [getVersionLabel](#), [getVersionSequence](#), [hasAIP](#), [hasDraft](#), [hasReferences](#), [hasVersions](#), [isClosed](#), [isPermanent](#), [modified](#), [opened](#), [save](#), [save](#), [search](#), [setCategories](#), [setClassificationCode](#), [setDescription](#), [setExternalIds](#), [setKeywords](#), [setOwner](#), [setSecurityClass](#), [setSignificance](#), [setStatus](#), [setTitle](#), [subEntityCount](#)

## Methods inherited from interface [com.imis.IAutoCloseable](#)

---

close

---

## Methods

### isChildClassificationCodeGenerated

```
public boolean isChildClassificationCodeGenerated()
    throws StorageConnectorException
```

Gets a value indicating whether or not child classification code is auto generated.

**Returns:**

true if child classification code is auto generated; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while getting child classification code generated value.

---

### setChildClassificationCodeGenerated

```
public void setChildClassificationCodeGenerated(boolean value)
    throws StorageConnectorException
```

Sets a value indicating whether or not child classification code is auto generated.

**Parameters:**

value - true if child classification code is auto generated; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while setting child classification code generated value.

---

### createClass

```
public IClass createClass(String templateId)
    throws javax.security.auth.login.LoginException,
        StorageConnectorException,
        IOException
```

Creates a new subclass entity instance.

**Note:** The instance is committed to the archive only if [IEntity.save\(\)](#) method is used on the new instance, otherwise it is discarded.

**Note:** This operation allows user to pass a reason for creating class. Reason can be passed through [AuditLog](#) message.

**Parameters:**

templateId - the class template identifier from the templates inventory.

**Returns:**

(continued from last page)

A new IClass instance.

**Throws:**

NullPointerException - if templateId is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on closed archive is not allowed.
- Template identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating entity on archive.

IOException - if an I/O error occurs.

---

## createFolder

```
public IFolder createFolder(String templateId)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Creates a new IFolder or sub-folder entity instance.

**Note:** The instance is committed to the archive only if [IEntity.save\(\)](#) method is used on the new instance, otherwise it is discarded.

**Note:** This operation allows user to pass a reason for creating folder. Reason can be passed through [AuditLog](#) message.

**Parameters:**

templateId - the folder template identifier from the templates inventory.

**Returns:**

A new IFolder instance.

**Throws:**

NullPointerException - if templateId is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on closed archive is not allowed.
- Template identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating entity on archive.

IOException - if an I/O error occurs.

---

## createDocument

```
public IDocument createDocument(String templateId)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

(continued from last page)

Creates a new `IDocument` entity instance.

**Note:** The instance is committed to the archive only if [IEntity.save\(\)](#) method is used on the new instance, otherwise it is discarded.

**Note:** This operation allows user to pass a reason for creating document. Reason can be passed through [AuditLog](#) message.

**Parameters:**

`templateId` - the document template identifier from the templates inventory.

**Returns:**

A new `IDocument` instance.

**Throws:**

`NullPointerException` - if `templateId` is a null reference.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on closed archive is not allowed.
- Template identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating entity on archive.

`IOException` - if an I/O error occurs.

# com.imis.storageconnector.entitymodel

## Interface IContent

All Superinterfaces:

[IReadOnlyContent](#)

public interface **IContent**  
extends [IReadOnlyContent](#)

Defines a content, a collection of content parts.

An **IContent** describes a collection of content parts assigned to an entity. It represents a digital (binary) collection of chapters. It has its attributes and methods for accessing the subsequent binary parts in a read-write fashion. It represents a digital (binary) collection of document chapters. It has its attributes and methods for accessing the subsequent binary parts in a read-write fashion.

**Note:** Any modifications made to the **IContent** is committed to the archive when owner **IDocument** entity is committed with [IEntity.save\(\)](#) method otherwise they are discarded.

**See Also:**

[IEntity](#), [IDocument](#), [IContentPart](#)

### Method Summary

void	<a href="#">clear()</a> Clears the content parts.
void	<a href="#">cloneParts</a> ( <a href="#">IEntity</a> target, String targetContentName, Collection partsIds) Clones the specified content parts to the specified content on the target entity.
void	<a href="#">detachParts</a> (Collection partsIds) Detaches the specified contained content parts and makes them root content parts.
<a href="#">IReadOnlyPropertyRights</a>	<a href="#">getEffectiveRights()</a> Gets the effective rights on the content if content has ACL defined.
void	<a href="#">moveParts</a> ( <a href="#">EntityIdKind</a> targetIdKind, String targetId, String targetContentName, Collection partsIds) Moves the specified root content parts to the specified content on the target entity using the target entity identifier.
void	<a href="#">moveParts</a> ( <a href="#">IEntity</a> target, String targetContentName, Collection partsIds) Moves the specified root content parts to the specified content on the target entity.
void	<a href="#">queuePartsForConvert</a> (Collection partsIds) Queues the specified content parts for conversion.
void	<a href="#">queuePartsForIndex</a> (Collection partsIds) Queues the specified content parts for full-text index.
void	<a href="#">setParts</a> (List parts) Sets the list of <b>IContentPart</b> objects.

Methods inherited from interface [com.imis.storageconnector.entitymodel.IReadOnlyContent](#)

[getEffectiveRights](#), [getName](#), [getParts](#), [isPublic](#), [isReadOnly](#), [isReadOnlyAfterCheckIn](#), [isReadOnlyAfterCreate](#), [isVersionable](#), [partsCount](#)

---

## Methods

### getEffectiveRights

```
public IReadOnlyPropertyRights getEffectiveRights()
```

Gets the effective rights on the content if content has ACL defined.

**Returns:**

The read-only set of rights describing users effective rights on the content or a null reference if none available.

---

### setParts

```
public void setParts(List parts)  
    throws StorageConnectorException
```

Sets the list of IContentPart objects.

This method provides access to the ordered collection of content parts assigned to this content instance. The sequence used when setting the content parts is used as content parts sequence, maintained at the archive.

Deletion of a part is accomplished through setting the content parts collection and not including an IContentPart in the collection.

**Note:** Any modifications to the IContent is committed to the archive when parent entity is saved with [IEntity.save\(\)](#) method, otherwise it is discarded.

**Parameters:**

parts - a list of IContentPart objects.

**Throws:**

NullPointerException - if parts is a null reference.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Content part is not compatible with the archive.
- Error occurred while getting properties.
- Error occurred while setting property values.

---

### cloneParts

```
public void cloneParts(IEntity target,  
    String targetContentName,  
    Collection partsIds)  
    throws StorageConnectorException,  
        IOException
```

Clones the specified content parts to the specified content on the target entity.

This operation requires server version 9.9 or above.

**Note:** This operation allows user to pass a reason for cloning content parts. Reason can be passed through [AuditLog](#) message.

**Parameters:**

target - the target entity.

targetContentName - the target content name.

(continued from last page)

partsIds - the identifiers of the content parts to clone.

**Throws:**

NullPointerException - if target, targetContentName or partsIds is a null reference.  
[StorageConnectorException](#) - if

- Error occurred while getting properties.
- Error occurred while cloning content parts.

IOException - if an I/O error occurs.

---

## detachParts

```
public void detachParts(Collection partsIds)
    throws StorageConnectorException,
           IOException
```

Detaches the specified contained content parts and makes them root content parts.

This operation requires server version 9.8 or above.

**Note:** This operation allows user to pass a reason for detaching content parts. Reason can be passed through [AuditLog](#) message.

**Parameters:**

partsIds - the identifiers of the contained content parts to detach.

**Throws:**

NullPointerException - if partsIds is a null reference.  
[StorageConnectorException](#) - if

- Error occurred while getting properties.
- Error occurred while detaching content parts.

IOException - if an I/O error occurs.

---

## moveParts

```
public void moveParts(IEntity target,
    String targetContentName,
    Collection partsIds)
    throws StorageConnectorException,
           IOException
```

Moves the specified root content parts to the specified content on the target entity.

This operation requires server version 9.9 or above.

**Note:** This operation allows user to pass a reason for moving content parts. Reason can be passed through [AuditLog](#) message.

**Parameters:**

target - the target entity.  
targetContentName - the target content name.  
partsIds - the identifiers of the root content parts to move.

**Throws:**

NullPointerException - if target, targetContentName or partsIds is a null reference.



(continued from last page)

[StorageConnectorException](#) - if

- Error occurred while getting properties.
- Error occurred while moving content parts.

IOException - if an I/O error occurs.

---

## moveParts

```
public void moveParts(EntityIdKind targetIdKind,  
    String targetId,  
    String targetContentName,  
    Collection partsIds)  
throws StorageConnectorException,  
    IOException
```

Moves the specified root content parts to the specified content on the target entity using the target entity identifier.

This operation requires server version 9.8 or above.

**Note:** This operation allows user to pass a reason for moving content parts. Reason can be passed through [AuditLog](#) message.

### Parameters:

targetIdKind - the target entity identifier kind.  
targetId - the target entity identifier.  
targetContentName - the target content name.  
partsIds - the identifiers of the root content parts to move.

### Throws:

NullPointerException - if targetId, targetContentName or partsIds is a null reference.  
[StorageConnectorException](#) - if

- Error occurred while getting properties.
- Error occurred while moving content parts.

IOException - if an I/O error occurs.

---

## queuePartsForConvert

```
public void queuePartsForConvert(Collection partsIds)  
throws StorageConnectorException,  
    IOException
```

Queues the specified content parts for conversion.

This operation requires server version 9.8 or above.

**Note:** This operation allows user to pass a reason for queuing content parts for conversion. Reason can be passed through [AuditLog](#) message.

### Parameters:

partsIds - the identifiers of the content parts to queue for conversion.

### Throws:

NullPointerException - if partsIds is a null reference.

(continued from last page)

[StorageConnectorException](#) - if

- Error occurred while getting properties.
- Error occurred while queuing content parts for conversion.

[IOException](#) - if an I/O error occurs.

---

## queuePartsForIndex

```
public void queuePartsForIndex(Collection partsIds)
    throws StorageConnectorException,
           IOException
```

Queues the specified content parts for full-text index.

This operation requires server version 9.8 or above.

**Note:** This operation allows user to pass a reason for queuing content parts for full-text index. Reason can be passed through [AuditLog](#) message.

### Parameters:

partsIds - the identifiers of the content parts to queue for full-text index.

### Throws:

[NullPointerException](#) - if partsIds is a null reference.  
[StorageConnectorException](#) - if

- Error occurred while getting properties.
- Error occurred while queuing content parts for full-text index.

[IOException](#) - if an I/O error occurs.

---

## clear

```
public void clear()
    throws StorageConnectorException
```

Clears the content parts.

This method does nothing if content had no parts assigned to it.

**Note:** Any modifications done to the [IContent](#) is committed to the archive when parent entity is saved with [IEntity.save\(\)](#) method, otherwise it is discarded.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs: IMIS/ARC Storage Server archive errors:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Error occurred while clearing property values.

## com.imis.storageconnector.entitymodel Interface IContentPart

public interface **IContentPart**  
extends

Defines a part of the IContent.

IContentPart describes a part of a IContent. It represents a digital (binary) chapter of a document. It has its properties which describe its attributes and methods for accessing the binary data.

Any modifications made to the IContentPart is committed to the archive when parent IDocument entity is committed with [IEntity.save\(\)](#) method otherwise they are discarded.

**See Also:**

[IEntity](#), [IDocument](#)

### Method Summary

Calendar	<a href="#">accessed()</a> Gets the content part access timestamp or a null reference for unsaved content part created with the <code>IDocument.createContentPart()</code> method.
void	<a href="#">clone(IEntity target, String targetContentName)</a> Clones this content part to the specified content on the target entity.
Calendar	<a href="#">created()</a> Gets the content part creation timestamp or a null reference for unsaved content part created with the <code>IDocument.createContentPart()</code> method.
void	<a href="#">detach()</a> Detaches this content part and makes it a root content part.
String	<a href="#">getContentTypeId()</a> Gets the content type of this content part.
String	<a href="#">getDescription()</a> Gets the content part description.
String	<a href="#">getExtension()</a> Gets a default extension for the content type of this content part.
String	<a href="#">getId()</a> Gets the content part identifier.
<a href="#">IContentPart</a>	<a href="#">getParent()</a> Gets the parent content part.
long	<a href="#">getSize()</a> Gets the size, in bytes, of the content part.
com.imis.util.IReadOn lyList	<a href="#">getSubParts()</a> Gets the read-only list of sub content parts.
boolean	<a href="#">isIndexed()</a> Gets the value indicating whether or not the content part is indexed.

boolean	<a href="#"><code>isSigned()</code></a> Gets the value indicating whether or not the content part is digitally signed.
Calendar	<a href="#"><code>modified()</code></a> Gets the content part modification timestamp or a null reference for unsaved content part created with the <code>IDocument.createContentPart()</code> method.
void	<a href="#"><code>move(EntityIdKind targetIdKind, String targetId, String targetContentName)</code></a> Moves this content part to the specified content on the target entity using the target entity identifier.
void	<a href="#"><code>move(IEntity target, String targetContentName)</code></a> Moves this content part to the specified content on the target entity.
InputStream	<a href="#"><code>openDataInputStream()</code></a> Opens a new content part data input stream.
OutputStream	<a href="#"><code>openDataOutputStream(boolean append)</code></a> Opens a new content part data output stream.
void	<a href="#"><code>queueForConvert()</code></a> Queues this content part for conversion.
void	<a href="#"><code>queueForIndex()</code></a> Queues this content part for full-text index.
void	<a href="#"><code>setDescription(String value)</code></a> Sets the content part description.

## Methods

### getId

public String **getId**()

Gets the content part identifier.

**Returns:**

The content part identifier.

### getDescription

public String **getDescription**()

Gets the content part description.

**Returns:**

A string containing the content part description.

### setDescription

public void **setDescription**(String value)

Sets the content part description.

**Parameters:**

(continued from last page)

value - a string containing the content part description.

**Throws:**

`NullPointerException` - if value is a null reference.

---

## getExtension

```
public String getExtension()
```

Gets a default extension for the content type of this content part.

**Returns:**

A default extension for the content type of this content part or a null reference if no extension is available.

---

## getContentType

```
public String getContentType()
```

Gets the content type of this content part.

**Returns:**

The content type of this content part.

---

## getSize

```
public long getSize()
```

Gets the size, in bytes, of the content part.

**Returns:**

The size, in bytes, of the content part.

---

## created

```
public Calendar created()
```

Gets the content part creation timestamp or a null reference for unsaved content part created with the `IDocument.createContentPart()` method.

**Returns:**

The date and time when the content part was created or a null reference.

---

## modified

```
public Calendar modified()
```

Gets the content part modification timestamp or a null reference for unsaved content part created with the `IDocument.createContentPart()` method.

**Returns:**

The date and time when the content part was modified or a null reference.

---

## accessed

```
public Calendar accessed()
```

Gets the content part access timestamp or a null reference for unsaved content part created with the `IDocument.createContentPart()` method.

(continued from last page)

**Returns:**

The date and time when the content part was accessed or a null reference.

---

**isIndexed**

```
public boolean isIndexed()
```

Gets the value indicating whether or not the content part is indexed.

**Returns:**

true if the content part is indexed; otherwise false.

---

**isSigned**

```
public boolean isSigned()
```

Gets the value indicating whether or not the content part is digitally signed.

**Returns:**

true if the content part is digitally signed; otherwise false.

---

**getParent**

```
public IContentPart getParent()
```

Gets the parent content part.

**Returns:**

The parent content part or a null reference if this content part has no parent.

---

**getSubParts**

```
public com.imis.util.IReadOnlyList getSubParts()
```

Gets the read-only list of sub content parts.

**Returns:**

The read-only list of sub content parts.

---

**openDataInputStream**

```
public InputStream openDataInputStream()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Opens a new content part data input stream.

This method enables the caller to open a new content part data input stream from which data can be read.

**Note:** Multiple simultaneous and independent input streams can be opened from one instance but only one output stream instance can exist at one time.

**Note:** This operation allows user to pass a reason for opening content part data input stream. Reason can be passed through [AuditLog](#) message.

**Returns:**

A content part data `InputStream`.

**Throws:**

(continued from last page)

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening content part data input stream.

IOException - if an I/O error occurs.

---

## openDataOutputStream

```
public OutputStream openDataOutputStream(boolean append)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Opens a new content part data output stream.

This method enables the caller to open a new content part data input stream to which data can be written.

**Note:** Multiple simultaneous and independent input streams can be opened from one instance but only one output stream instance can exist at one time.

**Note:** This operation allows user to pass a reason for opening content part data output stream. Reason can be passed through [AuditLog](#) message.

### Parameters:

append - true to append data to the end of the output stream; otherwise false.

### Returns:

A content part data OutputStream.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Cannot open content part in output stream on a read-only content.
- Error occurred while opening content part data output stream.

IOException - if an I/O error occurs.

---

## clone

```
public void clone(IEntity target,
                  String targetContentName)
    throws StorageConnectorException,
           IOException
```

Clones this content part to the specified content on the target entity.

This operation requires server version 9.9 or above.

**Note:** This operation allows user to pass a reason for cloning content part. Reason can be passed through [AuditLog](#) message.

### Throws:

NullPointerException - if target or targetContentName is a null reference.

(continued from last page)

[StorageConnectorException](#) - if

- Error occurred while cloning content part.

[IOException](#) - if an I/O error occurs.

---

## detach

```
public void detach()  
    throws StorageConnectorException,  
           IOException
```

Detaches this content part and makes it a root content part.

This operation requires server version 9.8 or above.

**Note:** This operation allows user to pass a reason for detaching content part. Reason can be passed through [AuditLog](#) message.

### Throws:

[StorageConnectorException](#) - if

- Error occurred while detaching content part.

[IOException](#) - if an I/O error occurs.

---

## move

```
public void move(IEntity target,  
                 String targetContentName)  
    throws StorageConnectorException,  
           IOException
```

Moves this content part to the specified content on the target entity.

This operation requires server version 9.9 or above.

**Note:** This operation allows user to pass a reason for moving content part. Reason can be passed through [AuditLog](#) message.

### Parameters:

target - the target entity.

targetContentName - the target content name.

### Throws:

[NullPointerException](#) - if target or targetContentName is a null reference.

[StorageConnectorException](#) - if

- Error occurred while moving content part.

[IOException](#) - if an I/O error occurs.

---

## move

```
public void move(EntityIdKind targetIdKind,  
                 String targetId,  
                 String targetContentName)  
    throws StorageConnectorException,  
           IOException
```



(continued from last page)

Moves this content part to the specified content on the target entity using the target entity identifier.

This operation requires server version 9.8 or above.

**Note:** This operation allows user to pass a reason for moving content part. Reason can be passed through [AuditLog](#) message.

**Parameters:**

targetIdKind - the target entity identifier kind.

targetId - the target entity identifier.

targetContentName - the target content name.

**Throws:**

NullPointerException - if targetId or targetContentName is a null reference.

[StorageConnectorException](#) - if

- Error occurred while moving content part.

IOException - if an I/O error occurs.

---

## queueForConvert

```
public void queueForConvert()  
    throws StorageConnectorException,  
           IOException
```

Queues this content part for conversion.

This operation requires server version 9.8 or above.

**Note:** This operation allows user to pass a reason for queuing content part for conversion. Reason can be passed through [AuditLog](#) message.

**Throws:**

[StorageConnectorException](#) - if

- Error occurred while queuing content part for conversion.

IOException - if an I/O error occurs.

---

## queueForIndex

```
public void queueForIndex()  
    throws StorageConnectorException,  
           IOException
```

Queues this content part for full-text index.

This operation requires server version 9.8 or above.

**Note:** This operation allows user to pass a reason for queuing content part for full-text index. Reason can be passed through [AuditLog](#) message.

**Throws:**

[StorageConnectorException](#) - if

- Error occurred while queuing content part for full-text index.

IOException - if an I/O error occurs.

---

## com.imis.storageconnector.entitymodel Interface IDeletionStub

public interface **IDeletionStub**  
extends

Defines a deleted IEntity descriptor.

An IDeletionStub defines a deleted IEntity descriptor containing the entity title, description and deletion metadata attributes.

### Method Summary

Calendar	<a href="#">deleted()</a> Gets the describing deleted IEntity deletion time-stamp.
<a href="#">IDirectoryEntity</a>	<a href="#">getAgent()</a> Gets the IDirectoryEntity of an agent who deleted the IEntity.
String	<a href="#">getClassificationCode()</a> Gets the describing deleted IEntity classification code in canonical format.
String	<a href="#">getDescription()</a> Gets the describing deleted IEntity description.
String	<a href="#">getId()</a> Gets the describing deleted IEntity unique identifier.
String	<a href="#">getPublicClassificationCode()</a> Gets the describing deleted IEntity classification code in public format.
String	<a href="#">getReason()</a> Gets the describing deleted IEntity reason for deletion.
String	<a href="#">getReference()</a> Gets the describing deleted IEntity reference to an entity (internal, external) which is relevant for deleted IEntity.
<a href="#">IEntityCollection</a>	<a href="#">getSubEntities(EntityFilter filter, Collection sortKeys, Collection categorizationPropertyNames)</a> Gets the read-only collection of deleted child entity stubs of the describing deleted IEntity using the specified entity filter, sort keys and a collection of property names used for categorization.
String	<a href="#">getTitle()</a> Gets the describing deleted IEntity title.
boolean	<a href="#">isContainer()</a> Gets the value indicating whether the describing deleted IEntity represents a container of deleted entities.
long	<a href="#">subEntityCount()</a> Gets the absolute sub-entity count.

### Methods

(continued from last page)

## getId

```
public String getId()
```

Gets the describing deleted `IEntity` unique identifier.

**Returns:**

A string that represents the describing deleted `IEntity` identifier.

---

## isContainer

```
public boolean isContainer()
```

Gets the value indicating whether the describing deleted `IEntity` represents a container of deleted entities.

**Returns:**

`true` if the describing deleted `IEntity` represents a container of deleted entities; otherwise `false`.

---

## getClassificationCode

```
public String getClassificationCode()  
throws StorageConnectorException
```

Gets the describing deleted `IEntity` classification code in canonical format.

**Returns:**

A string that represents the describing deleted `IEntity` classification code or a `null` reference if this deletion stub represents an entity container.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getPublicClassificationCode

```
public String getPublicClassificationCode()  
throws StorageConnectorException
```

Gets the describing deleted `IEntity` classification code in public format.

The entity classification code in canonical format is converted to entity public classification code by replacing the entity type prefix and classification hierarchy delimiter with the respective entity type hierarchy delimiters.

**Returns:**

A string that represents the describing deleted `IEntity` public classification code or a `null` reference if this deletion stub represents an entity container.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

(continued from last page)

---

## getTitle

```
public String getTitle()  
    throws StorageConnectorException
```

Gets the describing deleted IEntity title.

**Returns:**

The describing deleted IEntity title or container title if this deletion stub represents an entity container.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getDescription

```
public String getDescription()  
    throws StorageConnectorException
```

Gets the describing deleted IEntity description.

**Returns:**

The describing deleted IEntity description or container description if this deletion stub represents an entity container.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## deleted

```
public Calendar deleted()  
    throws StorageConnectorException
```

Gets the describing deleted IEntity deletion time-stamp.

**Returns:**

The date and time when the IEntity was deleted or a creation time-stamp if this deletion stub represents an entity container.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getAgent

```
public IDirectoryEntity getAgent()  
    throws StorageConnectorException
```

Gets the IDirectoryEntity of an agent who deleted the IEntity.

**Returns:**

(continued from last page)

The agent `IDirectoryEntity` who deleted the `IEntity` or a creator if this deletion stub represents an entity container.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getReason

```
public String getReason()  
    throws StorageConnectorException
```

Gets the describing deleted `IEntity` reason for deletion.

**Returns:**

A reason for deleting the `IEntity` or a null reference if this deletion stub represents an entity container.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getReference

```
public String getReference()  
    throws StorageConnectorException
```

Gets the describing deleted `IEntity` reference to an entity (internal, external) which is relevant for deleted `IEntity`.

**Returns:**

A reference to an entity (internal, external) which is relevant for deleted `IEntity` or a null reference if this deletion stub represents an entity container.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## subEntityCount

```
public long subEntityCount()
```

Gets the absolute sub-entity count.

**Returns:**

The number of all immediate sub-entities.

---

(continued from last page)

## getSubEntities

```
public IEntityCollection getSubEntities(EntityFilter filter,  
    Collection sortKeys,  
    Collection categorizationPropertyNames)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Gets the read-only collection of deleted child entity stubs of the describing deleted `IEntity` using the specified entity filter, sort keys and a collection of property names used for categorization.

### Parameters:

`filter` - the entity filter or a null reference.

`sortKeys` - the collection of `EntitySortKey` objects.

`categorizationPropertyNames` - the collection of property names used for categorization.

### Returns:

The read-only collection of `IDeletionStub` objects representing the entity children.

### Throws:

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while getting entity child entities on archive.

`IOException` - if an I/O error occurs.

## com.imis.storageconnector.entitymodel Interface IDispositionHold

public interface **IDispositionHold**  
extends

Represents a disposition hold.

### Method Summary

Calendar	<a href="#">accessed()</a> Gets the disposition hold last access time-stamp.
Calendar	<a href="#">created()</a> Gets the disposition hold creation time-stamp.
<a href="#">IDirectoryEntity</a>	<a href="#">getCreator()</a> Gets the disposition hold creator.
String	<a href="#">getDescription()</a> Gets the disposition hold description.
String	<a href="#">getId()</a> Gets the disposition hold internal identifier.
String	<a href="#">getName()</a> Gets the disposition hold name.
String	<a href="#">getReason()</a> Gets the disposition hold reason.
Calendar	<a href="#">modified()</a> Gets the disposition hold last modification time-stamp.

### Methods

#### getId

public String **getId()**

Gets the disposition hold internal identifier.

**Returns:**

A string that represents the disposition hold internal identifier.

#### getName

public String **getName()**  
throws [StorageConnectorException](#)

Gets the disposition hold name.

**Returns:**

(continued from last page)

A string that represents the disposition hold name.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getDescription

```
public String getDescription()  
    throws StorageConnectorException
```

Gets the disposition hold description.

**Returns:**

A string that represents the disposition hold description.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getCreator

```
public IDirectoryEntity getCreator()  
    throws StorageConnectorException
```

Gets the disposition hold creator.

**Returns:**

A [IDirectoryEntity](#) that represents the creator of the disposition hold.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Property value is not allowed based on the current template settings.

---

## created

```
public Calendar created()
```

Gets the disposition hold creation time-stamp.

**Returns:**

The date and time when the disposition hold was created.

---

## modified

```
public Calendar modified()
```

Gets the disposition hold last modification time-stamp.

**Returns:**



(continued from last page)

The date and time when the disposition hold was last modified.

---

## accessed

```
public Calendar accessed()
```

Gets the disposition hold last access time-stamp.

### Returns:

The date and time when the disposition hold was last accessed.

---

## getReason

```
public String getReason()  
    throws StorageConnectorException
```

Gets the disposition hold reason.

### Returns:

A string that represents the disposition hold reason.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

## com.imis.storageconnector.entitymodel Interface IDispositionHoldEntry

public interface **IDispositionHoldEntry**  
extends

Represents a disposition hold entry.

### Method Summary

void	<a href="#">delete()</a> Deletes the retention policy.
<a href="#">IDispositionHold</a>	<a href="#">getDefinition()</a> Gets the disposition hold definition.
boolean	<a href="#">isInherited()</a> Gets the value indicating whether or not the disposition hold entry is inherited.

### Methods

#### isInherited

public boolean **isInherited()**

Gets the value indicating whether or not the disposition hold entry is inherited.

**Returns:**

true the disposition hold entry is inherited; otherwise false.

#### getDefinition

public [IDispositionHold](#) **getDefinition()**  
throws [StorageConnectorException](#)

Gets the disposition hold definition.

**Returns:**

An [IDispositionHoldDefinition](#) object.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.

#### delete

public void **delete()**  
throws [StorageConnectorException](#)

Deletes the retention policy.

(continued from last page)

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed retention is not allowed.
- Operation on inherited disposition hold entry is not allowed.
- Error occurred while removing disposition hold entry.
- Error occurred while reading retention entries.

# com.imis.storageconnector.entitymodel

## Interface IDocument

All Superinterfaces:

[IEntity](#)

public interface **IDocument**

extends [IEntity](#)

Defines a IDocument entity.

A IDocument entity represents a physical document which can contain multiple IContent identified by unique identifiers. It can be contained within a IClass or IFolder entity and it is a final entity in the tree. It may contain its metadata attributes, has its inherited and explicit ACL entries and contains one or more binary objects, grouped by IContent.

Current implementation inherits from the base IEntity and adds factory methods for creating new instances of IContent.

See Also:

[IEntity](#), [IClass](#), [IFolder](#), [IContent](#)

## Method Summary

<a href="#">IContentPart</a>	<a href="#">createContentPart</a> (String contentType) Creates a new <a href="#">IContentPart</a> which represents a new content part assigned to no owner.
<a href="#">IContent</a>	<a href="#">getContent</a> () Gets the document content.
com.imis.util.IReadOnlyMap	<a href="#">getCustomContents</a> () Gets the read-only map of document custom contents.

## Methods inherited from interface [com.imis.storageconnector.entitymodel.IEntity](#)

[accessed](#), [areValuesInherited](#), [close](#), [closed](#), [createBinaryValue](#), [created](#), [createStringMaxValue](#), [getACL](#), [getArchivalInformationalPackage](#), [getArchive](#), [getAuditLog](#), [getCategories](#), [getClassificationCode](#), [getCreator](#), [getDescription](#), [getDraftCreator](#), [getDraftId](#), [getEffectiveRights](#), [getEmailEntity](#), [getEvidenceRecords](#), [getExternalIds](#), [getId](#), [getKeywords](#), [getMode](#), [getModifiedBy](#), [getMoveDetails](#), [getOwner](#), [getParent](#), [getPhysicalEntity](#), [getPickListValues](#), [getProperties](#), [getPublicClassificationCode](#), [getReferences](#), [getReport](#), [getRetention](#), [getSaveLog](#), [getSecurityClass](#), [getSecurityClassChangeDetails](#), [getSignificance](#), [getStatus](#), [getStub](#), [getSubEntities](#), [getSubEntityTemplates](#), [getSystemProperties](#), [getSystemProperty](#), [getTemplate](#), [getTitle](#), [getTransferDetails](#), [getType](#), [getVersionComment](#), [getVersionLabel](#), [getVersionSequence](#), [hasAIP](#), [hasDraft](#), [hasReferences](#), [hasVersions](#), [isClosed](#), [isPermanent](#), [modified](#), [opened](#), [save](#), [save](#), [search](#), [setCategories](#), [setClassificationCode](#), [setDescription](#), [setExternalIds](#), [setKeywords](#), [setOwner](#), [setSecurityClass](#), [setSignificance](#), [setStatus](#), [setTitle](#), [subEntityCount](#)

## Methods inherited from interface [com.imis.IAutoCloseable](#)

[close](#)

## Methods

(continued from last page)

## getContent

```
public IContent getContent()  
    throws StorageConnectorException
```

Gets the document content.

**Returns:**

The document content.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getCustomContents

```
public com.imis.util.IReadOnlyMap getCustomContents()  
    throws StorageConnectorException
```

Gets the read-only map of document custom contents.

**Returns:**

A read-only map of [IContent](#) objects sorted by document content unique identifiers.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.

---

## createContentPart

```
public IContentPart createContentPart(String contentType)  
    throws javax.security.auth.login.LoginException,  
        StorageConnectorException,  
        IOException
```

(continued from last page)

Creates a new [IContentPart](#) which represents a new content part assigned to no owner.

This instance must be added to the `IContent.getParts` container of the appropriate document content and committed to the archive using the [IEntity.save\(\)](#) method otherwise it is discarded.

The content types used are taken from the list of IANA-registered content types.

The following code example demonstrates how to create a content part.

```
// Open a document specified by internal identifier from an archive instance
com.imis.storageconnector.entitymodel.IDocument doc =
archive.openDocument(EntityIdKind.INTERNAL,
    id, AccessMode.READ_WRITE);

// Open a document content named "content1"
com.imis.storageconnector.entitymodel.IContent content =
doc.getContents().get("content1");

// Get the list of document content parts
List<IContentPart> parts = content.getParts();

// Create a content part, open data stream, write data, close data stream
com.imis.storageconnector.entitymodel.IContentPart part =
doc.createContentPart("application/pdf");
...

// Add a content part to the list
parts.add(part);

// Assign the list back to the document content
content.setParts(parts);

// Save document changes
doc.save();
```

**Parameters:**

contentType - the content type (MIME).

**Returns:**

A new `IContentPart` instance.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while creating file value.

`IOException` - if an I/O error occurs.

(continued from last page)

**See Also:**

[IDocument](#)

[IContent](#)

## com.imis.storageconnector.entitymodel Interface IDraft

public interface **IDraft**  
extends com.imis.IAutoCloseable

Defines a IDraft entity.

A IDraft entity represents a draft which can contain multiple IContent identified by unique identifiers.

### Method Summary

Calendar	<a href="#"><u>accessed()</u></a> Gets the draft last access timestamp.
boolean	<a href="#"><u>areValuesInherited(SystemProperty systemProperty)</u></a> Gets the value indicating whether the specified system property values are inherited.
void	<a href="#"><u>close()</u></a> Closes the draft and releases all its resources.
<a href="#"><u>IBinaryValue</u></a>	<a href="#"><u>createBinaryValue(String contentType)</u></a> Creates a new IBinaryValue with no owner which represents a new binary value assigned to no owner.
<a href="#"><u>IContentPart</u></a>	<a href="#"><u>createContentPart(String contentType)</u></a> Creates a new IContentPart which represents a new content part assigned to no owner.
Calendar	<a href="#"><u>created()</u></a> Gets the draft creation timestamp.
<a href="#"><u>IStringMaxValue</u></a>	<a href="#"><u>createStringMaxValue()</u></a> Creates a new IStringMaxValue which represents a new unbounded string property value assigned to no owner.
<a href="#"><u>IACL</u></a>	<a href="#"><u>getACL()</u></a> Gets the draft IACL.
com.imis.util.IReadOnlyList	<a href="#"><u>getCategories()</u></a> Gets the list of draft categories.
<a href="#"><u>IContent</u></a>	<a href="#"><u>getContent()</u></a> Gets the document content.
<a href="#"><u>IDirectoryEntity</u></a>	<a href="#"><u>getCreator()</u></a> Gets the creator IDirectoryEntity.
com.imis.util.IReadOnlyMap	<a href="#"><u>getCustomContents()</u></a> Gets the read-only map of document custom contents.
String	<a href="#"><u>getDescription()</u></a> Gets the IDraft description.
<a href="#"><u>IReadOnlyEntityRights</u></a>	<a href="#"><u>getEffectiveRights()</u></a> Gets the users effective rights on the IDraft.



<a href="#">IEmailEntity</a>	<a href="#">getEmailEntity()</a> Gets the email properties of this IDraft.
String	<a href="#">getId()</a> Gets the draft identifier.
com.imis.util.IReadOnlyList	<a href="#">getKeywords()</a> Gets the list of draft keywords.
<a href="#">IDirectoryEntity</a>	<a href="#">getModifiedBy()</a> Gets the last modifier directory entity.
<a href="#">IDirectoryEntity</a>	<a href="#">getOwner()</a> Gets the owner IDirectoryEntity.
<a href="#">IPhysicalEntity</a>	<a href="#">getPhysicalEntity()</a> Gets the physical entity properties of this IDraft.
com.imis.util.IReadOnlyList	<a href="#">getPickListValues(SystemProperty systemProperty)</a> Gets the read-only list of the pick-list values for the specified system property.
com.imis.util.IReadOnlyMap	<a href="#">getProperties()</a> Gets the read-only map of properties sorted by property unique names.
String	<a href="#">getSourceId()</a> Gets the describing IDraft source entity identifier.
com.imis.util.IReadOnlyMap	<a href="#">getSystemProperties()</a> Gets the read-only map of read-only system properties sorted by SystemProperty enum values.
<a href="#">IReadOnlyProperty</a>	<a href="#">getSystemProperty(SystemProperty systemProperty)</a> Gets the read-only property for the specified system property.
<a href="#">ITemplate</a>	<a href="#">getTemplate()</a> Gets the IDraft template.
String	<a href="#">getTitle()</a> Gets the IDraft title.
Calendar	<a href="#">modified()</a> Gets the draft last modification timestamp.
void	<a href="#">save()</a> Saves the changes and reopens the draft in read-only mode.
void	<a href="#">save(boolean close)</a> Saves the changes and optionally closes the draft or reopens the draft in read-only mode.
void	<a href="#">setCategories(List value)</a> Sets the list of draft categories.
void	<a href="#">setDescription(String value)</a> Sets the IDraft description.
void	<a href="#">setKeywords(List value)</a> Sets the list of draft keywords.
void	<a href="#">setOwner(IDirectoryEntity value)</a> Sets the owner IDirectoryEntity.

void	<a href="#">setTitle</a> (String value) Sets the IDraft title.
------	---

**Methods inherited from interface** `com.imis.IAutoCloseable`

close

## Methods

### getId

```
public String getId()  
    throws StorageConnectorException
```

Gets the draft identifier.

**Returns:**

A string that represents the draft identifier.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.

### getTitle

```
public String getTitle()  
    throws StorageConnectorException
```

Gets the IDraft title.

**Returns:**

The draft title.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

### setTitle

```
public void setTitle(String value)  
    throws StorageConnectorException
```

Sets the IDraft title.

**Parameters:**

value - the draft title.

**Throws:**

[NullPointerException](#) - if value is a null reference.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Operation on draft opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getDescription

```
public String getDescription()  
    throws StorageConnectorException
```

Gets the IDraft description.

### Returns:

The draft description.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setDescription

```
public void setDescription(String value)  
    throws StorageConnectorException
```

Sets the IDraft description.

### Parameters:

value - the draft description.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Operation on draft opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getSourceId

```
public String getSourceId()  
    throws StorageConnectorException
```

Gets the describing IDraft source entity identifier.

### Returns:

The describing IDraft source entity identifier.

### Throws:

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## created

```
public Calendar created()  
    throws StorageConnectorException
```

Gets the draft creation timestamp.

### Returns:

The date and time when the draft was created.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.

---

## modified

```
public Calendar modified()  
    throws StorageConnectorException
```

Gets the draft last modification timestamp.

### Returns:

The date and time when the draft was last modified.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.

---

## getModifiedBy

```
public IDirectoryEntity getModifiedBy()  
    throws StorageConnectorException
```

Gets the last modifier directory entity.

### Returns:

The last modifier directory entity or a null reference, if this instance has no last modifier directory entity.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

(continued from last page)

---

## accessed

```
public Calendar accessed()  
    throws StorageConnectorException
```

Gets the draft last access timestamp.

### Returns:

The date and time when the draft was last accessed.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.

---

## getACL

```
public IACL getACL()  
    throws StorageConnectorException
```

Gets the draft IACL.

The IACL property provides access to an IACL instance containing ACL entries which define access rights for different entity operations.

### Returns:

A IACL instance.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting ACL entries.

---

## getEffectiveRights

```
public IReadOnlyEntityRights getEffectiveRights()  
    throws StorageConnectorException
```

Gets the users effective rights on the IDraft.

### Returns:

The read-only set of rights describing users effective rights on the IDraft.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.

---

## getCreator

```
public IDirectoryEntity getCreator()  
    throws StorageConnectorException
```

Gets the creator IDirectoryEntity.

(continued from last page)

**Returns:**

The creator `IDirectoryEntity`.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getOwner

```
public IDirectoryEntity getOwner()  
    throws StorageConnectorException
```

Gets the owner `IDirectoryEntity`.

**Returns:**

The owner `IDirectoryEntity` or a null reference, if this instance has no owner `IDirectoryEntity`.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setOwner

```
public void setOwner(IDirectoryEntity value)  
    throws StorageConnectorException
```

Sets the owner `IDirectoryEntity`.

**Parameters:**

value - the owner `IDirectoryEntity` or a null reference, if this instance has no owner `IDirectoryEntity`.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Operation on draft opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getEmailEntity

```
public IEmailEntity getEmailEntity()  
    throws StorageConnectorException
```

Gets the email properties of this `IDraft`.

**Returns:**

(continued from last page)

An IEmailEntity instance or a null reference if there are no email properties present.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.

---

## getPhysicalEntity

```
public IPhysicalEntity getPhysicalEntity()  
    throws StorageConnectorException
```

Gets the physical entity properties of this IDraft.

**Returns:**

An IPhysicalEntity instance or a null reference if there are no physical entity properties present.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.

---

## getTemplate

```
public ITemplate getTemplate()  
    throws StorageConnectorException
```

Gets the IDraft template.

**Returns:**

An ITemplate that represents the draft template.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.

---

## getKeywords

```
public com.imis.util.IReadOnlyList getKeywords()  
    throws StorageConnectorException
```

Gets the list of draft keywords.

**Returns:**

The list of draft keywords.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

## setKeywords

```
public void setKeywords(List value)  
    throws StorageConnectorException
```

Sets the list of draft keywords.

### Parameters:

value - the list of draft keywords.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Operation on draft opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getCategories

```
public com.imis.util.IReadOnlyList getCategories()  
    throws StorageConnectorException
```

Gets the list of draft categories.

### Returns:

The list of draft categories.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setCategories

```
public void setCategories(List value)  
    throws StorageConnectorException
```

Sets the list of draft categories.

### Parameters:

value - the list of draft categories.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Operation on draft opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.



## getSystemProperties

```
public com.imis.util.IReadOnlyMap getSystemProperties()  
    throws StorageConnectorException
```

Gets the read-only map of read-only system properties sorted by `SystemProperty` enum values.

### Returns:

The read-only map of `IReadOnlyProperty` objects sorted by `SystemProperty` enum values.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting properties.

---

## getProperties

```
public com.imis.util.IReadOnlyMap getProperties()  
    throws StorageConnectorException
```

Gets the read-only map of properties sorted by property unique names.

This method provides access to collection of properties assigned to the draft. You can iterate and access the different fields of property values with this method. The collection is read-only whereas collection items are not.

If the property is part of a group, a 'group namespace' prefix is used in property name. See [IReadOnlyProperty.getName\(\)](#) method for further info.

### Returns:

The read-only map of `IProperty` objects sorted by property unique names.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting properties.

### See Also:

[IProperty](#)

---

## createStringMaxValue

```
public IStringMaxValue createStringMaxValue()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Creates a new `IStringMaxValue` which represents a new unbounded string property value assigned to no owner.

This instance must be set to the `IProperty` values using the property `setValue()` or `setValues()` methods. `IDraft` must also be committed to the archive using the [save\(\)](#) method otherwise this value is lost.

### Returns:

A new `IStringMaxValue` instance.

### Throws:

(continued from last page)

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[`StorageConnectorException`](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from session pool.
- Error occurred while creating string value.

`IOException` - if an I/O error occurs.

**See Also:**

[`IProperty`](#)

[`PropertyType.STRINGMAX`](#)

---

## createBinaryValue

```
public IBinaryValue createBinaryValue(String contentType)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Creates a new `IBinaryValue` with no owner which represents a new binary value assigned to no owner.

This instance must be set to the `IProperty` values using the property `setValue()` or `setValues()` methods. `IDraft` must also be committed to the archive using the [`save\(\)`](#) method otherwise this value is lost.

The binary value content types used are taken from the list of IANA-registered content types.

**Parameters:**

`contentType` - the binary value content type.

**Returns:**

A new `IBinaryValue` instance.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[`StorageConnectorException`](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from session pool.
- Error occurred while creating binary value.

`IOException` - if an I/O error occurs.

**See Also:**

[`IProperty`](#)

[`PropertyType.BINARY`](#)

---

## getSystemProperty

```
public IReadOnlyProperty getSystemProperty(SystemProperty systemProperty)
    throws StorageConnectorException
```

Gets the read-only property for the specified system property.

**Parameters:**

(continued from last page)

systemProperty - the system property.

**Returns:**

The read-only property for the specified system property.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## areValuesInherited

```
public boolean areValuesInherited(SystemProperty systemProperty)  
    throws StorageConnectorException
```

Gets the value indicating whether the specified system property values are inherited.

**Parameters:**

systemProperty - the system property.

**Returns:**

true if the specified system property values are inherited; otherwise false.

**Throws:**

[IllegalArgumentException](#) - if invalid system property.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getPickListValues

```
public com.imis.util.IReadOnlyList getPickListValues(SystemProperty systemProperty)  
    throws StorageConnectorException
```

Gets the read-only list of the pick-list values for the specified system property.

**Parameters:**

systemProperty - the system property.

**Returns:**

The read-only list of IPickListValue objects representing the pick-list values for the specified system property.

**Throws:**

[IllegalArgumentException](#) - if invalid system property.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Property is not present in template properties.
- Property is not a pick-list property.

## save

```
public void save()  
    throws StorageConnectorException,  
           IOException
```

Saves the changes and reopens the draft in read-only mode.

Calling this method is same as calling [save\(boolean\)](#) with false parameter.

**Note:** This operation allows user to pass a reason for saving draft. Reason can be passed through [AuditLog](#) message.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Operation on draft opened in read-only mode is not allowed.
- Error occurred while committing draft changes on archive.

IOException - if an I/O error occurs.

---

## save

```
public void save(boolean close)  
    throws StorageConnectorException,  
           IOException
```

Saves the changes and optionally closes the draft or reopens the draft in read-only mode.

Call to this method commits all changes made to this draft instance after it was opened in write mode or created.

[getId\(\)](#) method can be used after completion of this routine to access draft unique identifier.

**Note:** This operation allows user to pass a reason for saving draft. Reason can be passed through [AuditLog](#) message.

**Note:** IDraft is closed after call to this method and needs to be reopened if the caller wants to continue to access methods and properties which depend on the opened instance of an draft.

**Parameters:**

close - true to close draft after save; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Operation on draft opened in read-only mode is not allowed.
- Error occurred while committing draft changes on archive.

IOException - if an I/O error occurs.

---

## close

```
public void close()  
    throws StorageConnectorException,  
           IOException
```

Closes the draft and releases all its resources.

**Note:** Any uncommitted changes are lost. Caller cannot use methods or properties which depend on an opened instance of the object.

---

(continued from last page)

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while closing draft on archive.

[IOException](#) - if an I/O error occurs.

---

## getContent

```
public IContent getContent()  
    throws StorageConnectorException
```

Gets the document content.

**Returns:**

The document content.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getCustomContents

```
public com.imis.util.IReadOnlyMap getCustomContents()  
    throws StorageConnectorException
```

Gets the read-only map of document custom contents.

**Returns:**

A read-only map of [IContent](#) objects sorted by document content unique identifiers.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while getting properties.

---

## createContentPart

```
public IContentPart createContentPart(String contentType)  
    throws javax.security.auth.login.LoginException,  
        StorageConnectorException,  
        IOException
```

Creates a new [IContentPart](#) which represents a new content part assigned to no owner.

This instance must be added to the `IContent.getParts` container of the appropriate document content and committed to the archive using the [save\(\)](#) method otherwise it is discarded.

The content types used are taken from the list of IANA-registered content types.

**Parameters:**

`contentType` - the content type (MIME).

(continued from last page)

**Returns:**

A new `IContentPart` instance.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed draft is not allowed.
- Error occurred while creating file value.

`IOException` - if an I/O error occurs.

**See Also:**

[IContent](#)

## com.imis.storageconnector.entitymodel Interface IDraftStub

public interface **IDraftStub**  
extends

Defines an IDraft descriptor.

An IDraftStub defines an IDraft descriptor containing the draft identifier and type, public metadata attributes, callers effective rights on the entity object, etc. It also enables a caller to open a full-feature IDraft object using the `open()` method. Accessing IDraftStub properties and methods other than `open()` will not be recorded in the archive's AuditLog records.

**Note:** Instance is automatically recycled after the last reference is released.

**See Also:**

[IDraft](#)

### Method Summary

Calendar	<a href="#">accessed()</a> Gets the describing IDraft last access timestamp.
boolean	<a href="#">areValuesInherited(SystemProperty systemProperty)</a> Gets the value indicating whether the specified system property values are inherited.
String	<a href="#">checkIn(Short position, String comment)</a> Checks in the draft.
Calendar	<a href="#">created()</a> Gets the describing IDraft creation timestamp.
void	<a href="#">discard()</a> Discards the entity draft.
com.imis.util.IReadOnlyList	<a href="#">getCategories()</a> Gets the list of the describing IDraft categories.
<a href="#">IReadOnlyContent</a>	<a href="#">getContent()</a> Gets the describing IDraft read-only content.
<a href="#">IDirectoryEntity</a>	<a href="#">getCreator()</a> Gets the describing IDraft creator IDirectoryEntity.
com.imis.util.IReadOnlyMap	<a href="#">getCustomContents()</a> Gets the describing IDraft read-only custom contents.
String	<a href="#">getDescription()</a> Gets the describing IDraft description.
<a href="#">IReadOnlyEntityRights</a>	<a href="#">getEffectiveRights()</a> Gets the users effective rights on the describing IDraft.
String	<a href="#">getId()</a> Gets the describing IDraft unique identifier.
com.imis.util.IReadOnlyList	<a href="#">getKeywords()</a> Gets the describing IDraft list of keywords.

<a href="#">IDirectoryEntity</a>	<a href="#">getModifiedBy()</a> Gets the last modifier directory entity.
<a href="#">IDirectoryEntity</a>	<a href="#">getOwner()</a> Gets the describing IDraft owner IDirectoryEntity.
com.imis.util.IReadOnlyList	<a href="#">getPickListValues(SystemProperty systemProperty)</a> Gets the read-only list of the pick-list values for the specified system property.
com.imis.util.IReadOnlyMap	<a href="#">getProperties()</a> Gets the entity public metadata attributes.
String	<a href="#">getSourceId()</a> Gets the describing IDraft source entity identifier.
com.imis.util.IReadOnlyMap	<a href="#">getSystemProperties()</a> Gets the read-only map of read-only system properties sorted by SystemProperty enum values.
<a href="#">IReadOnlyProperty</a>	<a href="#">getSystemProperty(SystemProperty systemProperty)</a> Gets the read-only property for the specified system property.
<a href="#">ITemplate</a>	<a href="#">getTemplate()</a> Gets the describing IDraft template.
String	<a href="#">getTitle()</a> Gets the describing IDraft title.
Calendar	<a href="#">modified()</a> Gets the describing IDraft last modification timestamp.
<a href="#">IDraft</a>	<a href="#">open(AccessMode mode)</a> Opens an IDraft described by this IDraftStub using a specified draft access mode.

## Methods

### getId

public String **getId()**

Gets the describing IDraft unique identifier.

**Returns:**

A string that represents the describing IDraft identifier.

### getTitle

public String **getTitle()**  
throws [StorageConnectorException](#)

Gets the describing IDraft title.

**Returns:**

The describing IDraft title.

**Throws:**



(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getDescription

```
public String getDescription()  
    throws StorageConnectorException
```

Gets the describing IDraft description.

### Returns:

The describing IDraft description.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getSourceId

```
public String getSourceId()  
    throws StorageConnectorException
```

Gets the describing IDraft source entity identifier.

### Returns:

The describing IDraft source entity identifier.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## created

```
public Calendar created()
```

Gets the describing IDraft creation timestamp.

### Returns:

The date and time when the describing entity was created.

---

## modified

```
public Calendar modified()
```

Gets the describing IDraft last modification timestamp.

### Returns:

The date and time when the describing entity was last modified.

(continued from last page)

---

## getModifiedBy

```
public IDirectoryEntity getModifiedBy()
```

Gets the last modifier directory entity.

**Returns:**

The last modifier directory entity or a null reference, if this instance has no last modifier directory entity.

---

## accessed

```
public Calendar accessed()
```

Gets the describing IDraft last access timestamp.

**Returns:**

The date and time when the describing entity was last accessed.

---

## getEffectiveRights

```
public IReadOnlyEntityRights getEffectiveRights()
```

Gets the users effective rights on the describing IDraft.

**Returns:**

The read-only set of rights describing users effective rights on the describing IDraft.

---

## getCreator

```
public IDirectoryEntity getCreator()  
throws StorageConnectorException
```

Gets the describing IDraft creator IDirectoryEntity.

**Returns:**

The creator IDirectoryEntity.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- 

## getOwner

```
public IDirectoryEntity getOwner()  
throws StorageConnectorException
```

Gets the describing IDraft owner IDirectoryEntity.

**Returns:**

The creator IDirectoryEntity or a null reference, if this instance has no owner IDirectoryEntity.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
-

## getTemplate

```
public ITemplate getTemplate()
```

Gets the describing IDraft template.

**Returns:**

A string value that represents the entity template.

---

## getKeywords

```
public com.imis.util.IReadOnlyList getKeywords()  
throws StorageConnectorException
```

Gets the describing IDraft list of keywords.

**Returns:**

The describing IDraft list of keywords.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- 

## getCategories

```
public com.imis.util.IReadOnlyList getCategories()  
throws StorageConnectorException
```

Gets the list of the describing IDraft categories.

**Returns:**

A list of String objects that represents the describing IDraft categories.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- 

## getSystemProperties

```
public com.imis.util.IReadOnlyMap getSystemProperties()
```

Gets the read-only map of read-only system properties sorted by SystemProperty enum values.

**Returns:**

The read-only map of IReadOnlyProperty objects sorted by SystemProperty enum values.

---

## getProperties

```
public com.imis.util.IReadOnlyMap getProperties()
```

Gets the entity public metadata attributes.

---

(continued from last page)

**Returns:**

The read-only map of `IReadOnlyProperty` objects sorted by the property names.

**See Also:**

[IReadOnlyProperty](#)

---

## getContent

```
public IReadOnlyContent getContent()  
    throws StorageConnectorException
```

Gets the describing `IDraft` read-only content.

**Returns:**

The `IReadOnlyContent` object.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getCustomContents

```
public com.imis.util.IReadOnlyMap getCustomContents()
```

Gets the describing `IDraft` read-only custom contents.

**Returns:**

A read-only map of `IReadOnlyContent` objects sorted by the content names.

**See Also:**

[IDocument](#)

[IReadOnlyContent](#)

---

## getSystemProperty

```
public IReadOnlyProperty getSystemProperty(SystemProperty systemProperty)  
    throws StorageConnectorException
```

Gets the read-only property for the specified system property.

**Parameters:**

`systemProperty` - the system property.

**Returns:**

The read-only property for the specified system property.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

(continued from last page)

## areValuesInherited

```
public boolean areValuesInherited(SystemProperty systemProperty)
    throws StorageConnectorException
```

Gets the value indicating whether the specified system property values are inherited.

### Parameters:

systemProperty - the system property.

### Returns:

true if the specified system property values are inherited; otherwise false.

### Throws:

[IllegalArgumentException](#) - if invalid system property.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getPickListValues

```
public com.imis.util.IReadOnlyList getPickListValues(SystemProperty systemProperty)
    throws StorageConnectorException
```

Gets the read-only list of the pick-list values for the specified system property.

### Parameters:

systemProperty - the system property.

### Returns:

The read-only list of IPickListValue objects representing the pick-list values for the specified system property.

### Throws:

[IllegalArgumentException](#) - if invalid system property.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.

---

## open

```
public IDraft open(AccessMode mode)
    throws javax.security.auth.login.LoginException,
        StorageConnectorException,
        IOException
```

Opens an IDraft described by this IDraftStub using a specified draft access mode.

### Parameters:

mode - the entity access mode.

### Returns:

An opened IDraft instance.

(continued from last page)

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening draft on archive.

`IOException` - if an I/O error occurs.

---

## checkIn

```
public String checkIn(Short position,  
    String comment)  
    throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Checks in the draft.

This operation allows user to pass reason for checking in entity draft. Reason can be passed through [AuditLog](#) message.

**Parameters:**

`position` - the check in position in the versioning tree.

`comment` - the entity draft check in comment recorded in version history or a null reference if not specified.

**Returns:**

The checked in entity identifier.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while checking in entity on archive.

`IOException` - if an I/O error occurs.

---

## discard

```
public void discard()  
    throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Discards the entity draft.

This operation allows user to pass reason for discarding entity draft. Reason can be passed through [AuditLog](#) message.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on non draft entity is not allowed.
- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while checking in entity on archive.

`IOException` - if an I/O error occurs.

## com.imis.storageconnector.entitymodel

### Interface IEmailEntity

public interface **IEmailEntity**  
extends

Describes an email entity.

#### Method Summary

List	<a href="#"><u>getBCC()</u></a> Gets the list of email entity blind-copy-to recipient addresses.
List	<a href="#"><u>getCC()</u></a> Gets the list of email entity copy-to recipient addresses.
Calendar	<a href="#"><u>getDate()</u></a> Gets the email entity date and time.
String	<a href="#"><u>getFrom()</u></a> Gets the email entity sender address.
String	<a href="#"><u>getMessageId()</u></a> Gets the email entity message identifier.
String	<a href="#"><u>getPriority()</u></a> Gets the email entity delivery priority.
String	<a href="#"><u>getSubject()</u></a> Gets the email entity subject.
List	<a href="#"><u>getTo()</u></a> Gets the list of email entity recipient addresses.
Boolean	<a href="#"><u>isSigned()</u></a> Gets the value indicating whether or not email entity is digitally signed.
void	<a href="#"><u>setBCC(List value)</u></a> Sets the list of email entity blind-copy-to recipient addresses.
void	<a href="#"><u>setCC(List value)</u></a> Sets the list of email entity copy-to recipient addresses.
void	<a href="#"><u>setDate(Calendar value)</u></a> Sets the email entity date and time.
void	<a href="#"><u>setFrom(String value)</u></a> Sets the email entity sender address.
void	<a href="#"><u>setMessageId(String value)</u></a> Sets the email entity message identifier.
void	<a href="#"><u>setPriority(String value)</u></a> Sets the email entity delivery priority.



void	<a href="#">setSigned</a> (Boolean value) Sets the value indicating whether or not email entity is digitally signed.
void	<a href="#">setSubject</a> (String value) Sets the email entity subject.
void	<a href="#">setTo</a> (List value) Sets the list of email entity recipient addresses.

## Methods

### getMessageId

```
public String getMessageId()
    throws StorageConnectorException
```

Gets the email entity message identifier.

**Returns:**

A string that represents the email entity message identifier.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

### setMessageId

```
public void setMessageId(String value)
    throws StorageConnectorException
```

Sets the email entity message identifier.

**Parameters:**

value - A string that represents the email entity message identifier.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

### getDate

```
public Calendar getDate()
    throws StorageConnectorException
```

Gets the email entity date and time.

(continued from last page)

**Returns:**

The date and time of the email.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setDate

```
public void setDate(Calendar value)  
    throws StorageConnectorException
```

Sets the email entity date and time.

**Parameters:**

value - the date and time of the email.

**Throws:**

[NullPointerException](#) - if value is a null reference.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getFrom

```
public String getFrom()  
    throws StorageConnectorException
```

Gets the email entity sender address.

**Returns:**

A string that represents the email entity sender address.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setFrom

```
public void setFrom(String value)  
    throws StorageConnectorException
```

Sets the email entity sender address.

**Parameters:**

(continued from last page)

value - a string that represents the email entity sender address.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getTo

```
public List getTo()  
    throws StorageConnectorException
```

Gets the list of email entity recipient addresses.

**Returns:**

A list of `String` objects that represents the email entity recipient addresses.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setTo

```
public void setTo(List value)  
    throws StorageConnectorException
```

Sets the list of email entity recipient addresses.

**Parameters:**

value - a list of `String` objects that represents the email entity recipient addresses.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getCC

```
public List getCC()  
    throws StorageConnectorException
```

Gets the list of email entity copy-to recipient addresses.

**Returns:**

(continued from last page)

A list of `String` objects that represents the email entity copy-to recipient addresses.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setCC

```
public void setCC(List value)
    throws StorageConnectorException
```

Sets the list of email entity copy-to recipient addresses.

**Parameters:**

`value` - a list of `String` objects that represents the email entity copy-to recipient addresses.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getBCC

```
public List getBCC()
    throws StorageConnectorException
```

Gets the list of email entity blind-copy-to recipient addresses.

**Returns:**

A list of `String` objects that represents the email entity blind-copy-to recipient addresses.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setBCC

```
public void setBCC(List value)
    throws StorageConnectorException
```

Sets the list of email entity blind-copy-to recipient addresses.

**Parameters:**

`value` - a list of `String` objects that represents the email entity blind-copy-to recipient addresses.

(continued from last page)

**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Operation on entity opened in read-only mode is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
  - Error occurred while setting property values.
- 

## getSubject

```
public String getSubject()  
    throws StorageConnectorException
```

Gets the email entity subject.

**Returns:**

A string that represents the email entity subject.

**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## setSubject

```
public void setSubject(String value)  
    throws StorageConnectorException
```

Sets the email entity subject.

**Parameters:**

value - a string that represents the email entity subject.

**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Operation on entity opened in read-only mode is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
  - Error occurred while setting property values.
- 

## getPriority

```
public String getPriority()  
    throws StorageConnectorException
```

Gets the email entity delivery priority.

**Returns:**

A string that represents the email entity delivery priority.

---

(continued from last page)

**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setPriority

```
public void setPriority(String value)  
    throws StorageConnectorException
```

Sets the email entity delivery priority.

**Parameters:**

value - a string that represents the email entity delivery priority.

**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Property value is not allowed based on the current template settings.
- Error occurred while setting property values.

---

## isSigned

```
public Boolean isSigned()  
    throws StorageConnectorException
```

Gets the value indicating whether or not email entity is digitally signed.

**Returns:**

true if the email entity is digitally signed; otherwise false.

**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setSigned

```
public void setSigned(Boolean value)  
    throws StorageConnectorException
```

Sets the value indicating whether or not email entity is digitally signed.

**Parameters:**

value - true if the email entity is digitally signed; otherwise false.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

## com.imis.storageconnector.entitymodel Interface IEntity

All Subinterfaces:

[IClass](#), [IDocument](#), [IFolder](#)

public interface **IEntity**  
extends com.imis.IAutoCloseable

Describes a generic entity.

There are currently 3 specializations of IEntity objects: IClass, IFolder and IDocument. Each has its specific methods and properties while the IEntity object prescribes the common features each entity contains.

Each instance of IEntity is never instantiated as an object with only a base interface implemented but rather an instance of specialized interface to which it can be cast based on the return value of the getType() method. When IEntity object (or its specialization) is initially created, it is not committed to the archive until appropriate save() method is used. IEntity objects should be closed after use to free its resources.

### Method Summary

Calendar	<a href="#">accessed()</a> Gets the entity last access timestamp.
boolean	<a href="#">areValuesInherited(SystemProperty systemProperty)</a> Gets the value indicating whether the specified system property values are inherited.
void	<a href="#">close()</a> Closes the entity and releases all its resources.
Calendar	<a href="#">closed()</a> Gets the entity status closed time-stamp.
<a href="#">IBinaryValue</a>	<a href="#">createBinaryValue(String contentType)</a> Creates a new IBinaryValue with no owner which represents a new binary value assigned to no owner.
Calendar	<a href="#">created()</a> Gets the entity creation timestamp.
<a href="#">IStringMaxValue</a>	<a href="#">createStringMaxValue()</a> Creates a new IStringMaxValue which represents a new unbounded string property value assigned to no owner.
<a href="#">IACL</a>	<a href="#">getACL()</a> Gets the entity IACL.
InputStream	<a href="#">getArchivalInformationalPackage()</a> Gets the IEntity archival informational package encoded in base 64 format.
<a href="#">IArchive</a>	<a href="#">getArchive()</a> Gets the IArchive where the entity is stored.
<a href="#">AuditLog</a>	<a href="#">getAuditLog()</a> Gets the audit log object from the owning Archive.



com.imis.util.IReadOnlyList	<a href="#">getCategories()</a> Gets the list of entity categories.
String	<a href="#">getClassificationCode()</a> Gets the IEntity classification code in canonical format.
<a href="#">IDirectoryEntity</a>	<a href="#">getCreator()</a> Gets the creator IDirectoryEntity.
String	<a href="#">getDescription()</a> Gets the IEntity description.
<a href="#">IDirectoryEntity</a>	<a href="#">getDraftCreator()</a> Gets the entity's draft creator.
String	<a href="#">getDraftId()</a> Gets the entity's draft identifier.
<a href="#">IReadOnlyEntityRights</a>	<a href="#">getEffectiveRights()</a> Gets the users effective rights on the IEntity.
<a href="#">IEmailEntity</a>	<a href="#">getEmailEntity()</a> Gets the email properties of this IEntity.
com.imis.util.IReadOnlyList	<a href="#">getEvidenceRecords()</a> Gets the IEntity read-only list of evidence records encoded in base 64 format.
com.imis.util.IReadOnlyList	<a href="#">getExternalIds()</a> Gets the list of entity unique external identifiers.
String	<a href="#">getId()</a> Gets the entity identifier.
com.imis.util.IReadOnlyList	<a href="#">getKeywords()</a> Gets the list of entity keywords.
<a href="#">AccessMode</a>	<a href="#">getMode()</a> Gets the entity access mode.
<a href="#">IDirectoryEntity</a>	<a href="#">getModifiedBy()</a> Gets the last modifier directory entity.
com.imis.util.IReadOnlyList	<a href="#">getMoveDetails()</a> Gets the read-only list of move details for the IEntity.
<a href="#">IDirectoryEntity</a>	<a href="#">getOwner()</a> Gets the owner IDirectoryEntity.
<a href="#">IEntityStub</a>	<a href="#">getParent()</a> Gets the parent IEntityStub.
<a href="#">IPhysicalEntity</a>	<a href="#">getPhysicalEntity()</a> Gets the physical entity properties of this IEntity.
com.imis.util.IReadOnlyList	<a href="#">getPickListValues(SystemProperty systemProperty)</a> Gets the read-only list of the pick-list values for the specified system property.
com.imis.util.IReadOnlyMap	<a href="#">getProperties()</a> Gets the read-only map of properties sorted by property unique names.

String	<a href="#"><u>getPublicClassificationCode()</u></a> Gets the IEntity classification code in public format.
com.imis.util.IReadOnlyList	<a href="#"><u>getReferences()</u></a> Gets the read-only list of IReadOnlyEntityReference objects.
InputStream	<a href="#"><u>getReport(ReportType type)</u></a> Gets the specified report data stream.
<a href="#"><u>IRetention</u></a>	<a href="#"><u>getRetention(AccessMode mode)</u></a> Gets the retention for the entity.
com.imis.util.IReadOnlyList	<a href="#"><u>getSaveLog()</u></a> Gets the read-only list of strings containing commit log reports for IEntity in XML format encoded per XML processing instruction.
String	<a href="#"><u>getSecurityClass()</u></a> Gets the IEntity security class.
com.imis.util.IReadOnlyList	<a href="#"><u>getSecurityClassChangeDetails()</u></a> Gets the read-only list of security class change details for the IEntity.
String	<a href="#"><u>getSignificance()</u></a> Gets the IEntity significance.
String	<a href="#"><u>getStatus()</u></a> Gets the IEntity status.
<a href="#"><u>IEntityStub</u></a>	<a href="#"><u>getStub()</u></a> Gets the describing IEntity.
<a href="#"><u>IEntityCollection</u></a>	<a href="#"><u>getSubEntities(EntityFilter filter, Collection sortKeys, Collection categorizationPropertyNames)</u></a> Gets the read-only collection of child entity stubs of the entity using the specified entity filter, sort keys and a collection of property names used for categorization.
com.imis.util.IReadOnlyList	<a href="#"><u>getSubEntityTemplates()</u></a> Gets the read-only list of templates that represent the IEntity sub-entity creation templates.
com.imis.util.IReadOnlyMap	<a href="#"><u>getSystemProperties()</u></a> Gets the read-only map of read-only system properties sorted by SystemProperty enum values.
<a href="#"><u>IReadOnlyProperty</u></a>	<a href="#"><u>getSystemProperty(SystemProperty systemProperty)</u></a> Gets the read-only property for the specified system property.
<a href="#"><u>ITemplate</u></a>	<a href="#"><u>getTemplate()</u></a> Gets the IEntity template identifier.
String	<a href="#"><u>getTitle()</u></a> Gets the IEntity title.
<a href="#"><u>ITransferDetails</u></a>	<a href="#"><u>getTransferDetails()</u></a> Gets the transferred details properties of this IEntity.
<a href="#"><u>EntityType</u></a>	<a href="#"><u>getType()</u></a> Gets the entity type identifier.

String	<a href="#"><code>getVersionComment()</code></a> Gets the entity check in comment.
String	<a href="#"><code>getVersionLabel()</code></a> Gets the label of the entity version.
Integer	<a href="#"><code>getVersionSequence()</code></a> Gets the entity version sequence.
boolean	<a href="#"><code>hasAIP()</code></a> Gets a value indicating whether or not the entity has AIP.
boolean	<a href="#"><code>hasDraft()</code></a> Gets the value indicating whether or not the entity has a draft.
boolean	<a href="#"><code>hasReferences()</code></a> Gets the value indicating whether or not the entity has references.
boolean	<a href="#"><code>hasVersions()</code></a> Gets the value indicating whether or not the entity has versions.
boolean	<a href="#"><code>isClosed()</code></a> Gets a value indicating whether the entity is closed.
boolean	<a href="#"><code>isPermanent()</code></a> Gets a value indicating whether the entity is permanent.
Calendar	<a href="#"><code>modified()</code></a> Gets the entity last modification timestamp.
Calendar	<a href="#"><code>opened()</code></a> Gets the entity status opened time-stamp.
void	<a href="#"><code>save()</code></a> Saves the changes and reopens the entity in read-only mode.
void	<a href="#"><code>save(boolean close)</code></a> Saves the changes and optionally closes the entity or reopens the entity in read-only mode.
<a href="#"><code>IEntityCollection</code></a>	<a href="#"><code>search(EntityFilter filter, Collection sortKeys, Set options, String searchExpression, Collection categorizationPropertyNames, Long maxElements)</code></a> Searches for child entity stubs of the entity using the specified entity search filter, sort keys, search options, expression string and a collection of property names used for categorization.
void	<a href="#"><code>setCategories(List value)</code></a> Sets the list of entity categories.
void	<a href="#"><code>setClassificationCode(String value)</code></a> Sets the IEntity classification code in canonical format.
void	<a href="#"><code>setDescription(String value)</code></a> Sets the IEntity description.
void	<a href="#"><code>setExternalIds(List value)</code></a> Sets the list of external entity identifiers.
void	<a href="#"><code>setKeywords(List value)</code></a> Sets the list of entity keywords.

void	<a href="#"><code>setOwner</code></a> ( <a href="#"><code>IDirectoryEntity</code></a> value) Gets the owner <code>IDirectoryEntity</code> .
void	<a href="#"><code>setSecurityClass</code></a> (String value) Sets the <code>IEntity</code> security class.
void	<a href="#"><code>setSignificance</code></a> (String value) Sets the <code>IEntity</code> significance.
void	<a href="#"><code>setStatus</code></a> (String value) Sets the <code>IEntity</code> status.
void	<a href="#"><code>setTitle</code></a> (String value) Sets the <code>IEntity</code> title.
long	<a href="#"><code>subEntityCount</code></a> () Gets the absolute sub-entity count.

#### Methods inherited from interface `com.imis.IAutoCloseable`

`close`

## Methods

### **getStub**

public [`IEntityStub`](#) **getStub**()

Gets the describing `IEntity`.

**Returns:**

The `IEntityStub` instance.

### **getId**

public String **getId**()  
throws [`StorageConnectorException`](#)

Gets the entity identifier.

**Returns:**

A string that represents the entity identifier.

**Throws:**

[`StorageConnectorException`](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on unsaved new entity is not allowed.

### **getClassificationCode**

public String **getClassificationCode**()  
throws [`StorageConnectorException`](#)

Gets the `IEntity` classification code in canonical format.

(continued from last page)

**Returns:**

A string that represents the entity classification code.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on unsaved new entity is not allowed.
- Error occurred while getting classification code.

---

## setClassificationCode

```
public void setClassificationCode(String value)  
    throws StorageConnectorException
```

Sets the IEntity classification code in canonical format.

**Parameters:**

value - a string that represents the entity classification code.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation is only allowed on a new entity.
- Error occurred while setting classification code.

---

## getPublicClassificationCode

```
public String getPublicClassificationCode()  
    throws StorageConnectorException
```

Gets the IEntity classification code in public format.

The entity classification code in canonical format is converted to entity public classification code by replacing the entity type prefix and classification hierarchy delimiter with the respective entity type hierarchy delimiters.

**Returns:**

A string that represents the entity public classification code.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on unsaved new entity is not allowed.
- Error occurred while getting classification code.

---

## getTitle

```
public String getTitle()  
    throws StorageConnectorException
```

Gets the IEntity title.

(continued from last page)

**Returns:**

The entity title.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setTitle

```
public void setTitle(String value)  
    throws StorageConnectorException
```

Sets the IEntity title.

**Parameters:**

value - the entity title.

**Throws:**

NullPointerException - if value is a null reference.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getDescription

```
public String getDescription()  
    throws StorageConnectorException
```

Gets the IEntity description.

**Returns:**

The entity description.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setDescription

```
public void setDescription(String value)  
    throws StorageConnectorException
```

Sets the IEntity description.

**Parameters:**

(continued from last page)

value - the entity description.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getSignificance

```
public String getSignificance()  
    throws StorageConnectorException
```

Gets the IEntity significance.

**Returns:**

The entity significance or a null reference if no entity significance is set.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setSignificance

```
public void setSignificance(String value)  
    throws StorageConnectorException
```

Sets the IEntity significance.

**Parameters:**

value - the entity significance, a null reference to reset entity significance to inherited value or an empty string to explicitly clear the entity significance value.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getStatus

```
public String getStatus()  
    throws StorageConnectorException
```

Gets the IEntity status.

(continued from last page)

**Returns:**

The entity status.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setStatus

```
public void setStatus(String value)  
    throws StorageConnectorException
```

Sets the IEntity status.

**Note:** Use this method on new unsaved entities only. Change status of existing entities with [IArchive.setEntityStatus\(EntityIdKind, String, String\)](#) method.

**Parameters:**

value - the entity status or a null reference to clear the entity status to inherited value.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on saved entity is not allowed.
- Property is not present in template properties.
- Property value is not allowed based on the current template settings.
- Error occurred while setting property values.

---

## getType

```
public EntityType getType()  
    throws StorageConnectorException
```

Gets the entity type identifier.

**Returns:**

An enumerator value that represents the entity type.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

---

## getMode

```
public AccessMode getMode()  
    throws StorageConnectorException
```

Gets the entity access mode.

**Returns:**



(continued from last page)

The entity access mode.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

---

## created

```
public Calendar created()  
    throws StorageConnectorException
```

Gets the entity creation timestamp.

**Returns:**

The date and time when the entity was created.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

---

## modified

```
public Calendar modified()  
    throws StorageConnectorException
```

Gets the entity last modification timestamp.

**Returns:**

The date and time when the entity was last modified.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

---

## getModifiedBy

```
public IDirectoryEntity getModifiedBy()  
    throws StorageConnectorException
```

Gets the last modifier directory entity.

**Returns:**

The last modifier directory entity or a null reference, if this instance has no last modifier directory entity.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

## accessed

```
public Calendar accessed()  
    throws StorageConnectorException
```

Gets the entity last access timestamp.

**Returns:**

The date and time when the entity was last accessed.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- 

## opened

```
public Calendar opened()  
    throws StorageConnectorException
```

Gets the entity status opened time-stamp.

**Returns:**

The date and time when the entity status was set to opened or a null reference if entity has no status.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## closed

```
public Calendar closed()  
    throws StorageConnectorException
```

Gets the entity status closed time-stamp.

**Returns:**

The date and time when the entity status was set to closed or a null reference if entity has no status.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## isPermanent

```
public boolean isPermanent()  
    throws StorageConnectorException
```

---

(continued from last page)

Gets a value indicating whether the entity is permanent.

**Returns:**

true if the entity is permanent; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

---

## hasAIP

```
public boolean hasAIP()  
    throws StorageConnectorException
```

Gets a value indicating whether or not the entity has AIP.

**Returns:**

true if the entity is has AIP; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

---

## hasDraft

```
public boolean hasDraft()  
    throws StorageConnectorException
```

Gets the value indicating whether or not the entity has a draft.

**Returns:**

true if the entity has a draft; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

---

## getDraftId

```
public String getDraftId()  
    throws StorageConnectorException
```

Gets the entity's draft identifier.

**Returns:**

The entity's draft identifier.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- 

## getDraftCreator

```
public IDirectoryEntity getDraftCreator()  
    throws StorageConnectorException
```

Gets the entity's draft creator.

**Returns:**

The entity's draft creator.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Error occurred while getting directory member on archive.
- 

## hasVersions

```
public boolean hasVersions()  
    throws StorageConnectorException
```

Gets the value indicating whether or not the entity has versions.

**Returns:**

true if the entity has versions; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- 

## getVersionSequence

```
public Integer getVersionSequence()  
    throws StorageConnectorException
```

Gets the entity version sequence.

**Returns:**

The entity version sequence.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
-

(continued from last page)

---

## getVersionLabel

```
public String getVersionLabel()  
    throws StorageConnectorException
```

Gets the label of the entity version.

**Returns:**

A string that represents the label of the entity version.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getVersionComment

```
public String getVersionComment()  
    throws StorageConnectorException
```

Gets the entity check in comment.

**Returns:**

The entity check in comment.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## isClosed

```
public boolean isClosed()
```

Gets a value indicating whether the entity is closed.

**Returns:**

true if the entity is closed; otherwise false.

---

## subEntityCount

```
public long subEntityCount()  
    throws StorageConnectorException
```

Gets the absolute sub-entity count.

**Returns:**

The number of all immediate sub-entities.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- 

## getACL

```
public IACL getACL()  
    throws StorageConnectorException
```

Gets the entity IACL.

The IACL property provides access to an IACL instance containing ACL entries which define access rights for different entity operations.

### Returns:

A IACL instance.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Error occurred while getting ACL entries.
- 

## getEffectiveRights

```
public IReadOnlyEntityRights getEffectiveRights()  
    throws StorageConnectorException
```

Gets the users effective rights on the IEntity.

### Returns:

The read-only set of rights describing users effective rights on the IEntity.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- 

## getSecurityClass

```
public String getSecurityClass()  
    throws StorageConnectorException
```

Gets the IEntity security class.

### Returns:

The IEntity security class system attribute enumeration value or a null reference if security class is inherited from parent entity or to clear the entity security class.

### Throws:

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setSecurityClass

```
public void setSecurityClass(String value)  
    throws StorageConnectorException
```

Sets the IEntity security class.

**Note:** Use this method on new unsaved entities only. Change security class of existing entities with [IArchive.setEntitySecurityClass\(EntityIdKind, String, String\)](#) method.

### Parameters:

value - the IEntity security class system attribute enumeration value or a null reference to clear the entity security class to inherited value.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on saved entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Property value is not allowed based on the current template settings.
- Error occurred while setting property values.

---

## getArchive

```
public IArchive getArchive()
```

Gets the IArchive where the entity is stored.

### Returns:

The IArchive where the entity is stored.

---

## getParent

```
public IEntityStub getParent()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the parent IEntityStub.

### Returns:

The parent IEntityStub.

### Throws:

[CancellationException](#) - if an operation was canceled by the user.  
[LoginException](#) - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting parent entity stub.

[IOException](#) - if an I/O error occurs.

---

## getCreator

```
public IDirectoryEntity getCreator()  
    throws StorageConnectorException
```

Gets the creator [IDirectoryEntity](#).

### Returns:

The creator [IDirectoryEntity](#).

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getOwner

```
public IDirectoryEntity getOwner()  
    throws StorageConnectorException
```

Gets the owner [IDirectoryEntity](#).

### Returns:

The owner [IDirectoryEntity](#) or a null reference, if this instance has no owner [IDirectoryEntity](#).

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setOwner

```
public void setOwner(IDirectoryEntity value)  
    throws StorageConnectorException
```

Gets the owner [IDirectoryEntity](#).

### Parameters:

value - the owner [IDirectoryEntity](#) or a null reference, if this instance has no owner [IDirectoryEntity](#).

### Throws:



(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getSaveLog

```
public com.imis.util.IReadOnlyList getSaveLog()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the read-only list of strings containing commit log reports for IEntity in XML format encoded per XML processing instruction.

### Returns:

A read-only list of strings containing commit log reports in XML format encoded per XML processing instruction.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

IOException - if an I/O error occurs.

---

## getSecurityClassChangeDetails

```
public com.imis.util.IReadOnlyList getSecurityClassChangeDetails()  
    throws StorageConnectorException
```

Gets the read-only list of security class change details for the IEntity.

### Returns:

A read-only list of ISecurityClassChangeDetails objects.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getMoveDetails

```
public com.imis.util.IReadOnlyList getMoveDetails()  
    throws StorageConnectorException
```

Gets the read-only list of move details for the IEntity.

(continued from last page)

**Returns:**

A read-only list of `IMoveDetails` objects.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## getEmailEntity

```
public IEmailEntity getEmailEntity()  
    throws StorageConnectorException
```

Gets the email properties of this `IEntity`.

**Returns:**

An `IEmailEntity` instance or a null reference if there are no email properties present.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- 

## getPhysicalEntity

```
public IPhysicalEntity getPhysicalEntity()  
    throws StorageConnectorException
```

Gets the physical entity properties of this `IEntity`.

**Returns:**

An `IPhysicalEntity` instance or a null reference if there are no physical entity properties present.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- 

## getTransferDetails

```
public ITransferDetails getTransferDetails()  
    throws StorageConnectorException
```

Gets the transferred details properties of this `IEntity`.

**Returns:**

An `ITransferDetails` instance or a null reference if there are no transferred details properties present.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

---

## getTemplate

```
public ITemplate getTemplate()  
    throws StorageConnectorException
```

Gets the IEntity template identifier.

**Returns:**

An ITemplate that represents the entity template.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

---

## getAuditLog

```
public AuditLog getAuditLog()  
    throws StorageConnectorException
```

Gets the audit log object from the owning Archive.

This property provides access to the AuditLog interface which provides a facility for notify archiving system about the cause for operations on entities executed from this IEntity instance.

**Returns:**

The AuditLog instance.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.

**See Also:**

[AuditLog](#)

---

## getArchivalInformationalPackage

```
public InputStream getArchivalInformationalPackage()  
    throws StorageConnectorException,  
           IOException
```

Gets the IEntity archival informational package encoded in base 64 format.

**Returns:**

An input stream containing entity archival informational package encoded in base 64 format.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting archival informational package.

[IOException](#) - if an I/O error occurs.

---

## getEvidenceRecords

```
public com.imis.util.IReadOnlyList getEvidenceRecords()  
    throws StorageConnectorException,  
           IOException
```

Gets the IEntity read-only list of evidence records encoded in base 64 format.

### Returns:

A read-only list of `InputStream` objects containing entity evidence records encoded in base 64 format.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting evidence records.

[IOException](#) - if an I/O error occurs.

---

## getKeywords

```
public com.imis.util.IReadOnlyList getKeywords()  
    throws StorageConnectorException
```

Gets the list of entity keywords.

### Returns:

The list of entity keywords.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setKeywords

```
public void setKeywords(List value)  
    throws StorageConnectorException
```

Sets the list of entity keywords.

### Parameters:

`value` - the list of entity keywords.

### Throws:

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Operation on entity opened in read-only mode is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
  - Error occurred while setting property values.
- 

## getExternalIds

```
public com.imis.util.IReadOnlyList getExternalIds()  
    throws StorageConnectorException
```

Gets the list of entity unique external identifiers.

**Returns:**

The list of entity external identifiers.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## setExternalIds

```
public void setExternalIds(List value)  
    throws StorageConnectorException
```

Sets the list of external entity identifiers.

**Parameters:**

value - the list of entity external identifiers.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Operation on entity opened in read-only mode is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
  - Error occurred while setting property values.
- 

## getCategories

```
public com.imis.util.IReadOnlyList getCategories()  
    throws StorageConnectorException
```

Gets the list of entity categories.

**Returns:**

The list of entity categories.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## setCategories

```
public void setCategories(List value)  
    throws StorageConnectorException
```

Sets the list of entity categories.

### Parameters:

value - the list of entity categories.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Operation on entity opened in read-only mode is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
  - Error occurred while setting property values.
- 

## getSystemProperties

```
public com.imis.util.IReadOnlyMap getSystemProperties()  
    throws StorageConnectorException
```

Gets the read-only map of read-only system properties sorted by SystemProperty enum values.

### Returns:

The read-only map of IReadOnlyProperty objects sorted by SystemProperty enum values.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Error occurred while getting properties.
- 

## getProperties

```
public com.imis.util.IReadOnlyMap getProperties()  
    throws StorageConnectorException
```

Gets the read-only map of properties sorted by property unique names.

This method provides access to collection of properties assigned to the entity. You can iterate and access the different fields of property values with this method. The collection is read-only whereas collection items are not.

If the property is part of a group, a 'group namespace' prefix is used in property name. See [IReadOnlyProperty.getName\(\)](#) method for further info.

### Returns:

The read-only map of IProperty objects sorted by property unique names.

---

(continued from last page)

**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.

**See Also:**[IProperty](#)

---

## getSubEntityTemplates

```
public com.imis.util.IReadOnlyList getSubEntityTemplates()  
    throws StorageConnectorException
```

Gets the read-only list of templates that represent the IEntity sub-entity creation templates.

**Returns:**

The read-only list of ITemplate objects that represent the entity sub-entity creation templates.

**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

---

## hasReferences

```
public boolean hasReferences()  
    throws StorageConnectorException
```

Gets the value indicating whether or not the entity has references.

**Returns:**

true if the entity has references; otherwise false.

**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

---

## getReferences

```
public com.imis.util.IReadOnlyList getReferences()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the read-only list of IReadOnlyEntityReference objects.

**Returns:**

The read-only list of IReadOnlyEntityReference objects.

**Throws:**

CancellationException - if an operation was canceled by the user.  
LoginException - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.

IOException - if an I/O error occurs.

---

## getRetention

```
public IRetention getRetention(AccessMode mode)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Gets the retention for the entity.

**Note:** This operation allows user to pass a reason for opening retention. Reason can be passed through [AuditLog](#) message.

### Parameters:

mode - the retention access mode.

### Returns:

An [IRetention](#) instance.

### Throws:

[CancellationException](#) - if an operation was canceled by the user.

[LoginException](#) - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while opening entity retention.

IOException - if an I/O error occurs.

---

## createStringMaxValue

```
public IStringMaxValue createStringMaxValue()
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```



(continued from last page)

Creates a new `IStringMaxValue` which represents a new unbounded string property value assigned to no owner.

This instance must be set to the `IProperty` values using the property `setValue()` or `setValues()` methods. `IEntity` must also be committed to the archive using the [save\(\)](#) method otherwise this value is lost.

The following code example demonstrates how to create a string value on a class property.

```
// Open a class specified by classification code from an archive instance
com.imis.storageconnector.entitymodel.IClass cls =
archive.openClass(EntityIdKind.CLASSIFICATION_CODE,
    id, AccessMode.READ_WRITE);

// Open a class property named "strmax1"
com.imis.storageconnector.entitymodel.IProperty property =
cls.getProperties().get("strmax1");

// Create a string value, open data stream, write data, close data stream
com.imis.storageconnector.entitymodel.IStringMaxValue strmax =
cls.createStringMaxValue();
...

// Set a string value to the property
property.setValue(strmax);

// Save class changes
cls.save();
```

**Returns:**

A new `IStringMaxValue` instance.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from session pool.
- Error occurred while creating string value.

`IOException` - if an I/O error occurs.

**See Also:**

[IEntity](#)

[IProperty](#)

[PropertyType.STRINGMAX](#)

---

(continued from last page)

## createBinaryValue

```
public IBinaryValue createBinaryValue(String contentType)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Creates a new [IBinaryValue](#) with no owner which represents a new binary value assigned to no owner.

This instance must be set to the [IProperty](#) values using the property [setValue\(\)](#) or [setValues\(\)](#) methods. [IEntity](#) must also be committed to the archive using the [save\(\)](#) method otherwise this value is lost.

The binary value content types used are taken from the list of IANA-registered content types.

The following code example demonstrates how to create a binary value on a class property.

```
// Open a class specified by classification code from an archive instance
com.imis.storageconnector.entitymodel.IClass cls =
archive.openClass(EntityIdKind.CLASSIFICATION_CODE,
    id, AccessMode.READ_WRITE);

// Open a class property named "bin1"
com.imis.storageconnector.entitymodel.IProperty property =
cls.getProperties().get("bin1");

// Create a binary value, open data stream, write data, close data stream
com.imis.storageconnector.entitymodel.IBinaryValue binary =
archive.createBinaryValue("image/tiff");
...

// Set a binary value to the property
property.setValue(binary);

// Save class changes
cls.save();
```

### Parameters:

contentType - the binary value content type.

### Returns:

A new [IBinaryValue](#) instance.

### Throws:

[CancellationException](#) - if an operation was canceled by the user.

[LoginException](#) - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from session pool.
- Error occurred while creating binary value.

IOException - if an I/O error occurs.

**See Also:**[IEntity](#)[IProperty](#)[PropertyType.BINARY](#)

---

## getSubEntities

```
public IEntityCollection getSubEntities(EntityFilter filter,
    Collection sortKeys,
    Collection categorizationPropertyNames)
    throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Gets the read-only collection of child entity stubs of the entity using the specified entity filter, sort keys and a collection of property names used for categorization.

**Parameters:**

filter - the entity filter or a null reference.

sortKeys - the collection of EntitySortKey objects.

categorizationPropertyNames - the collection of property names used for categorization.

**Returns:**

The read-only collection of IEntityStub objects representing the entity children.

**Throws:**

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting entity child entities on archive.

IOException - if an I/O error occurs.

**See Also:**[EntityTypeFilter](#)[EntitySortKey](#)

---

## getSystemProperty

```
public IReadOnlyProperty getSystemProperty(SystemProperty systemProperty)
    throws StorageConnectorException
```

Gets the read-only property for the specified system property.

**Parameters:**

systemProperty - the system property.

**Returns:**

(continued from last page)

The read-only property for the specified system property.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## areValuesInherited

```
public boolean areValuesInherited(SystemProperty systemProperty)  
    throws StorageConnectorException
```

Gets the value indicating whether the specified system property values are inherited.

**Parameters:**

systemProperty - the system property.

**Returns:**

true if the specified system property values are inherited; otherwise false.

**Throws:**

[IllegalArgumentException](#) - if invalid system property.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getPickListValues

```
public com.imis.util.IReadOnlyList getPickListValues(SystemProperty systemProperty)  
    throws StorageConnectorException
```

Gets the read-only list of the pick-list values for the specified system property.

**Parameters:**

systemProperty - the system property.

**Returns:**

The read-only list of IPickListValue objects representing the pick-list values for the specified system property.

**Throws:**

[IllegalArgumentException](#) - if invalid system property.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Property is not present in template properties.
- Property is not a pick-list property.

(continued from last page)

## search

```
public ICollection search(EntityFilter filter,
    Collection sortKeys,
    Set options,
    String searchExpression,
    Collection categorizationPropertyNames,
    Long maxElements)
throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Searches for child entity stubs of the entity using the specified entity search filter, sort keys, search options, expression string and a collection of property names used for categorization.

### Parameters:

`filter` - the entity search filter or a null reference.  
`sortKeys` - the collection of `EntitySortKey` objects.  
`options` - the search options.  
`searchExpression` - the search expression string.  
`categorizationPropertyNames` - the collection of property names used for categorization.  
`maxElements` - the number of elements the resulting collection should be truncated to.

### Returns:

The read-only collection of `ISearchedEntityStub` objects representing the search results.

### Throws:

`CancellationException` - if an operation was canceled by the user.  
`LoginException` - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on unsaved new entity is not allowed.
- Error occurred while performing search on entity.

`IOException` - if an I/O error occurs.

## getReport

```
public InputStream getReport(ReportType type)
throws StorageConnectorException
```

Gets the specified report data stream.

**Note:** Valid report type on `IEntity` is:

- [ReportType.ENTITY\\_SAVE](#)

### Parameters:

`type` - the report type.

### Returns:

The report data stream.

### Throws:

`UnsupportedOperationException` - if report not supported.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on unsaved new entity is not allowed.
- Error occurred while getting entity report.

---

## save

```
public void save()  
    throws StorageConnectorException,  
           IOException
```

Saves the changes and reopens the entity in read-only mode.

Calling this method is same as calling [save\(boolean\)](#) with false parameter.

**Note:** This operation allows user to pass a reason for saving entity. Reason can be passed through [AuditLog](#) message.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while committing entity changes on archive.

IOException - if an I/O error occurs.

---

## save

```
public void save(boolean close)  
    throws StorageConnectorException,  
           IOException
```

Saves the changes and optionally closes the entity or reopens the entity in read-only mode.

Call to this method commits all changes made to this entity instance after it was opened in write mode or created.

[getId\(\)](#) method can be used after completion of this routine to access entity unique identifier.

**Note:** This operation allows user to pass a reason for saving entity. Reason can be passed through [AuditLog](#) message.

**Note:** IEntity is closed after call to this method and needs to be reopened if the caller wants to continue to access methods and properties which depend on the opened instance of an entity.

### Parameters:

close - true to close entity after save; otherwise false.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while committing entity changes on archive.

IOException - if an I/O error occurs.

(continued from last page)

## close

```
public void close()  
    throws StorageConnectorException,  
           IOException
```

Closes the entity and releases all its resources.

**Note:** Any uncommitted changes are lost. Caller cannot use methods or properties which depend on an opened instance of the object.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while closing entity on archive.

IOException - if an I/O error occurs.

## com.imis.storageconnector.entitymodel Interface IEntityACLEntry

public interface **IEntityACLEntry**  
extends

Represents an Access Control List entry, a set of rights attached to a subject.

The `IEntityACLEntry` object represents ACL entry bound to the subject of the owner `IEntity`. It provides access to the inherited and explicit set of rights defined for the subject. It also enables subjects with `ChangeRights` right to modify the set of rights of the `IEntity`, its validity period and have the option of deleting the entry from the set of rights.

**See Also:**

[IACL](#), [IEntity](#)

### Method Summary

void	<a href="#">delete()</a> Deletes the ACL entry from the owner <code>IACL</code> .
<a href="#">IReadOnlyEntityRights</a>	<a href="#">getEffectiveRights()</a> Gets the users effective rights on the <code>IEntity</code> to which ACL entry is bound to.
<a href="#">IEntityRights</a>	<a href="#">getExplicitAllowRights()</a> Gets the Explicit Allow set of rights for the subject in question.
<a href="#">IEntityRights</a>	<a href="#">getExplicitDenyRights()</a> Gets the Explicit Deny set of rights for the subject in question.
<a href="#">IReadOnlyEntityRights</a>	<a href="#">getInheritedAllowRights()</a> Gets the Inherited Allow set of rights for the subject in question.
<a href="#">IReadOnlyEntityRights</a>	<a href="#">getInheritedDenyRights()</a> Gets the Inherited Deny set of rights for the subject in question.
<a href="#">IACL</a>	<a href="#">getOwner()</a> Gets the owner <code>IACL</code> .
String	<a href="#">getSubject()</a> Gets the subject to which ACL entry is bound to.
<a href="#">ACLEntrySubjectType</a>	<a href="#">getSubjectType()</a> Gets the ACL entry subject type.

### Methods

#### getOwner

public [IACL](#) **getOwner()**

Gets the owner `IACL`.

**Note:** This property never changes its value after `IEntityACLEntry` is instantiated.

**Returns:**



(continued from last page)

A owner IACL instance.

---

## getSubjectType

```
public ACLEntrySubjectType getSubjectType()
```

Gets the ACL entry subject type.

**Returns:**

The ACL entry subject type.

---

## getSubject

```
public String getSubject()
```

Gets the subject to which ACL entry is bound to.

**Returns:**

The ACL entry subject.

---

## getInheritedDenyRights

```
public IReadOnlyEntityRights getInheritedDenyRights()  
throws StorageConnectorException
```

Gets the Inherited Deny set of rights for the subject in question.

**Returns:**

The IReadOnlyEntityRights to the Inherited Deny set of rights.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

**See Also:**

[getSubject\(\)](#)

---

## getInheritedAllowRights

```
public IReadOnlyEntityRights getInheritedAllowRights()  
throws StorageConnectorException
```

Gets the Inherited Allow set of rights for the subject in question.

**Returns:**

The IReadOnlyEntityRights to the Inherited Allow set of rights.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

**See Also:**

[getSubject\(\)](#)

## getExplicitDenyRights

```
public IEntityRights getExplicitDenyRights()  
    throws StorageConnectorException
```

Gets the Explicit Deny set of rights for the subject in question.

Caller must have ChangeRights right to the owner entity to be allowed to modify set of rights.

### Returns:

The IEntityRights to the Explicit Deny set of rights.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while adding ExplicitDeny ACL entry to the entity.

### See Also:

[getSubject\(\)](#)

---

## getExplicitAllowRights

```
public IEntityRights getExplicitAllowRights()  
    throws StorageConnectorException
```

Gets the Explicit Allow set of rights for the subject in question.

Caller must have ChangeRights right to the owner entity to be allowed to modify set of rights.

### Returns:

The IEntityRights to the Explicit Allow set of rights.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while adding ExplicitAllow ACL entry to the entity.

### See Also:

[getSubject\(\)](#)

---

## getEffectiveRights

```
public IReadOnlyEntityRights getEffectiveRights()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the users effective rights on the IEntity to which ACL entry is bound to.

### Returns:

The users effective rights on the IEntity to which ACL entry is bound to or a null reference if none available.

### Throws:

[CancellationException](#) - if an operation was canceled by the user.

---

(continued from last page)

`LoginException` - if a login error occurs.

[`StorageConnectorException`](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.

`IOException` - if an I/O error occurs.

---

## delete

```
public void delete()  
    throws StorageConnectorException
```

Deletes the ACL entry from the owner `IACL`.

**Note:** Removal is not committed to the `Archive` unless the owner entity of the owner `IACL` object is committed to the archive using [`IEntity.save\(\)`](#) method. Caller must have `ChangeRights` right to the owner entity to be allowed to call this method.

**Throws:**

[`StorageConnectorException`](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while removing set of rights.

**See Also:**

[`IACL`](#)

[`getOwner\(\)`](#)

## com.imis.storageconnector.entitymodel Interface IEntityCollection

public interface **IEntityCollection**  
extends com.imis.util.ILargeReadOnlyList

Represents a read-only collection of entity stubs.

**Parameters:**

**E** - the type of the element in the entity collection.

### Method Summary

long	<a href="#"><u>getAccessibleCount()</u></a> Gets available entity collection element count.
com.imis.util.IReadOnlyMap	<a href="#"><u>getCategories()</u></a> Gets available entity collection elements per category.
String	<a href="#"><u>getCategory()</u></a> Gets the effective entity collection category.
com.imis.util.IReadOnlyList	<a href="#"><u>getTemplates()</u></a> Gets the entity collection element templates.
boolean	<a href="#"><u>isTruncated()</u></a> Gets a value indicating whether or not the collection was truncated to the specified number of elements.
void	<a href="#"><u>setCategory(String value)</u></a> Sets the effective entity collection category.

#### Methods inherited from interface com.imis.util.ILargeReadOnlyList

contains, containsAll, equals, get, hashCode, indexOf, isEmpty, iterator, lastIndexOf, listIterator, listIterator, size, subList, toArray, toArray

#### Methods inherited from interface com.imis.util.ILargeReadOnlyCollection

contains, containsAll, equals, hashCode, isEmpty, iterator, size, toArray, toArray

#### Methods inherited from interface java.lang.Iterable

iterator

### Methods

#### getTemplates

public com.imis.util.IReadOnlyList **getTemplates()**

Gets the entity collection element templates.

**Returns:**

(continued from last page)

The read-only list of entity collection element templates.

**Throws:**

`IllegalStateException` - if entity collection is closed.

---

## getCategory

```
public String getCategory()
```

Gets the effective entity collection category.

**Returns:**

The effective entity collection category or a null reference to use uncategorized entity collection.

---

## setCategory

```
public void setCategory(String value)  
    throws StorageConnectorException
```

Sets the effective entity collection category.

**Parameters:**

value - the effective entity collection category or a null reference to use uncategorized entity collection.

**Throws:**

`IllegalStateException` - if entity collection is closed.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Category is not valid category for this entity collection.

---

## getCategories

```
public com.imis.util.IReadOnlyMap getCategories()
```

Gets available entity collection elements per category.

**Returns:**

The read-only map of entity collection element count sorted by category.

**Throws:**

`IllegalStateException` - if entity collection is closed.

---

## getAccessibleCount

```
public long getAccessibleCount()  
    throws StorageConnectorException,  
           IOException
```

Gets available entity collection element count.

**Returns:**

The number of elements in collection available to the caller.

**Throws:**

`IllegalStateException` - if entity collection is closed.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs.

`IOException` - if an I/O error occurs.

---

## isTruncated

```
public boolean isTruncated()
```

Gets a value indicating whether or not the collection was truncated to the specified number of elements.

**Returns:**

true if collection was truncated; otherwise false.

**Throws:**

`IllegalStateException` - if entity collection is closed.

## com.imis.storageconnector.entitymodel Interface IEntityReference

public interface **IEntityReference**  
extends

Defines the entity reference.

### Method Summary

void	<a href="#"><u>close()</u></a> Closes the entity reference and releases all its resources.
Calendar	<a href="#"><u>created()</u></a> Gets the creation time-stamp of the entity reference.
<a href="#"><u>IACL</u></a>	<a href="#"><u>getACL()</u></a> Gets the entity reference IACL.
<a href="#"><u>IDirectoryEntity</u></a>	<a href="#"><u>getCreator()</u></a> Gets the creator IDirectoryEntity of the entity reference.
String	<a href="#"><u>getDescription()</u></a> Gets the description of the entity reference.
<a href="#"><u>IReadOnlyEntityRights</u></a>	<a href="#"><u>getEffectiveRights()</u></a> Gets the users effective rights on the entity reference.
String	<a href="#"><u>getId()</u></a> Gets the identifier of the entity reference.
<a href="#"><u>IDirectoryEntity</u></a>	<a href="#"><u>getModifiedBy()</u></a> Gets the last modifier directory entity.
com.imis.util.IReadOn lyList	<a href="#"><u>getReferenceIds()</u></a> Gets the referenced entities identifiers.
String	<a href="#"><u>getTitle()</u></a> Gets the title of the entity reference.
Calendar	<a href="#"><u>modified()</u></a> Gets the last modification time-stamp of the entity reference.
void	<a href="#"><u>save()</u></a> Saves the changes and reopens the entity reference in read-write mode.
void	<a href="#"><u>save(boolean close)</u></a> Saves the entity reference changes and optionally closes the entity reference or reopens the entity reference in read-write mode.
void	<a href="#"><u>setDescription(String value)</u></a> Sets the description of the entity reference.
void	<a href="#"><u>setReferenceIds(List value)</u></a> Sets the referenced entities identifiers.

void

[setTitle](#)(String value)

Sets the title of the entity reference.

## Methods

### getId

```
public String getId()  
    throws StorageConnectorException
```

Gets the identifier of the entity reference.

**Returns:**

A string that represents the identifier the entity reference.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity reference is not allowed.
- Operation on unsaved new entity reference is not allowed.
- Error occurred while getting entity reference identifier.

### getTitle

```
public String getTitle()  
    throws StorageConnectorException
```

Gets the title of the entity reference.

**Returns:**

The title of the entity reference.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity reference is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

### setTitle

```
public void setTitle(String value)  
    throws StorageConnectorException
```

Sets the title of the entity reference.

**Parameters:**

value - the title of the entity reference.

**Throws:**

[NullPointerException](#) - if value is a null reference.



(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity reference is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getDescription

```
public String getDescription()  
    throws StorageConnectorException
```

Gets the description of the entity reference.

### Returns:

The description of the entity reference.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity reference is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setDescription

```
public void setDescription(String value)  
    throws StorageConnectorException
```

Sets the description of the entity reference.

### Parameters:

value - the description of the entity reference.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity reference is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## created

```
public Calendar created()  
    throws StorageConnectorException
```

Gets the creation time-stamp of the entity reference.

### Returns:

The date and time when the entity reference was created.

### Throws:

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity reference is not allowed.
- 

## getCreator

```
public IDirectoryEntity getCreator()  
    throws StorageConnectorException
```

Gets the creator [IDirectoryEntity](#) of the entity reference.

**Returns:**

The creator [IDirectoryEntity](#) of the entity reference.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity reference is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## modified

```
public Calendar modified()  
    throws StorageConnectorException
```

Gets the last modification time-stamp of the entity reference.

**Returns:**

The date and time when the describing entity was last modified.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- 

## getModifiedBy

```
public IDirectoryEntity getModifiedBy()  
    throws StorageConnectorException
```

Gets the last modifier directory entity.

**Returns:**

The last modifier directory entity or a null reference, if this instance has no last modifier directory entity.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
-

(continued from last page)

## getEffectiveRights

```
public IReadOnlyEntityRights getEffectiveRights()  
    throws StorageConnectorException
```

Gets the users effective rights on the entity reference.

**Note:** The following rights are not applicable on entity references: MoveAccess, CreateSubEntities, ChangeSecurityClass, ChangeStatus, ChangeRetention and CreateReferences.

**Returns:**

The read-only set of rights describing users effective rights on the entity reference.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity reference is not allowed.

---

## getACL

```
public IACL getACL()  
    throws StorageConnectorException
```

Gets the entity reference IACL.

The IACL property provides access to an IACL instance containing ACL entries which define access rights for different entity reference operations.

**Note:** The following rights are not applicable on entity references: MoveAccess, CreateSubEntities, ChangeSecurityClass, ChangeStatus, ChangeRetention and CreateReferences.

**Returns:**

A IACL instance.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity reference is not allowed.
- Error occurred while getting ACL entries.

---

## getReferenceIds

```
public com.imis.util.IReadOnlyList getReferenceIds()  
    throws StorageConnectorException
```

Gets the referenced entities identifiers.

**Returns:**

The read-only list of referenced entities identifiers.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
-

(continued from last page)

## setReferenceIds

```
public void setReferenceIds(List value)  
    throws StorageConnectorException
```

Sets the referenced entities identifiers.

### Parameters:

value - the list of referenced entities identifiers.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity reference is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## save

```
public void save()  
    throws StorageConnectorException,  
        javax.security.auth.login.LoginException,  
        IOException
```

Saves the changes and reopens the entity reference in read-write mode.

**Note:** This operation allows user to pass a reason for saving entity reference. Reason can be passed through [AuditLog](#) message.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity reference is not allowed.
- Error occurred while committing entity reference changes on archive.

IOException - if an I/O error occurs.

---

## save

```
public void save(boolean close)  
    throws javax.security.auth.login.LoginException,  
        StorageConnectorException,  
        IOException
```

Saves the entity reference changes and optionally closes the entity reference or reopens the entity reference in read-write mode.

**Note:** This operation allows user to pass a reason for saving entity reference. Reason can be passed through [AuditLog](#) message.

### Parameters:

close - true to close the entity reference after save; otherwise false.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity reference is not allowed.
- Error occurred while committing entity reference changes on archive.

`IOException` - if an I/O error occurs.

---

## **close**

```
public void close()  
    throws StorageConnectorException,  
           IOException
```

Closes the entity reference and releases all its resources.

### **Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while closing entity reference on archive.

`IOException` - if an I/O error occurs.

## com.imis.storageconnector.entitymodel Interface IEntityRights

All Superinterfaces:

[IReadOnlyEntityRights](#)

public interface **IEntityRights**  
extends [IReadOnlyEntityRights](#)

Allows the caller to access the set of rights for a certain subject in a read-write fashion.

The following rights are defined:

Right	Description
ReadAccess	Subject has read access to entity metadata attributes and other properties.
WriteAccess	Subject has write access to entity metadata attributes and other properties.
DeleteAccess	Subject has the right to delete the entity from the archive.
ChangeRights	Subject has the right to modify the entity explicit security descriptor.
CreateSubEntities	Subject has the right to create new IEntity sub-entities.
ChangeSecurityClass	Subject has the right to set the entity security class.
ChangeStatus	Subject has the right to set the entity status.
ChangeRetention	Subject has the right to change the entity retention.
CreateReferences	Subject has the right to create entity references on the entity.

See Also:

[IEntity](#), [IEntityACLEntry](#), [IReadOnlyEntityRights](#)

### Method Summary

void	<a href="#">delete()</a> Removes this set of rights from the owning IEntityACLEntry.
void	<a href="#">setChangeRetention(boolean value)</a> Sets the value indicating whether the subject has the right to change the current entity retention.
void	<a href="#">setChangeRights(boolean value)</a> Sets the value indicating whether the subject has the right to modify explicit ACL entries bound to the owning entity.
void	<a href="#">setChangeSecurityClass(boolean value)</a> Sets the value indicating whether the subject has the right to change the current entity security class.
void	<a href="#">setChangeStatus(boolean value)</a> Sets the value indicating whether the subject has the right to change the current entity status.
void	<a href="#">setCreateReferences(boolean value)</a> Sets the value indicating whether the subject has the right to create entity references.

void	<a href="#"><u>setCreateSubEntities</u></a> (boolean value) Sets the value indicating whether the subject has the right to create sub-entities relative to the current entity.
void	<a href="#"><u>setDeleteAccess</u></a> (boolean value) Sets the value indicating whether the subject has the right to delete the owning entity from the archive.
void	<a href="#"><u>setMoveAccess</u></a> (boolean value) Sets the value indicating whether the subject has the right to move the current entity.
void	<a href="#"><u>setPropagated</u></a> (boolean propagated) Sets the value indicating whether or not the entry is propagated.
void	<a href="#"><u>setReadAccess</u></a> (boolean value) Sets the value indicating whether the subject has the right to access (read) entity metadata attributes and other properties.
void	<a href="#"><u>setValidFrom</u></a> (Calendar value) Sets the Calendar after which this entry is valid when calculating the set of rights for the owning entity.
void	<a href="#"><u>setValidTo</u></a> (Calendar value) Sets the Calendar before which this entry is valid when calculating the set of rights for the owning entity.
void	<a href="#"><u>setWriteAccess</u></a> (boolean value) Sets the value indicating whether the subject has the right to add or modify entity metadata attributes and other properties.

Methods inherited from interface [com.imis.storageconnector.entitymodel.IReadOnlyEntityRights](#)

[changeRetention](#), [changeRights](#), [changeSecurityClass](#), [changeStatus](#), [createReferences](#), [createSubEntities](#), [deleteAccess](#), [getOwner](#), [isPropagated](#), [moveAccess](#), [readAccess](#), [validFrom](#), [validTo](#), [writeAccess](#)

## Methods

### setReadAccess

public void **setReadAccess**(boolean value)  
throws [StorageConnectorException](#)

Sets the value indicating whether the subject has the right to access (read) entity metadata attributes and other properties.

#### Parameters:

value - true to grant ReadAccess right; otherwise false.

#### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of ReadAccess right.

**See Also:**

(continued from last page)

[IEntityACLEntry.getSubject\(\)](#)

---

## setWriteAccess

public void **setWriteAccess**(boolean value)  
throws [StorageConnectorException](#)

Sets the value indicating whether the subject has the right to add or modify entity metadata attributes and other properties.

### Parameters:

value - true to grant WriteAccess right; otherwise false.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of WriteAccess right.

### See Also:

[IEntityACLEntry.getSubject\(\)](#)

---

## setMoveAccess

public void **setMoveAccess**(boolean value)  
throws [StorageConnectorException](#)

Sets the value indicating whether the subject has the right to move the current entity.

**Note:** This right is not applicable on entity references.

### Parameters:

value - true if the MoveAccess right is granted; otherwise false.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of MoveAccess right.

### See Also:

[IEntityACLEntry.getSubject\(\)](#)

---

## setDeleteAccess

public void **setDeleteAccess**(boolean value)  
throws [StorageConnectorException](#)

Sets the value indicating whether the subject has the right to delete the owning entity from the archive.

### Parameters:

value - true to grant DeleteAccess right; otherwise false.

### Throws:



(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of `DeleteAccess` right.

See Also:

[IEntityACLEntry.getSubject\(\)](#)

---

## setChangeRights

```
public void setChangeRights(boolean value)
    throws StorageConnectorException
```

Sets the value indicating whether the subject has the right to modify explicit ACL entries bound to the owning entity.

**Parameters:**

value - true to grant `ChangeRights` right; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of `ChangeRights` right.

See Also:

[IEntityACLEntry.getSubject\(\)](#)

---

## setCreateSubEntities

```
public void setCreateSubEntities(boolean value)
    throws StorageConnectorException
```

Sets the value indicating whether the subject has the right to create sub-entities relative to the current entity.

**Note:** This right is not applicable on entity references.

**Parameters:**

value - true to grant `CreateSubEntities` right; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of `CreateSubEntities` right.

See Also:

[IEntityACLEntry.getSubject\(\)](#)

---

(continued from last page)

## setChangeSecurityClass

```
public void setChangeSecurityClass(boolean value)
    throws StorageConnectorException
```

Sets the value indicating whether the subject has the right to change the current entity security class.

**Note:** This right is not applicable on entity references.

### Parameters:

value - true if the ChangeSecurityClass right is granted; otherwise false.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of ChangeSecurityClass right.

### See Also:

[IEntityACLEntry.getSubject\(\)](#)

---

## setChangeStatus

```
public void setChangeStatus(boolean value)
    throws StorageConnectorException
```

Sets the value indicating whether the subject has the right to change the current entity status.

**Note:** This right is not applicable on entity references.

### Parameters:

value - true if the ChangeStatus right is granted; otherwise false.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of ChangeStatus right.

### See Also:

[IEntityACLEntry.getSubject\(\)](#)

---

## setChangeRetention

```
public void setChangeRetention(boolean value)
    throws StorageConnectorException
```

Sets the value indicating whether the subject has the right to change the current entity retention.

**Note:** This right is not applicable on entity references.

### Parameters:

value - true if the ChangeRetention right is granted; otherwise false.

### Throws:

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of `ChangeRetention` right.

**See Also:**

[IEntityACLEntry.getSubject\(\)](#)

---

## setCreateReferences

```
public void setCreateReferences(boolean value)
    throws StorageConnectorException
```

Sets the value indicating whether the subject has the right to create entity references.

**Note:** This right is not applicable on entity references.

**Parameters:**

value - true if the `CreateReferences` right is granted; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of `CreateReferences` right.

**See Also:**

[IEntityACLEntry.getSubject\(\)](#)

---

## setValidFrom

```
public void setValidFrom(Calendar value)
    throws StorageConnectorException
```

Sets the `Calendar` after which this entry is valid when calculating the set of rights for the owning entity.

**Parameters:**

value - a `Calendar` value that represents the entry validity start timestamp.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting `Calendar` value for set of rights.

---

## setValidTo

```
public void setValidTo(Calendar value)
    throws StorageConnectorException
```

Sets the `Calendar` before which this entry is valid when calculating the set of rights for the owning entity.

(continued from last page)

**Parameters:**

value - a Calendar value that represents the entry validity end timestamp.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting Calendar value for set of rights.

---

## setPropagated

```
public void setPropagated(boolean propagated)  
    throws StorageConnectorException
```

Sets the value indicating whether or not the entry is propagated.

**Parameters:**

propagated - the propagated value.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.

---

## delete

```
public void delete()  
    throws StorageConnectorException
```

Removes this set of rights from the owning IEntityACLEntry.

Removal is not committed to the archive unless owning IEntity is committed to the Archive instance using [IEntity.save\(\)](#) method.

Caller must have ChangeRights right to the owner entity to be allowed to call this method.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while removing set of rights.

**See Also:**

[IEntityACLEntry](#)

## com.imis.storageconnector.entitymodel Interface IEntityStub

All Subinterfaces:

[ISearchedEntityStub](#)

public interface **IEntityStub**  
extends

Defines an IEntity descriptor.

An IEntityStub defines an IEntity descriptor containing the entity identifier and type, public metadata attributes, its position in the entity tree (parent and children), callers effective rights on the entity object, etc. It also enables a caller to open a full-feature IEntity object using the open() method. Accessing IEntityStub properties and methods other than open() will not be recorded in the archive's AuditLog records.

**Note:** Instance is automatically recycled after the last reference is released.

**See Also:**

[IEntity](#)

### Method Summary

Calendar	<a href="#">accessed()</a> Gets the describing IEntity last access timestamp.
boolean	<a href="#">areValuesInherited(SystemProperty systemProperty)</a> Gets the value indicating whether the specified system property values are inherited.
String	<a href="#">checkOut()</a> Checks out the entity.
Calendar	<a href="#">closed()</a> Gets the describing IEntity status closed time-stamp.
Calendar	<a href="#">created()</a> Gets the describing IEntity creation timestamp.
void	<a href="#">delete()</a> Deletes the describing IEntity and associated resources.
void	<a href="#">delete(String description)</a> Deletes the describing IEntity and associated resources.
<a href="#">IArchive</a>	<a href="#">getArchive()</a> Gets the IArchive where the describing IEntity is stored.
com.imis.util.IReadOnlyList	<a href="#">getCategories()</a> Gets the list of the describing IEntity categories.
String	<a href="#">getClassificationCode()</a> Gets the describing IEntity classification code in canonical format.
<a href="#">IReadOnlyContent</a>	<a href="#">getContent()</a> Gets the describing IEntity read-only content.

<a href="#">IDirectoryEntity</a>	<a href="#">getCreator()</a> Gets the describing IEntity creator IDirectoryEntity.
com.imis.util.IReadOnlyMap	<a href="#">getCustomContents()</a> Gets the describing IEntity read-only custom contents.
String	<a href="#">getDescription()</a> Gets the describing IEntity description.
<a href="#">IDirectoryEntity</a>	<a href="#">getDraftCreator()</a> Gets the entity's draft creator.
String	<a href="#">getDraftId()</a> Gets the entity's draft identifier.
<a href="#">IDraftStub</a>	<a href="#">getDraftInfo()</a> Queries archiving server for draft public properties (metadata attributes) and other specifics which can be accessed without opening a draft.
<a href="#">IReadOnlyEntityRights</a>	<a href="#">getEffectiveRights()</a> Gets the users effective rights on the describing IEntity.
com.imis.util.IReadOnlyList	<a href="#">getExternalIds()</a> Gets the list of the describing IEntity unique external identifiers.
String	<a href="#">getId()</a> Gets the describing IEntity unique identifier.
com.imis.util.IReadOnlyList	<a href="#">getKeywords()</a> Gets the describing IEntity list of keywords.
<a href="#">IDirectoryEntity</a>	<a href="#">getModifiedBy()</a> Gets the last modifier directory entity.
<a href="#">IDirectoryEntity</a>	<a href="#">getOwner()</a> Gets the describing IEntity owner IDirectoryEntity.
<a href="#">IEntityStub</a>	<a href="#">getParent()</a> Gets the describing IEntity parent IEntityStub.
com.imis.util.IReadOnlyList	<a href="#">getPickListValues(SystemProperty systemProperty)</a> Gets the read-only list of the pick-list values for the specified system property.
com.imis.util.IReadOnlyMap	<a href="#">getProperties()</a> Gets the entity public metadata attributes.
String	<a href="#">getPublicClassificationCode()</a> Gets the describing IEntity classification code in public format.
com.imis.util.IReadOnlyList	<a href="#">getReferences()</a> Gets the read-only list of IReadOnlyEntityReference objects.
InputStream	<a href="#">getReport(ReportType type)</a> Gets the specified report data stream for the describing IEntity.
<a href="#">IRetention</a>	<a href="#">getRetention(AccessMode mode)</a> Gets the retention for the entity.
String	<a href="#">getSecurityClass()</a> Gets the IEntity security class.

String	<a href="#"><u>getSignificance()</u></a> Gets the describing IEntity significance.
String	<a href="#"><u>getStatus()</u></a> Gets the describing IEntity status.
<a href="#"><u>IEntityCollection</u></a>	<a href="#"><u>getSubEntities()</u></a> ( <a href="#"><u>EntityFilter</u></a> filter, Collection sortKeys, Collection categorizationPropertyNames) Gets the read-only collection of child entity stubs of the describing IEntity using the specified entity filter, sort keys and a collection of property names used for categorization.
com.imis.util.IReadOnlyList	<a href="#"><u>getSubEntityTemplates()</u></a> Gets the read-only list of templates that represent the describing IEntity sub-entity creation templates.
com.imis.util.IReadOnlyMap	<a href="#"><u>getSystemProperties()</u></a> Gets the read-only map of read-only system properties sorted by SystemProperty enum values.
<a href="#"><u>IReadOnlyProperty</u></a>	<a href="#"><u>getSystemProperty()</u></a> ( <a href="#"><u>SystemProperty</u></a> systemProperty) Gets the read-only property for the specified system property.
<a href="#"><u>ITemplate</u></a>	<a href="#"><u>getTemplate()</u></a> Gets the describing IEntity template.
String	<a href="#"><u>getTitle()</u></a> Gets the describing IEntity title.
<a href="#"><u>EntityType</u></a>	<a href="#"><u>getType()</u></a> Gets the describing IEntity type identifier.
String	<a href="#"><u>getVersionComment()</u></a> Gets the entity check in comment.
String	<a href="#"><u>getVersionLabel()</u></a> Gets the label of the entity version.
com.imis.util.IReadOnlyList	<a href="#"><u>getVersions()</u></a> Gets the read-only list of entity versions.
Integer	<a href="#"><u>getVersionSequence()</u></a> Gets the entity version sequence.
boolean	<a href="#"><u>hasAIP()</u></a> Gets a value indicating whether or not the describing IEntity has AIP.
boolean	<a href="#"><u>hasDraft()</u></a> Gets the value indicating whether or not the entity has a draft.
boolean	<a href="#"><u>hasReferences()</u></a> Gets the value indicating whether or not the entity has references.
boolean	<a href="#"><u>hasVersions()</u></a> Gets the value indicating whether or not the entity has versions.
boolean	<a href="#"><u>isChildClassificationCodeGenerated()</u></a> Gets a value indicating whether or not classification codes of describing IEntity children are auto generated.

boolean	<a href="#"><code>isPermanent()</code></a> Gets a value indicating whether or not the describing IEntity is permanent.
Calendar	<a href="#"><code>modified()</code></a> Gets the describing IEntity last modification timestamp.
void	<a href="#"><code>move(IEntityStub parent)</code></a> Moves the describing IEntity under the specified parent entity.
<a href="#"><code>IEntity</code></a>	<a href="#"><code>open(AccessMode mode)</code></a> Opens an IEntity described by this IEntityStub using a specified entity access mode.
Calendar	<a href="#"><code>opened()</code></a> Gets the describing IEntity status opened time-stamp.
<a href="#"><code>IEntityCollection</code></a>	<a href="#"><code>search(EntityFilter filter, Collection sortKeys, Set options, String searchExpression, Collection categorizationPropertyNames, Long maxElements)</code></a> Searches for child entity stubs of the describing IEntity using the specified entity search filter, sort keys, search options, expression string and a collection of property names used for categorization.
void	<a href="#"><code>setSecurityClass(String securityClass)</code></a> Sets the security class of the describing IEntity.
void	<a href="#"><code>setStatus(String status)</code></a> Sets the status of the describing IEntity.
long	<a href="#"><code>subEntityCount()</code></a> Gets the absolute sub-entity count.

## Methods

### getId

public String **getId()**

Gets the describing IEntity unique identifier.

**Returns:**

A string that represents the describing IEntity identifier.

### getClassificationCode

public String **getClassificationCode()**

Gets the describing IEntity classification code in canonical format.

**Returns:**

A string that represents the describing IEntity classification code.

### getPublicClassificationCode

public String **getPublicClassificationCode()**



(continued from last page)

Gets the describing IEntity classification code in public format.

The entity classification code in canonical format is converted to entity public classification code by replacing the entity type prefix and classification hierarchy delimiter with the respective entity type hierarchy delimiters.

**Returns:**

A string that represents the describing IEntity public classification code.

---

## isChildClassificationCodeGenerated

```
public boolean isChildClassificationCodeGenerated()
```

Gets a value indicating whether or not classification codes of describing IEntity children are auto generated.

**Note:** Operation supported for entities of type CLASS and FOLDER only.

**Returns:**

true if children classification codes are auto generated; otherwise false.

**Throws:**

UnsupportedOperationException - if unsupported entity type.

---

## getTitle

```
public String getTitle()  
throws StorageConnectorException
```

Gets the describing IEntity title.

**Returns:**

The describing IEntity title.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getDescription

```
public String getDescription()  
throws StorageConnectorException
```

Gets the describing IEntity description.

**Returns:**

The describing IEntity description.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getSignificance

```
public String getSignificance()  
throws StorageConnectorException
```

(continued from last page)

Gets the describing IEntity significance.

**Returns:**

The describing IEntity significance.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getStatus

```
public String getStatus()  
    throws StorageConnectorException
```

Gets the describing IEntity status.

**Returns:**

The describing IEntity status.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getType

```
public EntityType getType()
```

Gets the describing IEntity type identifier.

**Returns:**

An enumerator value that represents the describing IEntity type.

---

## created

```
public Calendar created()
```

Gets the describing IEntity creation timestamp.

**Returns:**

The date and time when the describing entity was created.

---

## modified

```
public Calendar modified()
```

Gets the describing IEntity last modification timestamp.

**Returns:**

The date and time when the describing entity was last modified.

---

## getModifiedBy

```
public IDirectoryEntity getModifiedBy()
```

(continued from last page)

Gets the last modifier directory entity.

**Returns:**

The last modifier directory entity or a null reference, if this instance has no last modifier directory entity.

---

**accessed**

```
public Calendar accessed()
```

Gets the describing IEntity last access timestamp.

**Returns:**

The date and time when the describing entity was last accessed.

---

**opened**

```
public Calendar opened()  
throws StorageConnectorException
```

Gets the describing IEntity status opened time-stamp.

**Returns:**

The date and time when the describing IEntity status was set to opened or a null reference if entity has no value.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

**closed**

```
public Calendar closed()  
throws StorageConnectorException
```

Gets the describing IEntity status closed time-stamp.

**Returns:**

The date and time when the describing IEntity status was set to closed or a null reference if entity has no value.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

**isPermanent**

```
public boolean isPermanent()
```

Gets a value indicating whether or not the describing IEntity is permanent.

**Returns:**

true if the describing IEntity is permanent; otherwise false.

(continued from last page)

## hasAIP

```
public boolean hasAIP()
```

Gets a value indicating whether or not the describing IEntity has AIP.

**Returns:**

true if the describing IEntity has AIP; otherwise false.

---

## hasDraft

```
public boolean hasDraft()
```

Gets the value indicating whether or not the entity has a draft.

**Returns:**

true if the entity has a draft; otherwise false.

---

## getDraftId

```
public String getDraftId()
```

Gets the entity's draft identifier.

**Returns:**

The entity's draft identifier.

---

## getDraftInfo

```
public IDraftStub getDraftInfo()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Queries archiving server for draft public properties (metadata attributes) and other specifics which can be accessed without opening a draft. The returned IDraftStub object describes the public information about the draft which was queried.

**Returns:**

An IDraftStub object.

**Throws:**

[CancellationException](#) - if an operation was canceled by the user.  
[LoginException](#) - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Draft identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.

[IOException](#) - if an I/O error occurs.

---

## getDraftCreator

```
public IDirectoryEntity getDraftCreator()  
    throws StorageConnectorException
```

Gets the entity's draft creator.

---

(continued from last page)

**Returns:**

The entity's draft creator.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while getting directory member on archive.

---

## hasVersions

```
public boolean hasVersions()
```

Gets the value indicating whether or not the entity has versions.

**Returns:**

true if the entity has versions; otherwise false.

---

## getVersionSequence

```
public Integer getVersionSequence()
```

Gets the entity version sequence.

**Returns:**

The entity version sequence.

---

## getVersionLabel

```
public String getVersionLabel()  
throws StorageConnectorException
```

Gets the label of the entity version.

**Returns:**

A string that represents the label of the entity version.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getVersionComment

```
public String getVersionComment()  
throws StorageConnectorException
```

Gets the entity check in comment.

**Returns:**

The entity check in comment.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getVersions

```
public com.imis.util.IReadOnlyList getVersions()
```

Gets the read-only list of entity versions.

**Returns:**

The read-only list of `IEntityStub` objects that represent the entity versions.

---

## subEntityCount

```
public long subEntityCount()
```

Gets the absolute sub-entity count.

**Returns:**

The number of all immediate sub-entities.

---

## getEffectiveRights

```
public IReadOnlyEntityRights getEffectiveRights()
```

Gets the users effective rights on the describing `IEntity`.

**Returns:**

The read-only set of rights describing users effective rights on the describing `IEntity`.

---

## getSecurityClass

```
public String getSecurityClass()  
throws StorageConnectorException
```

Gets the `IEntity` security class.

**Returns:**

The string value from SecurityClass system attribute enumeration representing the `IEntity` security class or null value if not defined.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getArchive

```
public IArchive getArchive()
```

Gets the `IArchive` where the describing `IEntity` is stored.

**Returns:**

The `IArchive` where the describing `IEntity` is stored.

## getParent

```
public IEntityStub getParent()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the describing IEntity parent IEntityStub.

**Returns:**

The parent IEntityStub or a null reference, if this instance is a root IEntityStub.

**Throws:**

[CancellationException](#) - if an operation was canceled by the user.

[LoginException](#) - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while getting entity stub parent.

[IOException](#) - if an I/O error occurs.

---

## getCreator

```
public IDirectoryEntity getCreator()  
    throws StorageConnectorException
```

Gets the describing IEntity creator IDirectoryEntity.

**Returns:**

The creator IDirectoryEntity.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- 

## getOwner

```
public IDirectoryEntity getOwner()  
    throws StorageConnectorException
```

Gets the describing IEntity owner IDirectoryEntity.

**Returns:**

The creator IDirectoryEntity or a null reference, if this instance has no owner IDirectoryEntity.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- 

## getTemplate

```
public ITemplate getTemplate()
```

---

(continued from last page)

Gets the describing IEntity template.

**Returns:**

A string value that represents the entity template.

---

## getKeywords

```
public com.imis.util.IReadOnlyList getKeywords()  
    throws StorageConnectorException
```

Gets the describing IEntity list of keywords.

**Returns:**

The describing IEntity list of keywords.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getExternalIds

```
public com.imis.util.IReadOnlyList getExternalIds()  
    throws StorageConnectorException
```

Gets the list of the describing IEntity unique external identifiers.

**Returns:**

A list of String objects that represents the describing IEntity external identifiers.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getCategories

```
public com.imis.util.IReadOnlyList getCategories()  
    throws StorageConnectorException
```

Gets the list of the describing IEntity categories.

**Returns:**

A list of String objects that represents the describing IEntity categories.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getSystemProperties

```
public com.imis.util.IReadOnlyMap getSystemProperties()
```



(continued from last page)

Gets the read-only map of read-only system properties sorted by `SystemProperty` enum values.

**Returns:**

The read-only map of `IReadOnlyProperty` objects sorted by `SystemProperty` enum values.

---

## getProperties

```
public com.imis.util.IReadOnlyMap getProperties()
```

Gets the entity public metadata attributes.

**Returns:**

The read-only map of `IReadOnlyProperty` objects sorted by the property names.

**See Also:**

[IReadOnlyProperty](#)

---

## getContent

```
public IReadOnlyContent getContent()  
throws StorageConnectorException
```

Gets the describing `IEntity` read-only content.

**Returns:**

The `IReadOnlyContent` object.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getCustomContents

```
public com.imis.util.IReadOnlyMap getCustomContents()
```

Gets the describing `IEntity` read-only custom contents.

**Returns:**

A read-only map of `IReadOnlyContent` objects sorted by the content names.

**See Also:**

[IDocument](#)

[IReadOnlyContent](#)

---

## getSubEntityTemplates

```
public com.imis.util.IReadOnlyList getSubEntityTemplates()
```

Gets the read-only list of templates that represent the describing `IEntity` sub-entity creation templates.

**Returns:**

The read-only list of `ITemplate` objects that represent the entity sub-entity creation templates.

(continued from last page)

## hasReferences

```
public boolean hasReferences()
```

Gets the value indicating whether or not the entity has references.

**Returns:**

true if the entity has references; otherwise false.

---

## getReferences

```
public com.imis.util.IReadOnlyList getReferences()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the read-only list of IReadOnlyEntityReference objects.

**Returns:**

The read-only list of IReadOnlyEntityReference objects.

**Throws:**

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.

IOException - if an I/O error occurs.

---

## getRetention

```
public IRetention getRetention(AccessMode mode)  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the retention for the entity.

**Note:** This operation allows user to pass a reason for opening retention. Reason can be passed through [AuditLog](#) message.

**Parameters:**

mode - the retention access mode.

**Returns:**

An IRetention instance.

**Throws:**

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening entity retention.

IOException - if an I/O error occurs.

## setStatus

```
public void setStatus(String status)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Sets the status of the describing IEntity.

**Note:** This operation allows user to pass a reason for changing entity status. Reason can be passed through [AuditLog](#) message.

### Parameters:

status - the describing IEntity status or a null reference to reset status to inherited value.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Property is not a pick-list property.
- Property value is not allowed based on the current template settings.
- Error occurred while setting status to value.

IOException - if an I/O error occurs.

---

## setSecurityClass

```
public void setSecurityClass(String securityClass)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Sets the security class of the describing IEntity.

**Note:** This operation requires user to pass a reason for changing entity security class. Reason must be passed through [AuditLog](#) message.

### Parameters:

securityClass - the describing IEntity security class or a null reference to reset security class to inherited value.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Property is not a pick-list property.
- Property value is not allowed based on the current template settings.
- Error occurred while setting security class to value.

IOException - if an I/O error occurs.

---

(continued from last page)

## getSubEntities

```
public ICollection getSubEntities(EntityFilter filter,  
    Collection sortKeys,  
    Collection categorizationPropertyNames)  
throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Gets the read-only collection of child entity stubs of the describing [IEntity](#) using the specified entity filter, sort keys and a collection of property names used for categorization.

### Parameters:

filter - the entity filter or a null reference.

sortKeys - the collection of [EntitySortKey](#) objects.

categorizationPropertyNames - the collection of property names used for categorization.

### Returns:

The read-only collection of [IEntityStub](#) objects representing the entity children.

### Throws:

[CancellationException](#) - if an operation was canceled by the user.

[LoginException](#) - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while getting entity child entities on archive.

[IOException](#) - if an I/O error occurs.

### See Also:

[EntityFilter](#)

[EntitySortKey](#)

---

## getSystemProperty

```
public IReadOnlyProperty getSystemProperty(SystemProperty systemProperty)  
throws StorageConnectorException
```

Gets the read-only property for the specified system property.

### Parameters:

systemProperty - the system property.

### Returns:

The read-only property for the specified system property.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## areValuesInherited

```
public boolean areValuesInherited(SystemProperty systemProperty)  
throws StorageConnectorException
```

Gets the value indicating whether the specified system property values are inherited.

(continued from last page)

**Parameters:**

systemProperty - the system property.

**Returns:**

true if the specified system property values are inherited; otherwise false.

**Throws:**

IllegalArgumentException - if invalid system property.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getPickListValues

```
public com.imis.util.IReadOnlyList getPickListValues(SystemProperty systemProperty)
    throws StorageConnectorException
```

Gets the read-only list of the pick-list values for the specified system property.

**Parameters:**

systemProperty - the system property.

**Returns:**

The read-only list of IPickListValue objects representing the pick-list values for the specified system property.

**Throws:**

IllegalArgumentException - if invalid system property.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.

---

## search

```
public IEntityCollection search(EntityFilter filter,
    Collection sortKeys,
    Set options,
    String searchExpression,
    Collection categorizationPropertyNames,
    Long maxElements)
    throws javax.security.auth.login.LoginException,
    StorageConnectorException,
    IOException
```

Searches for child entity stubs of the describing IEntity using the specified entity search filter, sort keys, search options, expression string and a collection of property names used for categorization.

**Parameters:**

filter - the entity search filter or a null reference.  
 sortKeys - the collection of EntitySortKey objects.  
 options - the search options.  
 searchExpression - the search expression string.  
 categorizationPropertyNames - the collection of property names used for categorization.  
 maxElements - the number of elements the resulting collection should be truncated to.

(continued from last page)

**Returns:**

The read-only collection of `ISearchedEntityStub` objects representing the search results.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while performing search on entity.

`IOException` - if an I/O error occurs.

---

## getReport

```
public InputStream getReport(ReportType type)
```

Gets the specified report data stream for the describing `IEntity`.

**Note:** Valid report types on `IEntityStub` are:

- [ReportType.ENTITY\\_MOVE](#)
- [ReportType.ENTITY\\_STATUS](#)
- [ReportType.ENTITY\\_SECURITY\\_CLASS](#)

**Parameters:**

type - the report type.

**Returns:**

The report data input stream or null if report not available.

**Throws:**

`UnsupportedOperationException` - if report not supported.

---

## open

```
public IEntity open(AccessMode mode)  
throws javax.security.auth.login.LoginException,  
       StorageConnectorException,  
       IOException
```

Opens an `IEntity` described by this `IEntityStub` using a specified entity access mode.

**Note:** This operation allows user to pass a reason for opening entity. Reason can be passed through [AuditLog](#) message.

**Parameters:**

mode - the entity access mode.

**Returns:**

An opened `IEntity` instance.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening entity on archive.

IOException - if an I/O error occurs.

---

## move

```
public void move(IEntityStub parent)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Moves the describing IEntity under the specified parent entity.

**Note:** This operation allows user to pass a reason for moving entity. Reason can be passed through [AuditLog](#) message.

### Parameters:

parent - the new parent entity.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while moving entity on archive.

IOException - if an I/O error occurs.

---

## delete

```
public void delete()
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Deletes the describing IEntity and associated resources.

Use this method to remove all resources associated with the entity and the entity itself. Delete right is required for this operation to succeed.

This operation requires user to pass reason for deleting entity Reason must be passed through [AuditLog](#) message.

This operation requires description for deleted entity. This method will fail if the entity does not have its own description.

This operation is irreversible. Entity will not be accessible after a successful call is completed.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while removing entity on archive.

IOException - if an I/O error occurs.

---

## delete

```
public void delete(String description)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Deletes the describing IEntity and associated resources.

Use this method to remove all resources associated with the entity and the entity itself. Delete right is required for this operation to succeed.

This operation requires user to pass reason for deleting entity Reason must be passed through [AuditLog](#) message.

This operation requires description for deleted entity. If one is not provided via `description` parameter, this method will fail if the entity does not have its own description.

This operation is irreversible. Entity will not be accessible after a successful call is completed.

### Parameters:

`description` - a custom description for deleted entity in case delete agent wants to specify it or a null reference to use existing entity description.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while removing entity on archive.

IOException - if an I/O error occurs.

---

## checkOut

```
public String checkOut()
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Checks out the entity.

This operation allows user to pass reason for checking out entity. Reason can be passed through [AuditLog](#) message.

### Returns:

The entity draft identifier.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.



(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on non document entity is not allowed.
- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while checking out entity on archive.

`IOException` - if an I/O error occurs.

## com.imis.storageconnector.entitymodel Interface IFolder

All Superinterfaces:

[IEntity](#)

public interface **IFolder**

extends [IEntity](#)

Defines a IFolder entity.

A IFolder entity represents a file folder which may contain metadata attributes, has its inherited and explicit ACL entries and can contain additional sub-folders and/or documents as sub-entities.

Current implementation inherits from the base IEntity and adds factory methods for creating new instances of sub-entities which can be either IFolder or IDocument instances.

See Also:

[IEntity](#), [IDocument](#)

### Method Summary

<a href="#">IDocument</a>	<a href="#">createDocument</a> (String templateId) Creates a new IDocument entity instance.
<a href="#">IFolder</a>	<a href="#">createFolder</a> (String templateId) Creates a new IFolder or sub-folder entity instance.
boolean	<a href="#">isChildClassificationCodeGenerated</a> () Gets a value indicating whether or not child classification code is auto generated.
void	<a href="#">setChildClassificationCodeGenerated</a> (boolean value) Sets a value indicating whether or not child classification code is auto generated.

### Methods inherited from interface [com.imis.storageconnector.entitymodel.IEntity](#)

[accessed](#), [areValuesInherited](#), [close](#), [closed](#), [createBinaryValue](#), [created](#), [createStringMaxValue](#), [getACL](#), [getArchivalInformationalPackage](#), [getArchive](#), [getAuditLog](#), [getCategories](#), [getClassificationCode](#), [getCreator](#), [getDescription](#), [getDraftCreator](#), [getDraftId](#), [getEffectiveRights](#), [getEmailEntity](#), [getEvidenceRecords](#), [getExternalIds](#), [getId](#), [getKeywords](#), [getMode](#), [getModifiedBy](#), [getMoveDetails](#), [getOwner](#), [getParent](#), [getPhysicalEntity](#), [getPickListValues](#), [getProperties](#), [getPublicClassificationCode](#), [getReferences](#), [getReport](#), [getRetention](#), [getSaveLog](#), [getSecurityClass](#), [getSecurityClassChangeDetails](#), [getSignificance](#), [getStatus](#), [getStub](#), [getSubEntities](#), [getSubEntityTemplates](#), [getSystemProperties](#), [getSystemProperty](#), [getTemplate](#), [getTitle](#), [getTransferDetails](#), [getType](#), [getVersionComment](#), [getVersionLabel](#), [getVersionSequence](#), [hasAIP](#), [hasDraft](#), [hasReferences](#), [hasVersions](#), [isClosed](#), [isPermanent](#), [modified](#), [opened](#), [save](#), [save](#), [search](#), [setCategories](#), [setClassificationCode](#), [setDescription](#), [setExternalIds](#), [setKeywords](#), [setOwner](#), [setSecurityClass](#), [setSignificance](#), [setStatus](#), [setTitle](#), [subEntityCount](#)

### Methods inherited from interface [com.imis.IAutoCloseable](#)

[close](#)

## Methods

### isChildClassificationCodeGenerated

public boolean **isChildClassificationCodeGenerated**()  
throws [StorageConnectorException](#)

Gets a value indicating whether or not child classification code is auto generated.

**Returns:**

true if child classification code is auto generated; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while getting child classification code generated value.

### setChildClassificationCodeGenerated

public void **setChildClassificationCodeGenerated**(boolean value)  
throws [StorageConnectorException](#)

Sets a value indicating whether or not child classification code is auto generated.

**Parameters:**

value - true if child classification code is auto generated; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while setting child classification code generated value.

### createFolder

public [IFolder](#) **createFolder**(String templateId)  
throws [javax.security.auth.login.LoginException](#),  
[StorageConnectorException](#),  
[IOException](#)

Creates a new IFolder or sub-folder entity instance.

**Note:** The instance is committed to the archive only if [IEntity.save\(\)](#) method is used on the new instance, otherwise it is discarded.

**Note:** This operation allows user to pass a reason for creating folder. Reason can be passed through [AuditLog](#) message.

**Parameters:**

templateId - the folder template identifier from the templates inventory.

**Returns:**

A new IFolder instance.

**Throws:**

[NullPointerException](#) - if templateId is a null reference.

(continued from last page)

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on closed archive is not allowed.
- Template identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating entity on archive.

IOException - if an I/O error occurs.

---

## createDocument

```
public IDocument createDocument(String templateId)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Creates a new IDocument entity instance.

**Note:** The instance is committed to the archive only if [IEntity.save\(\)](#) method is used on the new instance, otherwise it is discarded.

**Note:** This operation allows user to pass a reason for creating document. Reason can be passed through [AuditLog](#) message.

### Parameters:

templateId - the document template identifier from the templates inventory.

### Returns:

A new IDocument instance.

### Throws:

NullPointerException - if templateId is a null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on closed archive is not allowed.
- Template identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating entity on archive.

IOException - if an I/O error occurs.

## com.imis.storageconnector.entitymodel Interface IMoveDetails

public interface **IMoveDetails**  
extends

Defines an IEntity relocation details.

**See Also:**

[IEntity](#)

### Method Summary

<a href="#">IDirectoryEntity</a>	<a href="#">getAgent()</a> Gets the relocated IEntity agent.
String	<a href="#">getClassificationCode()</a> Gets the relocated IEntity classification code in canonical format.
String	<a href="#">getPublicClassificationCode()</a> Gets the relocated IEntity classification code in public format.
String	<a href="#">getReason()</a> Gets the reason for relocating the IEntity.
Calendar	<a href="#">moved()</a> Gets the IEntity relocation date and time.

### Methods

#### getClassificationCode

public String **getClassificationCode()**

Gets the relocated IEntity classification code in canonical format.

**Returns:**

A string that represents the moved IEntity classification code.

#### getPublicClassificationCode

public String **getPublicClassificationCode()**

Gets the relocated IEntity classification code in public format.

**Returns:**

A string that represents the moved IEntity public classification code.

#### getAgent

public [IDirectoryEntity](#) **getAgent()**

Gets the relocated IEntity agent.

(continued from last page)

**Returns:**

The user who relocated the IEntity.

---

**getReason**

```
public String getReason()
```

Gets the reason for relocating the IEntity.

**Returns:**

A reason for relocating the IEntity.

---

**moved**

```
public Calendar moved()
```

Gets the IEntity relocation date and time.

**Returns:**

The date and time when the IEntity was relocated.

## com.imis.storageconnector.entitymodel

### Interface IPhysicalEntity

public interface **IPhysicalEntity**  
extends

Describes a physical entity.

#### Method Summary

String	<a href="#"><u>getCurrentLocation()</u></a> Gets the physical entity current location.
String	<a href="#"><u>getCustodian()</u></a> Gets the physical entity current custodian.
String	<a href="#"><u>getDescription()</u></a> Gets the physical entity description.
String	<a href="#"><u>getHomeLocation()</u></a> Gets the physical entity home location.
String	<a href="#"><u>getId()</u></a> Gets the physical entity identifier.
String	<a href="#"><u>getStatus()</u></a> Gets the physical entity status.
Calendar	<a href="#"><u>returnDue()</u></a> Gets the date and time when CheckedOut physical entity is due to be CheckedIn.
void	<a href="#"><u>setCurrentLocation(String value)</u></a> Sets the physical entity current location.
void	<a href="#"><u>setCustodian(String value)</u></a> Sets the physical entity current custodian.
void	<a href="#"><u>setDescription(String value)</u></a> Sets the physical entity description.
void	<a href="#"><u>setHomeLocation(String value)</u></a> Sets the physical entity home location.
void	<a href="#"><u>setId(String value)</u></a> Sets the physical entity identifier.
void	<a href="#"><u>setReturnDue(Calendar value)</u></a> Sets the date and time when CheckedOut physical entity is due to be CheckedIn.
void	<a href="#"><u>setStatus(String value)</u></a> Sets the physical entity status.
Calendar	<a href="#"><u>statusChange()</u></a> Gets the last date and time when status changed its value.

## Methods

### getId

```
public String getId()  
    throws StorageConnectorException
```

Gets the physical entity identifier.

**Returns:**

A string that represents the physical entity identifier.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

### setId

```
public void setId(String value)  
    throws StorageConnectorException
```

Sets the physical entity identifier.

**Parameters:**

value - a string that represents the physical entity identifier.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

### getDescription

```
public String getDescription()  
    throws StorageConnectorException
```

Gets the physical entity description.

**Returns:**

The physical entity description.

**Throws:**



(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setDescription

```
public void setDescription(String value)  
    throws StorageConnectorException
```

Sets the physical entity description.

### Parameters:

value - the physical entity description.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getStatus

```
public String getStatus()  
    throws StorageConnectorException
```

Gets the physical entity status.

### Returns:

The physical entity status.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setStatus

```
public void setStatus(String value)  
    throws StorageConnectorException
```

Sets the physical entity status.

### Parameters:

value - the physical entity status.

### Throws:

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Error occurred while clearing property values.
- Property is not present in template properties.
- Property value is not allowed based on the current template settings.
- Error occurred while setting property values.

---

## statusChange

```
public Calendar statusChange()  
    throws StorageConnectorException
```

Gets the last date and time when status changed its value.

### Returns:

The last date and time when status changed its value or a null reference if initial status has not changed.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getHomeLocation

```
public String getHomeLocation()  
    throws StorageConnectorException
```

Gets the physical entity home location.

### Returns:

The physical entity home location.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setHomeLocation

```
public void setHomeLocation(String value)  
    throws StorageConnectorException
```

Sets the physical entity home location.

### Parameters:

value - the physical entity home location.

(continued from last page)

**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getCurrentLocation

```
public String getCurrentLocation()  
    throws StorageConnectorException
```

Gets the physical entity current location.

**Returns:**

The physical entity current location.

**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setCurrentLocation

```
public void setCurrentLocation(String value)  
    throws StorageConnectorException
```

Sets the physical entity current location.

**Parameters:**

value - the physical entity current location.

**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getCustodian

```
public String getCustodian()  
    throws StorageConnectorException
```

Gets the physical entity current custodian.

The physical entity custodian is internal personnel, if status is `CheckedIn` (normally owner) or external if status is `CheckedOut`.

(continued from last page)

**Returns:**

The physical entity current custodian.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setCustodian

```
public void setCustodian(String value)  
    throws StorageConnectorException
```

Sets the physical entity current custodian.

The physical entity custodian is internal personnel, if status is `CheckedIn` (normally owner) or external if status is `CheckedOut`.

**Parameters:**

value - the physical entity current custodian.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## returnDue

```
public Calendar returnDue()  
    throws StorageConnectorException
```

Gets the date and time when `CheckedOut` physical entity is due to be `CheckedIn`.

**Returns:**

The date and time when `CheckedOut` physical entity is due to be `CheckedIn` or a `null` reference if the date and time is not set.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Operation on entity opened in read-only mode is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
-

(continued from last page)

## setReturnDue

```
public void setReturnDue(Calendar value)  
    throws StorageConnectorException
```

Sets the date and time when CheckedOut physical entity is due to be CheckedIn.

### Parameters:

value - the date and time when CheckedOut physical entity is due to be CheckedIn or a null reference if the date and time is not set.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

## com.imis.storageconnector.entitymodel Interface IPickListValue

public interface **IPickListValue**  
extends

Represents the pick-list property value.

### Method Summary

String	<a href="#">getKey()</a> Gets the generic culture insensitive key of the pick-list property value.
String	<a href="#">getKey(Locale locale)</a> Gets the culture sensitive key of the pick-list property value or the culture insensitive key if pick-list value key is not localized into the specified <code>Locale</code> .
Object	<a href="#">getValue()</a> Gets the value of the pick-list property value.
boolean	<a href="#">isDefault()</a> Gets the value indicating whether the value is the default one for the owning pick-list value list.

### Methods

#### getKey

public String **getKey()**

Gets the generic culture insensitive key of the pick-list property value.

**Returns:**

The key of the pick-list property value or a null reference.

#### getValue

public Object **getValue()**

Gets the value of the pick-list property value.

**Returns:**

The value of the pick-list property value.

#### isDefault

public boolean **isDefault()**

Gets the value indicating whether the value is the default one for the owning pick-list value list.

**Returns:**

true if value is the default pick-list value; otherwise false.

## getKey

```
public String getKey(Locale locale)
```

Gets the culture sensitive key of the pick-list property value or the culture insensitive key if pick-list value key is not localized into the specified `Locale`.

### Parameters:

`locale` - a locale of the localized key to get.

### Returns:

The localized key of the pick-list property value.

## com.imis.storageconnector.entitymodel

### Interface IProperty

All Superinterfaces:

[IReadOnlyProperty](#)

public interface **IProperty**  
 extends [IReadOnlyProperty](#)

Defines a property of an IEntity.

A IProperty interface prescribes a way with which an application can interface with IEntity metadata attribute in read-write fashion. Each property is contained within an IEntity in a appropriate container. Since properties can be multi-valued, the interface provides and accepts a collection of property values, which need to be of same type. Convenience methods are also provided for single-value access. If the property type is a "complex" one, value factory methods are provided to instantiate new instances of complex values.

**Note:** Any changes made to the property value(s) are committed to the archive at IEntity commit stage by using [IEntity.save\(\)](#) method. If this method is not called, any changes made to the properties are discarded.

**See Also:**

[IReadOnlyProperty](#), [IEntity](#)

#### Method Summary

void	<a href="#">clear()</a> Clears the property values.
void	<a href="#">clear(boolean resetToInheritedValues)</a> Clears the property values.
int	<a href="#">committedValueCount()</a> Gets the property committed value count.
void	<a href="#">setValue(Class type, Object value)</a> Sets the property value.
void	<a href="#">setValue(String value)</a> Sets the property value using its equivalent string representation.
void	<a href="#">setValue(String value, java.text.Format format)</a> Sets the property value using its equivalent string representation using the specified format.
void	<a href="#">setValues(Class type, List values)</a> Sets the list of property values.
void	<a href="#">setValues(List values)</a> Sets the list of property values in its equivalent string representation.
void	<a href="#">setValues(List values, java.text.Format format)</a> Sets the list of property values in its equivalent string representation using the specified format.
void	<a href="#">setXmlValue(String value)</a> Sets the property value using its equivalent XML string representation.



void

[setXmlValues](#)(List values)

Sets the list of property values in its equivalent XML string representation.

**Methods inherited from interface** [com.imis.storageconnector.entitymodel.IReadOnlyProperty](#)

[areValuesInherited](#), [getDescription](#), [getEffectiveRights](#), [getLabel](#), [getName](#),  
[getPickListValues](#), [getType](#), [getValue](#), [getValue](#), [getValue](#), [getValues](#), [getValues](#),  
[getValues](#), [getXmlValue](#), [getXmlValues](#), [isAppendOnly](#), [isIncludedInAIP](#), [isInherited](#),  
[isMultiValue](#), [isNonEmpty](#), [isPickList](#), [isPublic](#), [isReadOnly](#), [isReadOnlyAfterCheckIn](#),  
[isReadOnlyAfterCreate](#), [isRequired](#), [isSearchable](#), [isUnique](#), [isVersionable](#), [valueCount](#)

## Methods

### committedValueCount

```
public int committedValueCount()
```

Gets the property committed value count.

**Returns:**

The property committed value count.

### setValue

```
public void setValue(String value)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Sets the property value using its equivalent string representation.

**Note:** Communication with archive is needed when property type is STRINGMAX.**Note:** This operation allows user to pass a reason for setting STRINGMAX property value. Reason can be passed through [AuditLog](#) message.**Parameters:**

value - the property value in its equivalent string representation.

**Throws:**

UnsupportedOperationException - if values for property cannot be set through strings.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.
- Error occurred while setting property values.

IOException - if an I/O error occurs.

(continued from last page)

## setValue

```
public void setValue(String value,  
    java.text.Format format)  
    throws javax.security.auth.login.LoginException,  
        StorageConnectorException,  
        IOException
```

Sets the property value using its equivalent string representation using the specified format.

**Note:** Communication with archive is needed when property type is STRINGMAX.

**Note:** This operation allows user to pass a reason for setting STRINGMAX property value. Reason can be passed through [AuditLog](#) message.

### Parameters:

value - the property value in its equivalent string representation.  
format - an object that supplies locale-sensitive formatting information.

### Throws:

UnsupportedOperationException - if values for property cannot be set through strings.  
CancellationException - if an operation was canceled by the user.  
LoginException - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.
- Error occurred while setting property values.

IOException - if an I/O error occurs.

---

## setXmlValue

```
public void setXmlValue(String value)  
    throws javax.security.auth.login.LoginException,  
        StorageConnectorException,  
        IOException
```

Sets the property value using its equivalent XML string representation.

**Note:** Communication with archive is needed when property type is STRINGMAX.

**Note:** This operation allows user to pass a reason for setting STRINGMAX property value. Reason can be passed through [AuditLog](#) message.

### Parameters:

value - the property value in its equivalent string XML representation.

### Throws:

UnsupportedOperationException - if unsupported property type.  
CancellationException - if an operation was canceled by the user.  
LoginException - if a login error occurs.

---

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.
- Error occurred while setting property values.

IOException - if an I/O error occurs.

---

## setValue

```
public void setValue(Class type,
                    Object value)
    throws StorageConnectorException
```

Sets the property value.

The property value type must be selected according to the [PropertyType](#).

This method provides a means of setting new property value, to modify it or to delete it. The discrete value type is set by using the appropriate native type as generic parameter. Implementation will use the value instance directly. It copies it to the property internal container so you can modify or discard it after it's set.

### Parameters:

type - the type that interfaces with [PropertyType](#).  
 value - the property value.

### Throws:

UnsupportedOperationException - if unsupported property type.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Generic argument type is not compatible with property type.
- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.
- Generic argument value of type T should be copied into a new BinaryValue or StringMaxValue, compatible with underlying storage.
- Error occurred while setting property values.

---

## setValues

```
public void setValues(List values)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Sets the list of property values in its equivalent string representation.

**Note:** Communication with archive is needed when property type is STRINGMAX.

**Note:** This operation allows user to pass a reason for setting STRINGMAX property values. Reason can be passed through [AuditLog](#) message.

(continued from last page)

**Parameters:**

values - the list of property values in its equivalent string representation.

**Throws:**

UnsupportedOperationException - if unsupported property type.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.
- Error occurred while setting property values.

IOException - if an I/O error occurs.

---

**setValues**

```
public void setValues(List values,  
    java.text.Format format)  
    throws javax.security.auth.login.LoginException,  
    StorageConnectorException,  
    IOException
```

Sets the list of property values in its equivalent string representation using the specified format.

**Note:** Communication with archive is needed when property type is STRINGMAX.

**Note:** This operation allows user to pass a reason for setting STRINGMAX property values. Reason can be passed through [AuditLog](#) message.

**Parameters:**

values - the list of property values in its equivalent string representation.

format - an object that supplies locale-sensitive formatting information.

**Throws:**

NullPointerException - if values is a null reference.

UnsupportedOperationException - if unsupported property type.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.
- Error occurred while setting property values.

IOException - if an I/O error occurs.

(continued from last page)

## setXmlValues

```
public void setXmlValues(List values)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Sets the list of property values in its equivalent XML string representation.

**Note:** Communication with archive is needed when property type is STRINGMAX.

**Note:** This operation allows user to pass a reason for setting STRINGMAX property values. Reason can be passed through [AuditLog](#) message.

### Parameters:

values - the list of property values in its equivalent XML string representation.

### Throws:

NullPointerException - if values is a null reference.

UnsupportedOperationException - if unsupported property type.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.
- Error occurred while setting property values.

IOException - if an I/O error occurs.

---

## setValues

```
public void setValues(Class type,
    List values)
    throws StorageConnectorException
```

Sets the list of property values.

The property value type must be selected according to the [PropertyType](#).

This method provides a means of setting new property values, to modify them or to delete them. The discrete values type is set by using the appropriate native type as generic parameter.

The list of values suggests their sequence and will be stored in such order. If the passed generic type parameter is not compatible with the property type an exception is thrown. Implementation will not use the container directly but rather copy the values from it.

### Parameters:

type - the type that interfaces with [PropertyType](#).

values - the list of property values.

### Throws:

NullPointerException - if values is a null reference.

UnsupportedOperationException - if unsupported property type.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Generic argument type is not compatible with property type.
- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.
- Generic argument value of type `T` should be copied into a new `BinaryValue` or `StringMaxValue`, compatible with underlying storage.
- Error occurred while setting property values.

---

## clear

```
public void clear()  
    throws StorageConnectorException
```

Clears the property values.

This method does nothing if the property had no values. On inherited property this method resets property values to inherited values if set on parent entity.

**Note:** Cleared properties are committed to the archive at `IEntity` commit stage by using [IEntity.save\(\)](#) method. If this method is not called, any changes made to the property are discarded.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Error occurred while clearing property values.

---

## clear

```
public void clear(boolean resetToInheritedValues)  
    throws StorageConnectorException
```

Clears the property values.

This method does nothing if the property had no values.

**Note:** Cleared properties are committed to the archive at `IEntity` commit stage by using [IEntity.save\(\)](#) method. If this method is not called, any changes made to the property are discarded.

### Parameters:

`resetToInheritedValues` - true resets property values on inherited property to inherited values if set on parent entity; otherwise false.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Error occurred while clearing property values.

## com.imis.storageconnector.entitymodel Interface IPropertyACLEntry

public interface **IPropertyACLEntry**  
extends

Represents an Access Control List entry for a property attached to a subject.

The `IPropertyACLEntry` object represents ACL entry bound to the subject for the owner `IEntity`. It provides access to the inherited and explicit set of rights defined for the subject. It also enables subjects with `ChangeRights` right to modify the set of rights of the `IEntity`, its validity period and have the option of deleting the entry from the set of rights.

**See Also:**

[IACL](#), [IEntity](#)

### Method Summary

void	<a href="#">delete()</a> Deletes the ACL entry from the owner <code>IACL</code> .
<a href="#">IReadOnlyPropertyRights</a>	<a href="#">getEffectiveRights()</a> Gets the users effective rights on the <code>IProperty</code> to which ACL entry is bound to.
<a href="#">IPropertyRights</a>	<a href="#">getExplicitAllowRights()</a> Gets the Explicit Allow set of rights for the subject in question.
<a href="#">IPropertyRights</a>	<a href="#">getExplicitDenyRights()</a> Gets the Explicit Deny set of rights for the subject in question.
<a href="#">IReadOnlyPropertyRights</a>	<a href="#">getInheritedAllowRights()</a> Gets the Inherited Allow set of rights for the subject in question.
<a href="#">IReadOnlyPropertyRights</a>	<a href="#">getInheritedDenyRights()</a> Gets the Inherited Deny set of rights for the subject in question.
<a href="#">IACL</a>	<a href="#">getOwner()</a> Gets the owner <code>IACL</code> .
String	<a href="#">getPropertyName()</a> Gets the property name which uniquely identifies the <code>IProperty</code> .
String	<a href="#">getSubject()</a> Gets the subject to which the ACL entry is bound to.
<a href="#">ACLEntrySubjectType</a>	<a href="#">getSubjectType()</a> Gets the ACL entry subject type.

### Methods

#### getOwner

public [IACL](#) **getOwner()**

---

(continued from last page)

Gets the owner IACL.

**Note:** This property never changes its value after IPropertyACLEntry is instantiated.

**Returns:**

A owner IACL instance.

---

## getSubjectType

```
public ACLEntrySubjectType getSubjectType()
```

Gets the ACL entry subject type.

**Returns:**

The ACL entry subject type.

---

## getSubject

```
public String getSubject()
```

Gets the subject to which the ACL entry is bound to.

**Returns:**

The ACL entry subject.

---

## getPropertyName

```
public String getPropertyName()
```

Gets the property name which uniquely identifies the IProperty.

**Returns:**

A string that represents the property name.

---

## getInheritedDenyRights

```
public IReadOnlyPropertyRights getInheritedDenyRights()  
throws StorageConnectorException
```

Gets the Inherited Deny set of rights for the subject in question.

**Returns:**

The IReadOnlyPropertyRights to the Inherited Deny set of rights.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

**See Also:**

[getSubject\(\)](#)

---

## getInheritedAllowRights

```
public IReadOnlyPropertyRights getInheritedAllowRights()  
throws StorageConnectorException
```

Gets the Inherited Allow set of rights for the subject in question.

---



(continued from last page)

**Returns:**

The `IReadOnlyPropertyRights` to the `Inherited Allow` set of rights.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

**See Also:**

[getSubject\(\)](#)

---

## getExplicitDenyRights

```
public IPropertyRights getExplicitDenyRights()  
    throws StorageConnectorException
```

Gets the `Explicit Deny` set of rights for the subject in question.

Caller must have `ChangeRights` right to the owner entity to be allowed to modify set of rights.

**Returns:**

The `IPropertyRights` to the `Explicit Deny` set of rights.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while adding `ExplicitDeny` ACL entry to the entity.

**See Also:**

[getSubject\(\)](#)

---

## getExplicitAllowRights

```
public IPropertyRights getExplicitAllowRights()  
    throws StorageConnectorException
```

Gets the `Explicit Allow` set of rights for the subject in question.

Caller must have `ChangeRights` right to the owner entity to be allowed to modify set of rights.

**Returns:**

The `IPropertyRights` to the `Explicit Allow` set of rights.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Error occurred while adding `ExplicitAllow` ACL entry to the entity.

**See Also:**

[getSubject\(\)](#)

---

(continued from last page)

## getEffectiveRights

```
public IReadOnlyPropertyRights getEffectiveRights()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the users effective rights on the `IProperty` to which ACL entry is bound to.

### Returns:

The users effective rights on the `IProperty` to which ACL entry is bound to or a `null` reference if none available.

### Throws:

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

`IOException` - if an I/O error occurs.

---

## delete

```
public void delete()  
    throws StorageConnectorException
```

Deletes the ACL entry from the owner `IACL`.

**Note:** Removal is not committed to the `Archive` unless the owner entity of the owner `IACL` object is committed to the archive using [IEntity.save\(\)](#) method. Caller must have `ChangeRights` right to the owner entity to be allowed to call this method.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while removing set of rights.

### See Also:

[IACL](#)

## com.imis.storageconnector.entitymodel

### Interface IPropertyRights

All Superinterfaces:

[IReadOnlyPropertyRights](#)

public interface **IPropertyRights**

extends [IReadOnlyPropertyRights](#)

Allows the caller to access the set of rights on a property for a certain subject in a read-write fashion.

The following rights are defined:

Right	Description
CreateAccess	Subject has the right to add values to an empty property metadata attribute.
ReadAccess	Subject has read access to property metadata attribute.
WriteAccess	Subject has write access to property metadata attribute.
DeleteAccess	Subject has the right to empty out the values of a property metadata attribute.

See Also:

[IPropertyACLEntry](#), [IReadOnlyEntityRights](#)

### Method Summary

void	<a href="#">delete()</a> Removes the property set of rights from the owning IPropertyACLEntry.
void	<a href="#">setCreateAccess</a> (boolean value) Sets the value indicating whether the subject has the right to add new values to a previously empty property metadata attribute.
void	<a href="#">setDeleteAccess</a> (boolean value) Sets the value indicating whether the subject has the right to empty out the values of the property metadata attribute.
void	<a href="#">setPropagated</a> (boolean propagated) Sets the value indicating whether or not the entry is propagated.
void	<a href="#">setReadAccess</a> (boolean value) Sets the value indicating whether the subject has the right to access (read) values of the property metadata attribute.
void	<a href="#">setValidFrom</a> (Calendar value) Sets the Calendar after which this entry is valid when calculating the set of rights for the owning entity.
void	<a href="#">setValidTo</a> (Calendar value) Sets the Calendar before which this entry is valid when calculating the set of rights for the owning entity.
void	<a href="#">setWriteAccess</a> (boolean value) Sets the value indicating whether the subject has the right to modify the values of the property metadata attribute.

Methods inherited from interface [com.imis.storageconnector.entitymodel.IReadOnlyPropertyRights](#)

[createAccess](#), [deleteAccess](#), [getOwner](#), [isPropagated](#), [readAccess](#), [validFrom](#), [validTo](#), [writeAccess](#)

---

## Methods

### setCreateAccess

public void **setCreateAccess**(boolean value)  
throws [StorageConnectorException](#)

Sets the value indicating whether the subject has the right to add new values to a previously empty property metadata attribute.

**Parameters:**

value - true to grant CreateAccess right; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of right.

**See Also:**

[IPropertyACLEntry.getSubject\(\)](#)

[IPropertyACLEntry.getPropertyName\(\)](#)

---

### setReadAccess

public void **setReadAccess**(boolean value)  
throws [StorageConnectorException](#)

Sets the value indicating whether the subject has the right to access (read) values of the property metadata attribute.

**Parameters:**

value - true to grant ReadAccess right; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of right.

**See Also:**

[IPropertyACLEntry.getSubject\(\)](#)

[IPropertyACLEntry.getPropertyName\(\)](#)

---

### setWriteAccess

public void **setWriteAccess**(boolean value)  
throws [StorageConnectorException](#)

(continued from last page)

Sets the value indicating whether the subject has the right to modify the values of the property metadata attribute.

**Parameters:**

value - true to grant WriteAccess right; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of right.

**See Also:**

[IPropertyACLEntry.getSubject\(\)](#)

[IPropertyACLEntry.getPropertyName\(\)](#)

---

## setDeleteAccess

```
public void setDeleteAccess(boolean value)  
    throws StorageConnectorException
```

Sets the value indicating whether the subject has the right to empty out the values of the property metadata attribute.

**Parameters:**

value - true to grant DeleteAccess right; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting granted value of right.

**See Also:**

[IPropertyACLEntry.getSubject\(\)](#)

[IPropertyACLEntry.getPropertyName\(\)](#)

---

## setPropagated

```
public void setPropagated(boolean propagated)  
    throws StorageConnectorException
```

Sets the value indicating whether or not the entry is propagated.

**Parameters:**

propagated - the propagated value.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
  - Operation on entity opened in read-only mode is not allowed.
-

(continued from last page)

## setValidFrom

```
public void setValidFrom(Calendar value)
    throws StorageConnectorException
```

Sets the Calendar after which this entry is valid when calculating the set of rights for the owning entity.

### Parameters:

value - a Calendar value that represents the entry validity start timestamp or null if there is no lower limit.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting Calendar value for set of rights.

---

## setValidTo

```
public void setValidTo(Calendar value)
    throws StorageConnectorException
```

Sets the Calendar before which this entry is valid when calculating the set of rights for the owning entity.

### Parameters:

value - a Calendar value that represents the entry validity end timestamp or null if there is no upper limit.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while setting Calendar value for set of rights.

---

## delete

```
public void delete()
    throws StorageConnectorException
```

Removes the property set of rights from the owning IPropertyACLEntry.

Removal is not committed to the archive unless owning IEntity is committed to the Archive instance using [IEntity.save\(\)](#) method.

Caller must have ChangeRights right to the owner entity to be allowed to call this method.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while removing set of rights.

### See Also:

[IPropertyACLEntry](#)

## com.imis.storageconnector.entitymodel Interface IReadOnlyContent

All Subinterfaces:

[IContent](#)

public interface **IReadOnlyContent**  
extends

Defines a collection of the `IContentPart` objects.

A `IReadOnlyContent` describes a collection of content parts assigned to an entity. It represents a digital (binary) collection of chapters. It has its attributes and methods for accessing the subsequent binary parts in read-only fashion.

**See Also:**

[IEntity](#), [IDocument](#)

### Method Summary

<a href="#">IReadOnlyPropertyRights</a>	<a href="#">getEffectiveRights()</a> Gets the effective rights on the content if content has ACL defined.
String	<a href="#">getName()</a> Gets the content unique name.
com.imis.util.IReadOnlyList	<a href="#">getParts()</a> Gets the read-only list of <code>IContentPart</code> objects.
boolean	<a href="#">isPublic()</a> Gets a value indicating whether the property is public.
boolean	<a href="#">isReadOnly()</a> Gets a value indicating whether the property is read-only.
boolean	<a href="#">isReadOnlyAfterCheckIn()</a> Gets a value indicating whether the property is read-only after entity has been checked in.
boolean	<a href="#">isReadOnlyAfterCreate()</a> Gets a value indicating whether the property is read-only after entity has been created.
boolean	<a href="#">isVersionable()</a> Gets a value indicating whether the property is versionable.
int	<a href="#">partsCount()</a> Gets the content parts count.

### Methods

#### getName

```
public String getName()
```

Gets the content unique name.

**Returns:**

(continued from last page)

A string that represents the content unique name.

---

## getEffectiveRights

```
public IReadOnlyPropertyRights getEffectiveRights()
```

Gets the effective rights on the content if content has ACL defined.

**Returns:**

The read-only set of rights describing users effective rights on the content or a null reference if none available.

---

## partsCount

```
public int partsCount()
```

Gets the content parts count.

**Returns:**

The number of content parts.

---

## getParts

```
public com.imis.util.IReadOnlyList getParts()
```

Gets the read-only list of `IContentPart` objects.

This method provides access to the ordered collection of content parts assigned to this content instance.

**Returns:**

A read-only list of `IContentPart` objects.

**See Also:**

[IContentPart](#)

---

## isReadOnly

```
public boolean isReadOnly()
```

Gets a value indicating whether the property is read-only.

Read-only properties cannot be modified. Any attempt made to modify such a property will cause [IEntity.save\(\)](#) method to fail.

**Returns:**

true if the property is read-only; otherwise false.

---

## isReadOnlyAfterCreate

```
public boolean isReadOnlyAfterCreate()
```

Gets a value indicating whether the property is read-only after entity has been created.

Read-only properties cannot be modified after entity has been created. Any attempt made to modify such a property will cause [IEntity.save\(\)](#) method to fail.

**Returns:**

true if the property is read-only after create; otherwise false.



(continued from last page)

## isReadOnlyAfterCheckIn

```
public boolean isReadOnlyAfterCheckIn()
```

Gets a value indicating whether the property is read-only after entity has been checked in.

Read-only properties cannot be modified after entity has been checked in. Any attempt made to modify such a property will cause [IEntity.save\(\)](#) method to fail.

**Returns:**

true if the property is read-only after check in; otherwise false.

---

## isVersionable

```
public boolean isVersionable()
```

Gets a value indicating whether the property is versionable.

**Returns:**

true if the property is versionable; otherwise false.

---

## isPublic

```
public boolean isPublic()
```

Gets a value indicating whether the property is public.

Public properties can be accessed by any authenticated user regardless if the entity ACL allows it or not.

**Returns:**

true if the property is public; otherwise false.

---

## com.imis.storageconnector.entitymodel Interface IReadOnlyEntityReference

public interface **IReadOnlyEntityReference**  
extends

Defines the read-only entity reference.

### Method Summary

Calendar	<a href="#"><u>created()</u></a> Gets the creation time-stamp of the entity reference.
void	<a href="#"><u>delete()</u></a> Deletes the entity reference and associated resources.
<a href="#"><u>IEntityReference</u></a>	<a href="#"><u>edit()</u></a> Opens an entity reference in read-write mode.
<a href="#"><u>IDirectoryEntity</u></a>	<a href="#"><u>getCreator()</u></a> Gets the creator <code>IDirectoryEntity</code> of the entity reference.
String	<a href="#"><u>getDescription()</u></a> Gets the description of the entity reference.
<a href="#"><u>IReadOnlyEntityRights</u></a>	<a href="#"><u>getEffectiveRights()</u></a> Gets the users effective rights on the entity reference.
String	<a href="#"><u>getId()</u></a> Gets the identifier of the entity reference.
<a href="#"><u>IDirectoryEntity</u></a>	<a href="#"><u>getModifiedBy()</u></a> Gets the last modifier directory entity.
com.imis.util.IReadOnlyEntityList	<a href="#"><u>getReferenceIds()</u></a> Gets the referenced entities identifiers.
String	<a href="#"><u>getTitle()</u></a> Gets the title of the entity reference.
Calendar	<a href="#"><u>modified()</u></a> Gets the last modification time-stamp of the entity reference.

### Methods

#### getId

public String **getId()**

Gets the identifier of the entity reference.

#### Returns:

A string that represents the identifier the entity reference.

## getTitle

```
public String getTitle()  
    throws StorageConnectorException
```

Gets the title of the entity reference.

**Returns:**

The title of the entity reference.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getDescription

```
public String getDescription()  
    throws StorageConnectorException
```

Gets the description of the entity reference.

**Returns:**

The description of the entity reference.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## created

```
public Calendar created()  
    throws StorageConnectorException
```

Gets the creation time-stamp of the entity reference.

**Returns:**

The date and time when the entity reference was created.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getCreator

```
public IDirectoryEntity getCreator()  
    throws StorageConnectorException
```

Gets the creator [IDirectoryEntity](#) of the entity reference.

**Returns:**

(continued from last page)

The creator `IDirectoryEntity` of the entity reference.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## modified

```
public Calendar modified()  
    throws StorageConnectorException
```

Gets the last modification time-stamp of the entity reference.

**Returns:**

The date and time when the describing entity was last modified.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getModifiedBy

```
public IDirectoryEntity getModifiedBy()  
    throws StorageConnectorException
```

Gets the last modifier directory entity.

**Returns:**

The last modifier directory entity or a null reference, if this instance has no last modifier directory entity.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getEffectiveRights

```
public IReadOnlyEntityRights getEffectiveRights()
```

Gets the users effective rights on the entity reference.

**Note:** The following rights are not applicable on entity references: `MoveAccess`, `CreateSubEntities`, `ChangeSecurityClass`, `ChangeStatus`, `ChangeRetention` and `CreateReferences`.

**Returns:**

The read-only set of rights describing users effective rights on the entity reference.

---

## getReferenceIds

```
public com.imis.util.IReadOnlyList getReferenceIds()  
    throws StorageConnectorException
```

Gets the referenced entities identifiers.

(continued from last page)

**Returns:**

The read-only list of the referenced entities identifiers.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## edit

```
public IEntityReference edit()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Opens an entity reference in read-write mode.

**Note:** This operation allows user to pass a reason for opening entity reference. Reason can be passed through [AuditLog](#) message.

**Returns:**

An [IEntityReference](#) instance.

**Throws:**

[CancellationException](#) - if an operation was canceled by the user.

[LoginException](#) - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening entity on archive.

[IOException](#) - if an I/O error occurs.

---

## delete

```
public void delete()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Deletes the entity reference and associated resources.

Use this method to remove all resources associated with the entity reference and the entity reference itself. Delete right is required for this operation to succeed.

**Note:** This operation allows user to pass a reason for deleting entity reference. Reason can be passed through [AuditLog](#) message.

This operation requires description for deleted entity reference. This method will fail if the entity reference does not have its own description.

This operation is irreversible. Entity reference will not be accessible after a successful call is completed.

**Throws:**

[CancellationException](#) - if an operation was canceled by the user.

[LoginException](#) - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on unsaved new entity is not allowed.
- Error occurred while closing entity on archive.
- Error occurred while removing entity on archive.

`IOException` - if an I/O error occurs.

## com.imis.storageconnector.entitymodel Interface IReadOnlyEntityRights

All Subinterfaces:

[IEntityRights](#)

public interface **IReadOnlyEntityRights**  
extends

Allows the caller to access the set of rights for a certain subject identified by the subject in a read-only fashion.

The following rights are defined:

Right	Description
ReadAccess	Subject has read access to entity metadata attributes and other properties.
WriteAccess	Subject has write access to entity metadata attributes and other properties.
MoveAccess	Subject has the right to move the entity within the archive.
DeleteAccess	Subject has the right to delete the entity from the archive.
ChangeRights	Subject has the right to modify the entity explicit security descriptor.
CreateSubEntities	Subject has the right to create new IEntity sub-entities.
ChangeSecurityClass	Subject has the right to set the entity security class.
ChangeStatus	Subject has the right to set the entity status.
ChangeRetention	Subject has the right to change the entity retention.
CreateReferences	Subject has the right to create entity references on the entity.

See Also:

[IEntity](#), [IEntityRights](#)

### Method Summary

boolean	<a href="#">changeRetention()</a> Gets the value indicating whether the subject has the right to change the current entity retention.
boolean	<a href="#">changeRights()</a> Gets the value indicating whether the subject has the right to modify explicit ACL entries bound to the owning entity.
boolean	<a href="#">changeSecurityClass()</a> Gets the value indicating whether the subject has the right to change the current entity security class.
boolean	<a href="#">changeStatus()</a> Gets the value indicating whether the subject has the right to change the current entity status.
boolean	<a href="#">createReferences()</a> Gets the value indicating whether the subject has the right to create entity reference.
boolean	<a href="#">createSubEntities()</a> Gets the value indicating whether the subject has the right to create sub-entities relative to the current entity.

boolean	<a href="#">deleteAccess()</a> Gets the value indicating whether the subject has the right to delete the owning entity from the archive.
<a href="#">IEntityACLEntry</a>	<a href="#">getOwner()</a> Gets the owner IEntityACLEntry.
boolean	<a href="#">isPropagated()</a> Gets the value indicating whether or not the entry is propagated.
boolean	<a href="#">moveAccess()</a> Gets the value indicating whether the subject has the right to move the current entity.
boolean	<a href="#">readAccess()</a> Gets the value indicating whether the subject has the right to access (read) entity metadata attributes and other properties.
Calendar	<a href="#">validFrom()</a> Gets the Calendar after which this entry is valid when calculating the set of rights for the owning entity.
Calendar	<a href="#">validTo()</a> Gets the Calendar before which this entry is valid when calculating the set of rights for the owning entity.
boolean	<a href="#">writeAccess()</a> Gets the value indicating whether the subject has the right to add or modify entity metadata attributes and other properties.

## Methods

### getOwner

public [IEntityACLEntry](#) **getOwner()**

Gets the owner IEntityACLEntry.

The return value never changes after entity rights is instantiated.

**Returns:**

The owner IEntityACLEntry or a null reference if this is a read-only entity rights.

### readAccess

public boolean **readAccess()**

Gets the value indicating whether the subject has the right to access (read) entity metadata attributes and other properties.

**Returns:**

true if the ReadAccess right is granted; otherwise false.

**See Also:**

[IEntityACLEntry.getSubject\(\)](#)

### writeAccess

public boolean **writeAccess()**



(continued from last page)

Gets the value indicating whether the subject has the right to add or modify entity metadata attributes and other properties.

**Returns:**

true if the WriteAccess right is granted; otherwise false.

**See Also:**

[IEntityACLEntry.GetSubject\(\)](#)

---

## moveAccess

```
public boolean moveAccess()
```

Gets the value indicating whether the subject has the right to move the current entity.

**Note:** This right is not applicable on entity references.

**Returns:**

true if the MoveAccess right is granted; otherwise false.

**See Also:**

[IEntityACLEntry.GetSubject\(\)](#)

---

## deleteAccess

```
public boolean deleteAccess()
```

Gets the value indicating whether the subject has the right to delete the owning entity from the archive.

**Returns:**

true if the DeleteAccess right is granted; otherwise false.

**See Also:**

[IEntityACLEntry.GetSubject\(\)](#)

---

## changeRights

```
public boolean changeRights()
```

Gets the value indicating whether the subject has the right to modify explicit ACL entries bound to the owning entity.

**Returns:**

true if the ChangeRights right is granted; otherwise false.

**See Also:**

[IEntityACLEntry.GetSubject\(\)](#)

---

## createSubEntities

```
public boolean createSubEntities()
```

Gets the value indicating whether the subject has the right to create sub-entities relative to the current entity.

**Note:** This right is not applicable on entity references.

**Returns:**

true if the CreateSubEntities right is granted; otherwise false.

**See Also:**

[IEntityACLEntry.GetSubject\(\)](#)

---

## changeSecurityClass

```
public boolean changeSecurityClass()
```

Gets the value indicating whether the subject has the right to change the current entity security class.

**Note:** This right is not applicable on entity references.

**Returns:**

true if the ChangeSecurityClass right is granted; otherwise false.

**See Also:**

[IEntityACLEntry.getSubject\(\)](#)

---

## changeStatus

```
public boolean changeStatus()
```

Gets the value indicating whether the subject has the right to change the current entity status.

**Note:** This right is not applicable on entity references.

**Returns:**

true if the ChangeStatus right is granted; otherwise false.

**See Also:**

[IEntityACLEntry.getSubject\(\)](#)

---

## changeRetention

```
public boolean changeRetention()
```

Gets the value indicating whether the subject has the right to change the current entity retention.

**Note:** This right is not applicable on entity references.

**Returns:**

true if the ChangeRetention right is granted; otherwise false.

**See Also:**

[IEntityACLEntry.getSubject\(\)](#)

---

## createReferences

```
public boolean createReferences()
```

Gets the value indicating whether the subject has the right to create entity reference.

**Note:** This right is not applicable on entity references.

**Returns:**

true if the CreateReferences right is granted; otherwise false.

**See Also:**

[IEntityACLEntry.getSubject\(\)](#)

---

(continued from last page)

## isPropagated

```
public boolean isPropagated()
```

Gets the value indicating whether or not the entry is propagated.

**Returns:**

true if the entry is propagated; otherwise false.

---

## validFrom

```
public Calendar validFrom()
```

Gets the Calendar after which this entry is valid when calculating the set of rights for the owning entity.

**Returns:**

A Calendar value that represents the entry validity start timestamp or null if there is no lower limit.

---

## validTo

```
public Calendar validTo()
```

Gets the Calendar before which this entry is valid when calculating the set of rights for the owning entity.

**Returns:**

A Calendar value that represents the entry validity end timestamp or null if there is no upper limit.

## com.imis.storageconnector.entitymodel Interface IReadOnlyProperty

All Subinterfaces:

[IProperty](#)

public interface **IReadOnlyProperty**  
extends

Defines a read-only property of an entity.

A `IReadOnlyProperty` interface prescribes a way with which an application can interface with `IEntity` metadata attribute in a read-only fashion. Each property is contained within an `IEntity` in a appropriate container. Since `IReadOnlyProperty` can be multi-valued, the interface provides access to the collection of property values, which are of the same type. Convenience methods are also provided for single-value access.

See Also:

[IProperty](#), [IEntity](#)

### Method Summary

boolean	<a href="#">areValuesInherited()</a> Gets the value indicating whether or not the property values are inherited from hierarchy.
String	<a href="#">getDescription()</a> Gets the property description.
<a href="#">IReadOnlyPropertyRights</a>	<a href="#">getEffectiveRights()</a> Gets the users effective rights on the property if the property has ACL defined.
String	<a href="#">getLabel()</a> Gets the property label.
String	<a href="#">getName()</a> Gets the property unique name.
com.imis.util.IReadOnlyList	<a href="#">getPickListValues()</a> Gets the read-only list of pick-list property values.
<a href="#">PropertyType</a>	<a href="#">getType()</a> Gets the property type.
String	<a href="#">getValue()</a> Gets the property value as its equivalent string representation.
Object	<a href="#">getValue(Class type)</a> Gets the property value.
String	<a href="#">getValue(java.text.Format format)</a> Gets the property value as its equivalent string representation using the specified format.
com.imis.util.IReadOnlyList	<a href="#">getValues()</a> Gets the read-only list of property values in its equivalent string representation.
com.imis.util.IReadOnlyList	<a href="#">getValues(Class type)</a> Gets the read-only list of property values.

com.imis.util.IReadOnlyList	<a href="#">getValues</a> (java.text.Format format) Gets the read-only list of property values in its equivalent string representation using the specified format.
String	<a href="#">getXmlValue</a> () Gets the property value as its equivalent XML string representation.
com.imis.util.IReadOnlyList	<a href="#">getXmlValues</a> () Gets the read-only list of property values in its equivalent XML string representation.
boolean	<a href="#">isAppendOnly</a> () Gets a value indicating whether the property value is append-only.
boolean	<a href="#">isIncludedInAIP</a> () Gets a value indicating whether the property value is included in Archive Informational Package (AIP).
boolean	<a href="#">isInherited</a> () Gets a value indicating whether the property value is inherited.
boolean	<a href="#">isMultiValue</a> () Gets a value indicating whether the property can hold more than one value.
boolean	<a href="#">isNonEmpty</a> () Gets a value indicating whether the property is non-empty.
boolean	<a href="#">isPickList</a> () Gets a value indicating whether the property values are predefined and can be chosen from a list of values.
boolean	<a href="#">isPublic</a> () Gets a value indicating whether the property is public.
boolean	<a href="#">isReadOnly</a> () Gets a value indicating whether the property is read-only.
boolean	<a href="#">isReadOnlyAfterCheckIn</a> () Gets a value indicating whether the property is read-only after entity has been checked in.
boolean	<a href="#">isReadOnlyAfterCreate</a> () Gets a value indicating whether the property is read-only after entity has been created.
boolean	<a href="#">isRequired</a> () Gets a value indicating whether the property is required.
boolean	<a href="#">isSearchable</a> () Gets a value indicating whether the property value is indexed.
boolean	<a href="#">isUnique</a> () Gets a value indicating whether the property is unique.
boolean	<a href="#">isVersionable</a> () Gets a value indicating whether the property is versionable.
int	<a href="#">valueCount</a> () Gets the number of values in the property value container.

## Methods

(continued from last page)

## getName

```
public String getName()
```

Gets the property unique name.

**Returns:**

The property unique name.

---

## getDescription

```
public String getDescription()
```

Gets the property description.

**Returns:**

The property description or an empty string.

---

## getLabel

```
public String getLabel()
```

Gets the property label.

**Returns:**

The property label or an empty string.

---

## getType

```
public PropertyType getType()
```

Gets the property type.

**Returns:**

The property type.

---

## isPublic

```
public boolean isPublic()
```

Gets a value indicating whether the property is public.

Public properties can be accessed by any authenticated user regardless if the entity ACL allows it or not.

**Returns:**

true if the property is public; otherwise false.

---

## isMultiValue

```
public boolean isMultiValue()
```

Gets a value indicating whether the property can hold more than one value.

**Returns:**

true if the property is can hold more than one value; otherwise false.

---

(continued from last page)

## isRequired

```
public boolean isRequired()
```

Gets a value indicating whether the property is required.

Required properties need to have a value assigned to them at [IEntity.save\(\)](#) stage otherwise the method will fail.

**Returns:**

true if the property is required; otherwise false.

---

## isNonEmpty

```
public boolean isNonEmpty()
```

Gets a value indicating whether the property is non-empty.

Non-empty properties cannot have empty values. Saving such a property with an empty value will cause [IEntity.save\(\)](#) method to fail.

**Returns:**

true if the property is non-empty; otherwise false.

---

## isReadOnly

```
public boolean isReadOnly()
```

Gets a value indicating whether the property is read-only.

Read-only properties cannot be modified. Any attempt made to modify such a property will cause [IEntity.save\(\)](#) method to fail.

**Returns:**

true if the property is read-only; otherwise false.

---

## isReadOnlyAfterCreate

```
public boolean isReadOnlyAfterCreate()
```

Gets a value indicating whether the property is read-only after entity has been created.

Read-only properties cannot be modified after entity has been created. Any attempt made to modify such a property will cause [IEntity.save\(\)](#) method to fail.

**Returns:**

true if the property is read-only; otherwise false.

---

## isReadOnlyAfterCheckIn

```
public boolean isReadOnlyAfterCheckIn()
```

Gets a value indicating whether the property is read-only after entity has been checked in.

Read-only properties cannot be modified after entity has been checked in. Any attempt made to modify such a property will cause [IEntity.save\(\)](#) method to fail.

**Returns:**

true if the property is read-only; otherwise false.

---

(continued from last page)

---

## isVersionable

```
public boolean isVersionable()
```

Gets a value indicating whether the property is versionable.

**Returns:**

true if the property is versionable; otherwise false.

---

## isInherited

```
public boolean isInherited()
```

Gets a value indicating whether the property value is inherited.

**Returns:**

true if the property is inherited; otherwise false.

---

## isAppendOnly

```
public boolean isAppendOnly()
```

Gets a value indicating whether the property value is append-only.

**Returns:**

true if the property is append-only; otherwise false.

---

## isIncludedInAIP

```
public boolean isIncludedInAIP()
```

Gets a value indicating whether the property value is included in Archive Informational Package (AIP).

**Returns:**

true if the property is included in AIP; otherwise false.

---

## isSearchable

```
public boolean isSearchable()
```

Gets a value indicating whether the property value is indexed.

Indexed values can be included in entity searches.

**Returns:**

true if the property is indexed; otherwise false.

---

## isUnique

```
public boolean isUnique()
```

Gets a value indicating whether the property is unique.

Unique properties must be assigned a property-wide unique value or the [IEntity.save\(\)](#) method will fail. In case of multi-value property all values must be unique.

**Returns:**

true if the property is unique; otherwise false.

---



## isPickList

```
public boolean isPickList()
```

Gets a value indicating whether the property values are predefined and can be chosen from a list of values.

Custom values cannot be assigned to pick-list properties. Such attempts will fail when the client attempts to commit such changes with the [IEntity.save\(\)](#) method.

**Returns:**

true if the property can be picked from a list of values; otherwise false.

---

## areValuesInherited

```
public boolean areValuesInherited()
```

Gets the value indicating whether or not the property values are inherited from hierarchy.

**Returns:**

true if the property values are inherited from hierarchy; otherwise false (defined explicitly).

---

## getPickListValues

```
public com.imis.util.IReadOnlyList getPickListValues()  
    throws StorageConnectorException
```

Gets the read-only list of pick-list property values.

**Returns:**

The read-only list of IPickListValue objects representing pick-list property values.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not a pick-list property.
- 

## getEffectiveRights

```
public IReadOnlyPropertyRights getEffectiveRights()
```

Gets the users effective rights on the property if the property has ACL defined.

**Returns:**

The read-only set of rights describing users effective rights on the property or a null reference if none available.

---

## valueCount

```
public int valueCount()
```

Gets the number of values in the property value container.

**Returns:**

The number of values in the property value container.

---

(continued from last page)

## getValue

```
public String getValue()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the property value as its equivalent string representation.

**Note:** For template properties this method returns non-null result for pick-list properties only.

**Note:** Communication with archive is needed when property type is STRINGMAX or BINARY.

### Returns:

A string representation of the property value or a null reference if no value is available.

### Throws:

UnsupportedOperationException - if unsupported property type.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.

IOException - if an I/O error occurs.

---

## getValue

```
public String getValue(java.text.Format format)  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the property value as its equivalent string representation using the specified format.

**Note:** For template properties this method returns non-null result for pick-list properties only.

**Note:** Communication with archive is needed when property type is STRINGMAX or BINARY.

### Parameters:

format - an object that supplies locale-sensitive formatting information.

### Returns:

A string representation of the property value as specified by format or a null reference if no value is available.

### Throws:

UnsupportedOperationException - if unsupported property type.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.

IOException - if an I/O error occurs.

## getXmlValue

```
public String getXmlValue()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the property value as its equivalent XML string representation.

**Note:** For template properties this method returns non-null result for pick-list properties only.

**Note:** Communication with archive is needed when property type is `STRINGMAX` or `BINARY`.

**Returns:**

An XML string representation of the property value or a `null` reference if no value is available.

**Throws:**

`UnsupportedOperationException` - if unsupported property type.

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.

`IOException` - if an I/O error occurs.

---

## getValue

```
public Object getValue(Class type)  
    throws StorageConnectorException
```

Gets the property value.

This method provides read-only access to the property value. The discrete value type can be determined using the returned value of the [getType\(\)](#) method. If the requested generic type is not compatible with the property type an exception is thrown.

**Note:** For template properties the only allowed discrete value type is `String` and this method returns non-null result for pick-list properties only.

**Parameters:**

`type` - the type that interfaces with [PropertyType](#).

**Returns:**

The property value of appropriate type or a `null` reference if no value is available.

**Throws:**

`UnsupportedOperationException` - if unsupported property type.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Generic argument type is not compatible with property type.
- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.

## getValues

```
public com.imis.util.IReadOnlyList getValues()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the read-only list of property values in its equivalent string representation.

**Note:** For template properties this method returns non-empty result for pick-list properties only.

**Note:** Communication with archive is needed when property type is STRINGMAX or BINARY.

### Returns:

The read-only list of property values in its string representation.

### Throws:

UnsupportedOperationException - if unsupported property type.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.

IOException - if an I/O error occurs.

---

## getValues

```
public com.imis.util.IReadOnlyList getValues(java.text.Format format)  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the read-only list of property values in its equivalent string representation using the specified format.

**Note:** For template properties this method returns non-empty result for pick-list properties only.

**Note:** Communication with archive is needed when property type is STRINGMAX or BINARY.

### Parameters:

`format` - an object that supplies locale-sensitive formatting information.

### Returns:

The read-only list of property values in its string representation as specified by `format`.

### Throws:

UnsupportedOperationException - if unsupported property type.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
  - Pick-list values are not supported for property type.
  - Error occurred while verifying property value.
  - Property value is not allowed based on the current template settings.
-

(continued from last page)

IOException - if an I/O error occurs.

---

## getXmlValues

```
public com.imis.util.IReadOnlyList getXmlValues()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the read-only list of property values in its equivalent XML string representation.

**Note:** For template properties this method returns non-empty result for pick-list properties only.

**Note:** Communication with archive is needed when property type is STRINGMAX or BINARY.

### Returns:

The read-only list of property values in its XML string representation.

### Throws:

UnsupportedOperationException - if unsupported property type.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.

IOException - if an I/O error occurs.

---

## getValues

```
public com.imis.util.IReadOnlyList getValues(Class type)  
    throws StorageConnectorException
```

Gets the read-only list of property values.

This method provides read-only access to the property values. The discrete values type can be determined using the returned value of the [getType\(\)](#) method. If the requested generic type is not compatible with the property type an exception is thrown.

**Note:** For template properties the only allowed discrete value type is `String` and this method returns non-empty result for pick-list properties only.

### Parameters:

type - the type that interfaces with [PropertyType](#).

### Returns:

The read-only list of property values of appropriate type.

### Throws:

UnsupportedOperationException - if unsupported property type.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Generic argument type is not compatible with property type.
- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.

## com.imis.storageconnector.entitymodel Interface IReadOnlyPropertyRights

All Subinterfaces:

[IPropertyRights](#)

public interface **IReadOnlyPropertyRights**  
extends

Allows the caller to access the set of rights on a property for a certain subject in a read-only fashion.

The following rights are defined:

Right	Description
CreateAccess	Subject has the right to add values to an empty property metadata attribute.
ReadAccess	Subject has read access to property metadata attribute.
WriteAccess	Subject has write access to property metadata attribute.
DeleteAccess	Subject has the right to empty out the values of a property metadata attribute.

See Also:

[IPropertyACLEntry](#), [IPropertyRights](#)

### Method Summary

boolean	<a href="#">createAccess</a> ( ) Gets the value indicating whether the subject has the right to add new values of the property metadata attribute.
boolean	<a href="#">deleteAccess</a> ( ) Gets the value indicating whether the subject has the right to empty out the values of the property metadata attribute.
<a href="#">IPropertyACLEntry</a>	<a href="#">getOwner</a> ( ) Gets the owner IPropertyACLEntry.
boolean	<a href="#">isPropagated</a> ( ) Gets the value indicating whether or not the entry is propagated.
boolean	<a href="#">readAccess</a> ( ) Gets the value indicating whether the subject has the right to access (read) values of the property metadata attribute.
Calendar	<a href="#">validFrom</a> ( ) Gets the Calendar after which this entry is valid when calculating the set of rights for the owning entity.
Calendar	<a href="#">validTo</a> ( ) Gets the Calendar before which this entry is valid when calculating the set of rights for the owning entity.
boolean	<a href="#">writeAccess</a> ( ) Gets the value indicating whether the subject has the right to modify the values of the property metadata attribute.

## Methods

### getOwner

```
public IPropertyACLEntry getOwner()
```

Gets the owner `IPropertyACLEntry`.

The return value never changes after property rights is instantiated.

**Returns:**

The owner `IPropertyACLEntry` or a null reference if this is a read-only property rights.

---

### createAccess

```
public boolean createAccess()
```

Gets the value indicating whether the subject has the right to add new values of the property metadata attribute.

**Returns:**

true if the `CreateAccess` right is granted; otherwise false.

**See Also:**

[IPropertyACLEntry.getSubject\(\)](#)

[IPropertyACLEntry.getPropertyName\(\)](#)

---

### readAccess

```
public boolean readAccess()
```

Gets the value indicating whether the subject has the right to access (read) values of the property metadata attribute.

**Returns:**

true if the `ReadAccess` right is granted; otherwise false.

**See Also:**

[IPropertyACLEntry.getSubject\(\)](#)

[IPropertyACLEntry.getPropertyName\(\)](#)

---

### writeAccess

```
public boolean writeAccess()
```

Gets the value indicating whether the subject has the right to modify the values of the property metadata attribute.

**Returns:**

true if the `WriteAccess` right is granted; otherwise false.

**See Also:**

[IPropertyACLEntry.getSubject\(\)](#)

[IPropertyACLEntry.getPropertyName\(\)](#)

---

### deleteAccess

```
public boolean deleteAccess()
```



(continued from last page)

Gets the value indicating whether the subject has the right to empty out the values of the property metadata attribute.

**Returns:**

true if the `DeleteAccess` right is granted; otherwise false.

**See Also:**

[`IPropertyACLEntry.getSubject\(\)`](#)

[`IPropertyACLEntry.getPropertyName\(\)`](#)

---

## **isPropagated**

```
public boolean isPropagated()
```

Gets the value indicating whether or not the entry is propagated.

**Returns:**

true if the entry is propagated; otherwise false.

---

## **validFrom**

```
public Calendar validFrom()
```

Gets the `Calendar` after which this entry is valid when calculating the set of rights for the owning entity.

**Returns:**

A `Calendar` value that represents the entry validity start timestamp or `null` if there is no lower limit.

---

## **validTo**

```
public Calendar validTo()
```

Gets the `Calendar` before which this entry is valid when calculating the set of rights for the owning entity.

**Returns:**

A `Calendar` value that represents the entry validity end timestamp or `null` if there is no upper limit.

---

## com.imis.storageconnector.entitymodel Interface IReadOnlyRetentionPolicyContext

All Subinterfaces:

[IRetentionPolicyContext](#)

public interface **IReadOnlyRetentionPolicyContext**  
extends

Allows the caller to access the retention policy context in a read-only fashion.

### Method Summary

Set	<a href="#">getFilter()</a> Gets the retention policy filter.
<a href="#">RetentionPolicyScope</a>	<a href="#">getScope()</a> Gets the retention policy scope.
boolean	<a href="#">isEffective()</a> Gets a value indicating whether or not the retention entry context is effective for the owning entity.

### Methods

#### isEffective

public boolean **isEffective()**

Gets a value indicating whether or not the retention entry context is effective for the owning entity.

**Returns:**

true if the retention entry context is effective for the owning entity; otherwise false.

#### getScope

public [RetentionPolicyScope](#) **getScope()**  
throws [StorageConnectorException](#)

Gets the retention policy scope.

**Returns:**

A RetentionPolicyScope value.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while getting retention policy entry scope value.

(continued from last page)

## getFilter

```
public Set getFilter()  
    throws StorageConnectorException
```

Gets the retention policy filter.

### Returns:

A `RetentionPolicyFilter` value.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while getting retention policy entry filter value.

## com.imis.storageconnector.entitymodel Interface IRetention

public interface **IRetention**  
extends com.imis.IAutoCloseable

Represents a Retention object, a collection of retention policy and disposition hold entries attached to an IEntity.

See Also:

[IEntity](#)

### Method Summary

<a href="#">IDispositionHoldEntry</a>	<a href="#">addDispositionHoldEntry</a> ( <a href="#">IDispositionHold</a> hold) Adds a new IDispositionHoldEntry specified with the identifier of the IDispositionHoldDefinition.
<a href="#">IRetentionPolicyEntry</a>	<a href="#">addPolicyEntry</a> ( <a href="#">IRetentionPolicy</a> policy) Adds a new IRetentionPolicyEntry for the specified retention policy.
<a href="#">IRetentionPolicyEntry</a>	<a href="#">addPolicyEntry</a> ( <a href="#">IRetentionPolicy</a> policy, <a href="#">RetentionPolicyScope</a> scope, Set filter) Adds a new IRetentionPolicyEntry specified with the identifier of the IRetentionPolicyDefinition.
void	<a href="#">close</a> () Closes the retention and releases all its resources.
com.imis.util.IReadOnlyMap	<a href="#">getDispositionHoldEntries</a> () Gets the read-only map of IDispositionHoldEntry objects assigned to the owner IEntity.
<a href="#">AccessMode</a>	<a href="#">getMode</a> () Gets the retention access mode.
com.imis.util.IReadOnlyMap	<a href="#">getPolicyEntries</a> () Gets the read-only map of IRetentionPolicyEntry objects assigned to the owner IEntity.
void	<a href="#">save</a> () Saves the changes and reopens retention in read-only mode.
void	<a href="#">save</a> (boolean close) Saves the changes and optionally closes retention or reopens retention in read-only mode.

### Methods inherited from interface com.imis.IAutoCloseable

[close](#)

## Methods

### getMode

```
public AccessMode getMode()  
    throws StorageConnectorException
```

(continued from last page)

Gets the retention access mode.

**Returns:**

The retention access mode.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed retention is not allowed.

---

## getPolicyEntries

```
public com.imis.util.IReadOnlyMap getPolicyEntries()  
    throws StorageConnectorException
```

Gets the read-only map of IRetentionPolicyEntry objects assigned to the owner IEntity.

**Returns:**

A read-only map of IRetentionPolicyEntry objects sorted by IRetentionPolicy unique identifiers.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed retention is not allowed.
- Error occurred while reading retention entries.

---

## getDispositionHoldEntries

```
public com.imis.util.IReadOnlyMap getDispositionHoldEntries()  
    throws StorageConnectorException
```

Gets the read-only map of IDispositionHoldEntry objects assigned to the owner IEntity.

**Returns:**

A read-only map of IDispositionHoldEntry objects assigned to the owner IEntity.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed retention is not allowed.
- Error occurred while reading retention entries.

---

## addPolicyEntry

```
public IRetentionPolicyEntry addPolicyEntry(IRetentionPolicy policy)  
    throws StorageConnectorException
```

Adds a new IRetentionPolicyEntry for the specified retention policy.

**Parameters:**

policy - the retention policy.

**Returns:**

An IRetentionPolicyEntry instance.

(continued from last page)

**Throws:**

[NullPointerException](#) - if policy is a null reference.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed retention is not allowed.
- Error occurred while reading retention entries.
- Retention policy entry already exists.
- Error occurred while adding retention entry.

---

## addPolicyEntry

```
public IRetentionPolicyEntry addPolicyEntry(IRetentionPolicy policy,  
      RetentionPolicyScope scope,  
      Set filter)  
throws StorageConnectorException
```

Adds a new [IRetentionPolicyEntry](#) specified with the identifier of the [IRetentionPolicyDefinition](#).

**Parameters:**

policy - the retention policy.

scope - the retention policy scope.

filter - the retention policy filter.

**Returns:**

An [IRetentionPolicyEntry](#) instance.

**Throws:**

[NullPointerException](#) - if policy is a null reference.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed retention is not allowed.
- Error occurred while reading retention entries.
- Retention policy entry already exists.
- Error occurred while adding retention entry.

---

## addDispositionHoldEntry

```
public IDispositionHoldEntry addDispositionHoldEntry(IDispositionHold hold)  
throws StorageConnectorException
```

Adds a new [IDispositionHoldEntry](#) specified with the identifier of the [IDispositionHoldDefinition](#).

**Parameters:**

hold - the disposition hold.

**Returns:**

An [IDispositionHoldEntry](#) instance.

**Throws:**

[NullPointerException](#) - if hold is a null reference.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed retention is not allowed.
- Error occurred while reading retention entries.
- Disposition hold entry already exists.
- Error occurred while adding retention entry.

---

## save

```
public void save()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Saves the changes and reopens retention in read-only mode.

**Note:** This operation allows user to pass a reason for saving retention. Reason can be passed through [AuditLog](#) message.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed retention is not allowed.
- Error occurred while closing entity retention.

IOException - if an I/O error occurs.

---

## save

```
public void save(boolean close)  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Saves the changes and optionally closes retention or reopens retention in read-only mode.

**Note:** This operation allows user to pass a reason for saving retention. Reason can be passed through [AuditLog](#) message.

### Parameters:

close - true to close the retention after save; otherwise false.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed retention is not allowed.
- Error occurred while closing entity retention.

IOException - if an I/O error occurs.

---

## close

```
public void close()  
    throws StorageConnectorException,  
           IOException
```

(continued from last page)

Closes the retention and releases all its resources.

**Note:** This operation allows user to pass a reason for closing retention. Reason can be passed through [AuditLog](#) message.

**Throws:**

[StorageConnectorException](#) - if

- Error occurred while closing entity retention.

`IOException` - if an I/O error occurs.



## com.imis.storageconnector.entitymodel Interface IRetentionPolicy

public interface **IRetentionPolicy**  
extends

Represents a retention policy.

### Method Summary

Calendar	<a href="#"><u>accessed()</u></a> Gets the retention policy access time-stamp.
Calendar	<a href="#"><u>created()</u></a> Gets the retention policy creation time-stamp.
String	<a href="#"><u>getAction()</u></a> Gets the retention policy action.
<a href="#"><u>IDirectoryEntity</u></a>	<a href="#"><u>getCreator()</u></a> Gets the retention policy creator.
String	<a href="#"><u>getDescription()</u></a> Gets the retention policy description.
String	<a href="#"><u>getDetailedDescription()</u></a> Gets the retention policy detailed description.
String	<a href="#"><u>getId()</u></a> Gets the retention policy internal identifier.
<a href="#"><u>IReadOnlyContent</u></a>	<a href="#"><u>getMandate()</u></a> Gets the retention policy mandate.
String	<a href="#"><u>getName()</u></a> Gets the retention policy name.
String	<a href="#"><u>getReason()</u></a> Gets the retention policy reason.
String	<a href="#"><u>getTrigger()</u></a> Gets the retention policy trigger.
Calendar	<a href="#"><u>modified()</u></a> Gets the retention policy modification time-stamp.

### Methods

#### getId

public String **getId()**

Gets the retention policy internal identifier.

(continued from last page)

**Returns:**

A string that represents the retention policy internal identifier.

---

## getName

```
public String getName()  
    throws StorageConnectorException
```

Gets the retention policy name.

**Returns:**

A string that represents the retention policy name.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getDescription

```
public String getDescription()  
    throws StorageConnectorException
```

Gets the retention policy description.

**Returns:**

A string that represents the retention policy description.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getCreator

```
public IDirectoryEntity getCreator()  
    throws StorageConnectorException
```

Gets the retention policy creator.

**Returns:**

A [IDirectoryEntity](#) that represents the creator of the retention policy.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.

---

## created

```
public Calendar created()
```

(continued from last page)

Gets the retention policy creation time-stamp.

**Returns:**

The date and time when the retention policy was created.

---

## modified

```
public Calendar modified()
```

Gets the retention policy modification time-stamp.

**Returns:**

The date and time when the retention policy was modified.

---

## accessed

```
public Calendar accessed()
```

Gets the retention policy access time-stamp.

**Returns:**

The date and time when the retention policy was accessed.

---

## getTrigger

```
public String getTrigger()  
    throws StorageConnectorException
```

Gets the retention policy trigger.

**Returns:**

A string that represents the retention policy trigger.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- 

## getAction

```
public String getAction()  
    throws StorageConnectorException
```

Gets the retention policy action.

**Returns:**

A string that represents the retention policy action.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
-

(continued from last page)

## getReason

```
public String getReason()  
    throws StorageConnectorException
```

Gets the retention policy reason.

**Returns:**

A string that represents the retention policy reason.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getDetailedDescription

```
public String getDetailedDescription()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the retention policy detailed description.

**Returns:**

A string that represents the retention policy detailed description.

**Throws:**

[CancellationException](#) - if an operation was canceled by the user.

[LoginException](#) - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Error occurred while opening string value data stream.

[IOException](#) - if an I/O error occurs.

---

## getMandate

```
public IReadOnlyContent getMandate()  
    throws StorageConnectorException
```

Gets the retention policy mandate.

**Returns:**

A [IReadOnlyContent](#) value that represents the retention policy mandate.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

## com.imis.storageconnector.entitymodel Interface IRetentionPolicyContext

All Superinterfaces:

[IReadOnlyRetentionPolicyContext](#)

public interface **IRetentionPolicyContext**

extends [IReadOnlyRetentionPolicyContext](#)

Allows the caller to access the retention policy context in a read-write fashion.

### Method Summary

<a href="#">IRetentionPolicyEntry</a>	<a href="#">getOwner()</a> Gets the retention policy context owner.
void	<a href="#">setFilter(Set value)</a> Sets the retention policy filter.
void	<a href="#">setScope(RetentionPolicyScope value)</a> Sets the retention policy scope.

#### Methods inherited from interface

[com.imis.storageconnector.entitymodel.IReadOnlyRetentionPolicyContext](#)

[getFilter](#), [getScope](#), [isEffective](#)

### Methods

#### getOwner

public [IRetentionPolicyEntry](#) **getOwner()**

Gets the retention policy context owner.

**Returns:**

The retention policy context owner.

#### setScope

public void **setScope**([RetentionPolicyScope](#) value)  
throws [StorageConnectorException](#)

Sets the retention policy scope.

**Parameters:**

value - a RetentionPolicyScope value.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while setting retention policy entry scope value.

---

## setFilter

```
public void setFilter(Set value)
    throws StorageConnectorException
```

Sets the retention policy filter.

### Parameters:

value - a RetentionPolicyFilter value.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while setting retention policy entry scope value.

## com.imis.storageconnector.entitymodel Interface IRetentionPolicyEntry

public interface **IRetentionPolicyEntry**  
extends

Represents a retention policy entry.

### Method Summary

void	<a href="#"><code>delete()</code></a> Deletes the explicit retention policy entry.
<a href="#"><code>IRetentionPolicy</code></a>	<a href="#"><code>getDefinition()</code></a> Gets the retention policy definition.
<a href="#"><code>IRetentionPolicyContext</code></a>	<a href="#"><code>getExplicitContext()</code></a> Gets the explicit retention policy context.
<code>com.imis.util.IReadOnlyList</code>	<a href="#"><code>getInheritedContext()</code></a> Gets the read-only list of inherited retention policy contexts.

### Methods

#### getDefinition

public [`IRetentionPolicy`](#) **getDefinition()**  
throws [`StorageConnectorException`](#)

Gets the retention policy definition.

**Returns:**

An [`IRetentionPolicy`](#) object.

**Throws:**

[`StorageConnectorException`](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.
- Error occurred trying to access `Policies` class.
- Error occurred while getting entity child entities on archive.

#### getInheritedContext

public `com.imis.util.IReadOnlyList` **getInheritedContext()**

Gets the read-only list of inherited retention policy contexts.

**Returns:**

(continued from last page)

The read-only list of `IRetentionPolicyContext` objects.

---

## getExplicitContext

```
public IRetentionPolicyContext getExplicitContext()
```

Gets the explicit retention policy context.

**Returns:**

The `IRetentionPolicyContext` object.

---

## delete

```
public void delete()  
    throws StorageConnectorException
```

Deletes the explicit retention policy entry.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed retention is not allowed.
- Error occurred while removing retention policy entry.
- Error occurred while reading retention entries.



## com.imis.storageconnector.entitymodel Interface IRetentionPolicySnapshot

public interface **IRetentionPolicySnapshot**  
extends

Represents a retention policy snapshot.

### Method Summary

Calendar	<a href="#"><code>created()</code></a> Gets the retention policy creation time-stamp.
String	<a href="#"><code>getAction()</code></a> Gets the retention policy action.
<a href="#"><code>IDirectoryEntity</code></a>	<a href="#"><code>getCreator()</code></a> Gets the retention policy creator.
String	<a href="#"><code>getDescription()</code></a> Gets the retention policy description.
String	<a href="#"><code>getDetailedDescription()</code></a> Gets the retention policy detailed description.
<a href="#"><code>IReadOnlyContent</code></a>	<a href="#"><code>getMandate()</code></a> Gets the retention policy mandate.
String	<a href="#"><code>getName()</code></a> Gets the retention policy name.
<a href="#"><code>IRetentionPolicy</code></a>	<a href="#"><code>getOriginal()</code></a> Gets the original retention policy.
String	<a href="#"><code>getReason()</code></a> Gets the retention policy reason.
String	<a href="#"><code>getTrigger()</code></a> Gets the retention policy trigger.

### Methods

#### **getName**

public String **getName()**  
throws [`StorageConnectorException`](#)

Gets the retention policy name.

#### **Returns:**

A string that represents the retention policy name.

#### **Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getDescription

```
public String getDescription()  
    throws StorageConnectorException
```

Gets the retention policy description.

**Returns:**

A string that represents the retention policy description.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getCreator

```
public IDirectoryEntity getCreator()  
    throws StorageConnectorException
```

Gets the retention policy creator.

**Returns:**

A [IDirectoryEntity](#) that represents the creator of the retention policy.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## created

```
public Calendar created()
```

Gets the retention policy creation time-stamp.

**Returns:**

The date and time when the retention policy was created.

---

## getTrigger

```
public String getTrigger()  
    throws StorageConnectorException
```

Gets the retention policy trigger.

**Returns:**

A string that represents the retention policy trigger.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getAction

```
public String getAction()  
    throws StorageConnectorException
```

Gets the retention policy action.

### Returns:

A string that represents the retention policy action.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getReason

```
public String getReason()  
    throws StorageConnectorException
```

Gets the retention policy reason.

### Returns:

A string that represents the retention policy reason.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getDetailedDescription

```
public String getDetailedDescription()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the retention policy detailed description.

### Returns:

A string that represents the retention policy detailed description.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Error occurred while opening string value data stream.

(continued from last page)

`IOException` - if an I/O error occurs.

---

## getMandate

```
public IReadOnlyContent getMandate()  
    throws StorageConnectorException
```

Gets the retention policy mandate.

### Returns:

A `IReadOnlyContent` value that represents the retention policy mandate.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getOriginal

```
public IRetentionPolicy getOriginal()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the original retention policy.

### Returns:

An `IRetentionPolicy` that represents the original retention policy.

### Throws:

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting entities info on archive.

`IOException` - if an I/O error occurs.

## com.imis.storageconnector.entitymodel Interface IReview

public interface **IReview**  
extends com.imis.IAutoCloseable

Represents a review folder.

### Method Summary

Calendar	<a href="#"><u>accessed()</u></a> Gets the review last access time-stamp.
boolean	<a href="#"><u>areValuesInherited(SystemProperty systemProperty)</u></a> Gets the value indicating whether the specified system property values are inherited.
void	<a href="#"><u>close()</u></a> Closes the review and releases all its resources.
Calendar	<a href="#"><u>closed()</u></a> Gets the review status closed time-stamp.
Calendar	<a href="#"><u>created()</u></a> Gets the review creation time-stamp.
<a href="#"><u>IDocument</u></a>	<a href="#"><u>createDocument(String templateId)</u></a> Creates a review document using the specified template identifier.
<a href="#"><u>IACL</u></a>	<a href="#"><u>getACL()</u></a> Gets the review IACL.
String	<a href="#"><u>getAction()</u></a> Gets the review action.
InputStream	<a href="#"><u>getArchivalInformationalPackage()</u></a> Gets the review archival informational package encoded in base 64 format.
<a href="#"><u>IArchive</u></a>	<a href="#"><u>getArchive()</u></a> Gets the IArchive where the review is stored.
<a href="#"><u>AuditLog</u></a>	<a href="#"><u>getAuditLog()</u></a> Gets the audit log object from the owning IArchive.
String	<a href="#"><u>getClassificationCode()</u></a> Gets the review classification code in canonical format.
String	<a href="#"><u>getComments()</u></a> Gets the review comments.
<a href="#"><u>IDirectoryEntity</u></a>	<a href="#"><u>getCreator()</u></a> Gets the review creator.
String	<a href="#"><u>getDescription()</u></a> Sets the review description.

com.imis.util.IReadOnlyList	<a href="#">getDocuments()</a> Gets the review list of documents.
<a href="#">IReadOnlyEntityRights</a>	<a href="#">getEffectiveRights()</a> Gets the users effective rights on the review.
com.imis.util.IReadOnlyList	<a href="#">getEvidenceRecords()</a> Gets the review read-only list of evidence records encoded in base 64 format.
String	<a href="#">getId()</a> Gets the review unique identifier.
com.imis.util.IReadOnlyList	<a href="#">getKeywords()</a> Gets the review list of keywords.
com.imis.util.IReadOnlyList	<a href="#">getMembers()</a> Gets the list of members of the review.
String	<a href="#">getMessage()</a> Gets the review message.
<a href="#">AccessMode</a>	<a href="#">getMode()</a> Gets the review access mode.
<a href="#">IDirectoryEntity</a>	<a href="#">getModifiedBy()</a> Gets the last modifier directory entity.
<a href="#">IDirectoryEntity</a>	<a href="#">getOwner()</a> Gets the review owner IDirectoryEntity.
com.imis.util.IReadOnlyList	<a href="#">getPickListValues(SystemProperty systemProperty)</a> Gets the read-only list of the pick-list values for the specified system property.
String	<a href="#">getPublicClassificationCode()</a> Gets the review classification code in public format.
String	<a href="#">getQueryExpression()</a> Gets the query expression.
InputStream	<a href="#">getReport(ReportType type)</a> Gets the specified report data stream.
com.imis.util.IReadOnlyList	<a href="#">getRetentionPolicies()</a> Gets the review list of retention policies.
<a href="#">IContent</a>	<a href="#">getReviewedItems()</a> Gets the review reviewed items.
com.imis.util.IReadOnlyList	<a href="#">getSaveLog()</a> Gets the read-only list of strings containing commit log reports for IReview in XML format encoded per XML processing instruction.
<a href="#">IReadOnlyContent</a>	<a href="#">getScheduledItems()</a> Gets the review scheduled items.
<a href="#">IEntityStub</a>	<a href="#">getScope()</a> Gets the review scope entity.
String	<a href="#">getState()</a> Gets the review state.

String	<a href="#"><u>getStatus()</u></a> Sets the review status.
<a href="#"><u>IReviewStub</u></a>	<a href="#"><u>getStub()</u></a> Gets the describing IReview.
com.imis.util.IReadOnlyList	<a href="#"><u>getSubEntityTemplates()</u></a> Gets the read-only list of templates that represent the review sub-entity creation templates.
com.imis.util.IReadOnlyMap	<a href="#"><u>getSystemProperties()</u></a> Gets the read-only map of read-only system properties sorted by SystemProperty enum values.
<a href="#"><u>IReadOnlyProperty</u></a>	<a href="#"><u>getSystemProperty(SystemProperty systemProperty)</u></a> Gets the read-only property for the specified system property.
<a href="#"><u>ITemplate</u></a>	<a href="#"><u>getTemplate()</u></a> Gets the review template identifier.
String	<a href="#"><u>getTitle()</u></a> Gets the review title.
boolean	<a href="#"><u>hasAIP()</u></a> Gets a value indicating whether or not the review has AIP.
boolean	<a href="#"><u>isClosed()</u></a> Gets a value indicating whether the review is closed.
Calendar	<a href="#"><u>modified()</u></a> Gets the review last modification time-stamp.
Calendar	<a href="#"><u>opened()</u></a> Gets the review status opened time-stamp.
void	<a href="#"><u>save()</u></a> Saves the changes and reopens the review in read-only mode.
void	<a href="#"><u>save(boolean close)</u></a> Saves the changes and optionally closes the review or reopens the review in read-only mode.
void	<a href="#"><u>setAction(String value)</u></a> Sets the review action.
void	<a href="#"><u>setComments(String value)</u></a> Sets the review comments.
void	<a href="#"><u>setDescription(String value)</u></a> Sets the review description.
void	<a href="#"><u>setKeywords(List value)</u></a> Sets the review list of keywords.
void	<a href="#"><u>setMembers(List value)</u></a> Sets the list of members of the review.
void	<a href="#"><u>setOwner(IDirectoryEntity value)</u></a> Sets the review owner IDirectoryEntity.

void	<a href="#">setStatus</a> (String value) Sets the review status.
void	<a href="#">setTitle</a> (String value) Sets the review title.

**Methods inherited from interface** `com.imis.IAutoCloseable`

close

## Methods

### getStub

```
public IReviewStub getStub()
```

Gets the describing IReview.

**Returns:**

The IReviewStub instance.

### getId

```
public String getId()  
throws StorageConnectorException
```

Gets the review unique identifier.

**Returns:**

A string that represents the review identifier.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Operation on unsaved new review is not allowed.
- Error occurred while getting review identifier.

### getClassificationCode

```
public String getClassificationCode()  
throws StorageConnectorException
```

Gets the review classification code in canonical format.

**Returns:**

A string that represents the review classification code.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting classification code.



## getPublicClassificationCode

```
public String getPublicClassificationCode()  
    throws StorageConnectorException
```

Gets the review classification code in public format.

The classification code in canonical format is converted to public classification code by replacing the entity type prefix and classification hierarchy delimiter with the respective entity type hierarchy delimiters.

**Returns:**

A string that represents the review public classification code.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting public classification code

---

## getTitle

```
public String getTitle()  
    throws StorageConnectorException
```

Gets the review title.

**Returns:**

The review title.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setTitle

```
public void setTitle(String value)  
    throws StorageConnectorException
```

Sets the review title.

**Parameters:**

value - the review title.

**Throws:**

[NullPointerException](#) - if value is a null reference.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
  - Operation on review opened in read-only mode is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
  - Error occurred while setting property values.
- 

## getDescription

```
public String getDescription()  
    throws StorageConnectorException
```

Sets the review description.

**Returns:**

The review description.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## setDescription

```
public void setDescription(String value)  
    throws StorageConnectorException
```

Sets the review description.

**Parameters:**

value - the review description.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
  - Operation on review opened in read-only mode is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
  - Error occurred while setting property values.
- 

## getStatus

```
public String getStatus()  
    throws StorageConnectorException
```

Sets the review status.

**Returns:**

The review status.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setStatus

```
public void setStatus(String value)  
    throws StorageConnectorException
```

Sets the review status.

**Note:** Use this method on new unsaved reviews only.

### Parameters:

value - the review status or a null reference to clear the review status to inherited value.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Operation on saved review is not allowed.
- Property is not present in template properties.
- Property value is not allowed based on the current template settings.
- Error occurred while setting property values.

---

## getMode

```
public AccessMode getMode()  
    throws StorageConnectorException
```

Gets the review access mode.

### Returns:

The review access mode.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.

---

## getState

```
public String getState()  
    throws StorageConnectorException
```

Gets the review state.

### Returns:

The review state.

### Throws:

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## getMessage

```
public String getMessage()  
    throws StorageConnectorException
```

Gets the review message.

### Returns:

The review message.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## getAction

```
public String getAction()  
    throws StorageConnectorException
```

Gets the review action.

### Returns:

The review action.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## setAction

```
public void setAction(String value)  
    throws StorageConnectorException
```

Sets the review action.

### Parameters:

value - the review action.

### Throws:

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Operation on review opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Property value is not allowed based on the current template settings.
- Error occurred while setting property values.

---

## getQueryExpression

```
public String getQueryExpression()  
    throws StorageConnectorException
```

Gets the query expression.

**Returns:**

The query expression.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.

---

## getScope

```
public IEntityStub getScope()  
    throws StorageConnectorException
```

Gets the review scope entity.

**Returns:**

The review scope entity.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.

---

## getMembers

```
public com.imis.util.IReadOnlyList getMembers()  
    throws StorageConnectorException
```

Gets the list of members of the review.

**Returns:**

The list of members of the review.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## setMembers

```
public void setMembers(List value)  
    throws StorageConnectorException
```

Sets the list of members of the review.

### Parameters:

value - the list of members of the review.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Operation on review opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getComments

```
public String getComments()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the review comments.

### Returns:

The review comments.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

IOException - if an I/O error occurs.

---

## setComments

```
public void setComments(String value)  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Sets the review comments.

(continued from last page)

**Parameters:**

value - the review comments.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Operation on review opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

`IOException` - if an I/O error occurs.

---

**created**

```
public Calendar created()  
    throws StorageConnectorException
```

Gets the review creation time-stamp.

**Returns:**

The date and time when the review was created.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.

---

**accessed**

```
public Calendar accessed()  
    throws StorageConnectorException
```

Gets the review last access time-stamp.

**Returns:**

The date and time when the review was last accessed.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.

---

**modified**

```
public Calendar modified()  
    throws StorageConnectorException
```

Gets the review last modification time-stamp.

**Returns:**

(continued from last page)

The date and time when the review was last modified.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.

---

## getModifiedBy

```
public IDirectoryEntity getModifiedBy()  
    throws StorageConnectorException
```

Gets the last modifier directory entity.

**Returns:**

The last modifier directory entity or a `null` reference, if this instance has no last modifier directory entity.

---

## opened

```
public Calendar opened()  
    throws StorageConnectorException
```

Gets the review status opened time-stamp.

**Returns:**

The date and time when the review status was set to opened or a `null` reference if review has no status.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## closed

```
public Calendar closed()  
    throws StorageConnectorException
```

Gets the review status closed time-stamp.

**Returns:**

The date and time when the review status was set to closed or a `null` reference if review has no status.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## isClosed

```
public boolean isClosed()
```



(continued from last page)

Gets a value indicating whether the review is closed.

**Returns:**

true if the review is closed; otherwise false.

---

## getACL

```
public IACL getACL()  
    throws StorageConnectorException
```

Gets the review IACL.

This method provides access to an IACL instance containing ACL entries which define access rights for different review operations.

**Returns:**

An IACL instance.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting ACL entries.

---

## getEffectiveRights

```
public IReadOnlyEntityRights getEffectiveRights()  
    throws StorageConnectorException
```

Gets the users effective rights on the review.

**Returns:**

The read-only set of rights describing users effective rights on the review.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.

---

## getArchive

```
public IArchive getArchive()
```

Gets the IArchive where the review is stored.

**Returns:**

The IArchive where the review is stored.

---

## getCreator

```
public IDirectoryEntity getCreator()  
    throws StorageConnectorException
```

Gets the review creator.

**Returns:**

(continued from last page)

A `IDirectoryEntity` that represents the creator of the review.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
  - Operation on review opened in read-only mode is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## getOwner

```
public IDirectoryEntity getOwner()  
    throws StorageConnectorException
```

Gets the review owner `IDirectoryEntity`.

**Returns:**

An `IDirectoryEntity` object representing the review owner or a null reference, if this review has no owner.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
  - Operation on review opened in read-only mode is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## setOwner

```
public void setOwner(IDirectoryEntity value)  
    throws StorageConnectorException
```

Sets the review owner `IDirectoryEntity`.

**Parameters:**

value - an `IDirectoryEntity` object representing the review owner or a null reference, if this review has no owner.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
  - Operation on review opened in read-only mode is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
  - Error occurred while setting property values.
- 

## getSaveLog

```
public com.imis.util.IReadOnlyList getSaveLog()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

(continued from last page)

Gets the read-only list of strings containing commit log reports for `IReview` in XML format encoded per XML processing instruction.

**Returns:**

A read-only list of strings containing commit log reports in XML format encoded per XML processing instruction.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

`IOException` - if an I/O error occurs.

---

## getTemplate

```
public ITemplate getTemplate()  
    throws StorageConnectorException
```

Gets the review template identifier.

**Returns:**

An `ITemplate` that represents the review template.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.

---

## getAuditLog

```
public AuditLog getAuditLog()  
    throws StorageConnectorException
```

Gets the audit log object from the owning `IArchive`.

**Returns:**

The `AuditLog` instance.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.

**See Also:**

[AuditLog](#)

---

## getArchivalInformationalPackage

```
public InputStream getArchivalInformationalPackage()  
    throws StorageConnectorException,  
        IOException
```

(continued from last page)

Gets the review archival informational package encoded in base 64 format.

**Returns:**

An input stream containing review archival informational package encoded in base 64 format.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting archival informational package.

`IOException` - if an I/O error occurs.

---

## getEvidenceRecords

```
public com.imis.util.IReadOnlyList getEvidenceRecords()  
    throws StorageConnectorException,  
           IOException
```

Gets the review read-only list of evidence records encoded in base 64 format.

**Returns:**

A read-only list of `InputStream` objects containing review evidence records encoded in base 64 format.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting evidence records.

`IOException` - if an I/O error occurs.

---

## getKeywords

```
public com.imis.util.IReadOnlyList getKeywords()  
    throws StorageConnectorException
```

Gets the review list of keywords.

**Returns:**

The review list of keywords.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
  - Error occurred while getting properties.
  - Property is not present in template properties.
- 

## setKeywords

```
public void setKeywords(List value)  
    throws StorageConnectorException
```

Sets the review list of keywords.

**Parameters:**

(continued from last page)

value - the review list of keywords.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Operation on review opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while setting property values.

---

## getSubEntityTemplates

```
public com.imis.util.IReadOnlyList getSubEntityTemplates()  
    throws StorageConnectorException
```

Gets the read-only list of templates that represent the review sub-entity creation templates.

**Returns:**

The read-only list of `ITemplate` objects that the review sub-entity creation templates.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.

---

## getScheduledItems

```
public IReadOnlyContent getScheduledItems()  
    throws StorageConnectorException
```

Gets the review scheduled items.

**Returns:**

An `IReadOnlyContent` object.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getReviewedItems

```
public IContent getReviewedItems()  
    throws StorageConnectorException
```

Gets the review reviewed items.

**Returns:**

An `IContent` object.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getRetentionPolicies

```
public com.imis.util.IReadOnlyList getRetentionPolicies()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the review list of retention policies.

### Returns:

The list of IRetentionPolicySnapshot objects.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Operation on unsaved new review is not allowed.
- Error occurred while getting entity child entities on archive.

IOException - if an I/O error occurs.

---

## getDocuments

```
public com.imis.util.IReadOnlyList getDocuments()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the review list of documents.

### Returns:

The list of IEntityStub objects representing review documents.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Operation on unsaved new review is not allowed.
- Error occurred while getting entity child entities on archive.

IOException - if an I/O error occurs.

---

## getSystemProperties

```
public com.imis.util.IReadOnlyMap getSystemProperties()  
    throws StorageConnectorException
```

(continued from last page)

Gets the read-only map of read-only system properties sorted by `SystemProperty` enum values.

**Returns:**

The read-only map of `IReadOnlyProperty` objects sorted by `SystemProperty` enum values.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting properties.

---

## getSystemProperty

```
public IReadOnlyProperty getSystemProperty(SystemProperty systemProperty)  
    throws StorageConnectorException
```

Gets the read-only property for the specified system property.

**Parameters:**

`systemProperty` - the system property.

**Returns:**

The read-only property for the specified system property.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## areValuesInherited

```
public boolean areValuesInherited(SystemProperty systemProperty)  
    throws StorageConnectorException
```

Gets the value indicating whether the specified system property values are inherited.

**Parameters:**

`systemProperty` - the system property.

**Returns:**

`true` if the specified system property values are inherited; otherwise `false`.

**Throws:**

`IllegalArgumentException` - if invalid system property.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

(continued from last page)

## getPickListValues

```
public com.imis.util.IReadOnlyList getPickListValues(SystemProperty systemProperty)
    throws StorageConnectorException
```

Gets the read-only list of the pick-list values for the specified system property.

### Parameters:

systemProperty - the system property.

### Returns:

The read-only list of IPickListValue objects representing the pick-list values for the specified system property.

### Throws:

[IllegalArgumentException](#) - if invalid system property.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Property is not present in template properties.
- Property is not a pick-list property.

---

## getReport

```
public InputStream getReport(ReportType type)
    throws StorageConnectorException
```

Gets the specified report data stream.

**Note:** Valid report type on IReview is:

- [ReportType.REVIEW\\_SAVE](#)

### Parameters:

type - the report type.

### Returns:

The report data stream.

### Throws:

[UnsupportedOperationException](#) - if report not supported.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Operation on unsaved new review is not allowed.
- Error occurred while getting review report.

---

## createDocument

```
public IDocument createDocument(String templateId)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Creates a review document using the specified template identifier.



(continued from last page)

**Parameters:**`templateId` - the template identifier.**Returns:**An `IDocument` instance.**Throws:**`CancellationException` - if an operation was canceled by the user.`LoginException` - if a login error occurs.[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Operation on unsaved new review is not allowed.
- Invalid template identifier for review document.
- Template identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while creating entity on archive.

`IOException` - if an I/O error occurs.

---

**save**

```
public void save()
    throws StorageConnectorException,
           IOException
```

Saves the changes and reopens the review in read-only mode.

Calling this method is same as calling [save\(boolean\)](#) with `false` parameter.Call to this method commits all changes made to this instance of review after it was opened in write mode or created. [getId\(\)](#) method can be used after completion of this routine to access review unique identifier.**Note:** This operation allows user to pass a reason for saving review. Reason can be passed through [AuditLog](#) message.**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Operation on review opened in read-only mode is not allowed.
- Error occurred while committing review changes on archive.

`IOException` - if an I/O error occurs.

---

**save**

```
public void save(boolean close)
    throws StorageConnectorException,
           IOException
```

Saves the changes and optionally closes the review or reopens the review in read-only mode.

Call to this method commits all changes made to this review instance after it was opened in write mode or created.

Call to this method commits all changes made to this instance of review after it was opened in write mode or created. [getId\(\)](#) method can be used after completion of this routine to access review unique identifier.**Note:** This operation allows user to pass a reason for saving review. Reason can be passed through [AuditLog](#) message.

(continued from last page)

**Parameters:**

close - true to close review after save; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed review is not allowed.
- Operation on review opened in read-only mode is not allowed.
- Error occurred while committing review changes on archive.

IOException - if an I/O error occurs.

---

**close**

```
public void close()  
    throws StorageConnectorException,  
           IOException
```

Closes the review and releases all its resources.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while closing review on archive.

IOException - if an I/O error occurs.

---

**hasAIP**

```
public boolean hasAIP()  
    throws StorageConnectorException
```

Gets a value indicating whether or not the review has AIP.

**Returns:**

true if the review is has AIP; otherwise false.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.

## com.imis.storageconnector.entitymodel Interface IReviewStub

public interface **IReviewStub**  
extends

Defines an IReview descriptor.

An IReviewStub defines an IReview descriptor containing the review folder metadata and enables a caller to open a full-feature IReview object using the `open()` method.

**Note:** Instance is automatically recycled after the last reference is released.

**See Also:**

[IReview](#)

### Method Summary

Calendar	<a href="#">accessed()</a> Gets the describing IReview last access time-stamp.
boolean	<a href="#">areValuesInherited(SystemProperty systemProperty)</a> Gets the value indicating whether the specified system property values are inherited.
Calendar	<a href="#">closed()</a> Gets the describing IReview status closed time-stamp.
Calendar	<a href="#">created()</a> Gets the describing IReview creation time-stamp.
<a href="#">IArchive</a>	<a href="#">getArchive()</a> Gets the IArchive where the describing IReview is stored.
String	<a href="#">getClassificationCode()</a> Gets the describing IReview classification code in canonical format.
<a href="#">IDirectoryEntity</a>	<a href="#">getCreator()</a> Gets the describing IReview creator.
String	<a href="#">getDescription()</a> Gets the describing IReview description.
<a href="#">IReadOnlyEntityRights</a>	<a href="#">getEffectiveRights()</a> Gets the users effective rights on the describing IReview.
String	<a href="#">getId()</a> Gets the describing IReview unique identifier.
com.imis.util.IReadOnlyList	<a href="#">getKeywords()</a> Gets the describing IReview list of keywords.
String	<a href="#">getMessage()</a> Gets the describing IReview message.
<a href="#">IDirectoryEntity</a>	<a href="#">getModifiedBy()</a> Gets the last modifier directory entity.

<a href="#">IDirectoryEntity</a>	<a href="#">getOwner()</a> Gets the describing IReview owner IDirectoryEntity.
com.imis.util.IReadOnlyList	<a href="#">getPickListValues(SystemProperty systemProperty)</a> Gets the read-only list of the pick-list values for the specified system property.
String	<a href="#">getPublicClassificationCode()</a> Gets the describing IReview classification code in public format.
InputStream	<a href="#">getReport(ReportType type)</a> Gets the specified report data stream for the describing IReview.
String	<a href="#">getState()</a> Gets the describing IReview state.
String	<a href="#">getStatus()</a> Gets the describing IReview status.
com.imis.util.IReadOnlyList	<a href="#">getSubEntityTemplates()</a> Gets the read-only list of templates that represent the describing IReview sub-entity creation templates.
com.imis.util.IReadOnlyMap	<a href="#">getSystemProperties()</a> Gets the read-only map of read-only system properties sorted by SystemProperty enum values.
<a href="#">IReadOnlyProperty</a>	<a href="#">getSystemProperty(SystemProperty systemProperty)</a> Gets the read-only property for the specified system property.
<a href="#">ITemplate</a>	<a href="#">getTemplate()</a> Gets the describing IReview template.
String	<a href="#">getTitle()</a> Gets the describing IReview title.
boolean	<a href="#">hasAIP()</a> Gets a value indicating whether or not the describing IReview has AIP.
Calendar	<a href="#">modified()</a> Gets the describing IReview last modification time-stamp.
<a href="#">IReview</a>	<a href="#">open(AccessMode mode)</a> Opens a review described by this IReviewStub using a specified review access mode.
Calendar	<a href="#">opened()</a> Gets the describing IReview status opened time-stamp.
void	<a href="#">setStatus(String status)</a> Sets the status of the describing IReview.

## Methods

### getId

```
public String getId()
```

Gets the describing IReview unique identifier.

---

(continued from last page)

**Returns:**

A string that represents the review identifier.

---

## getClassificationCode

```
public String getClassificationCode()
```

Gets the describing IReview classification code in canonical format.

**Returns:**

A string that represents the review classification code.

---

## getPublicClassificationCode

```
public String getPublicClassificationCode()
```

Gets the describing IReview classification code in public format.

The classification code in canonical format is converted to public classification code by replacing the entity type prefix and classification hierarchy delimiter with the respective entity type hierarchy delimiters.

**Returns:**

A string that represents the review public classification code.

---

## getTitle

```
public String getTitle()  
throws StorageConnectorException
```

Gets the describing IReview title.

**Returns:**

The review title.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- 

## getDescription

```
public String getDescription()  
throws StorageConnectorException
```

Gets the describing IReview description.

**Returns:**

The review description.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
-

(continued from last page)

---

## getStatus

```
public String getStatus()  
    throws StorageConnectorException
```

Gets the describing IReview status.

**Returns:**

The review status.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getState

```
public String getState()  
    throws StorageConnectorException
```

Gets the describing IReview state.

**Returns:**

The review state.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getMessage

```
public String getMessage()  
    throws StorageConnectorException
```

Gets the describing IReview message.

**Returns:**

The review message.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## created

```
public Calendar created()
```

Gets the describing IReview creation time-stamp.

**Returns:**

The date and time when the review was created.

## modified

```
public Calendar modified()
```

Gets the describing IReview last modification time-stamp.

**Returns:**

The date and time when the review was last modified.

---

## getModifiedBy

```
public IDirectoryEntity getModifiedBy()
```

Gets the last modifier directory entity.

**Returns:**

The last modifier directory entity or a null reference, if this instance has no last modifier directory entity.

---

## accessed

```
public Calendar accessed()
```

Gets the describing IReview last access time-stamp.

**Returns:**

The date and time when the review was last accessed.

---

## opened

```
public Calendar opened()  
throws StorageConnectorException
```

Gets the describing IReview status opened time-stamp.

**Returns:**

The date and time when the review status was set to opened or a null reference if entity has no status.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- 

## closed

```
public Calendar closed()  
throws StorageConnectorException
```

Gets the describing IReview status closed time-stamp.

**Returns:**

The date and time when the review status was set to closed or a null reference if entity has no status.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getEffectiveRights

```
public IReadOnlyEntityRights getEffectiveRights()
```

Gets the users effective rights on the describing IReview.

**Returns:**

The read-only set of rights describing users effective rights on the review.

---

## getArchive

```
public IArchive getArchive()
```

Gets the IArchive where the describing IReview is stored.

**Returns:**

The IArchive where the review is stored.

---

## getCreator

```
public IDirectoryEntity getCreator()  
throws StorageConnectorException
```

Gets the describing IReview creator.

**Returns:**

A IDirectoryEntity that represents the creator of the review.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getOwner

```
public IDirectoryEntity getOwner()  
throws StorageConnectorException
```

Gets the describing IReview owner IDirectoryEntity.

**Returns:**

An IDirectoryEntity object representing the review owner or a null reference, if this review has no owner.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.



(continued from last page)

---

## getTemplate

```
public ITemplate getTemplate()
```

Gets the describing IReview template.

**Returns:**

An ITemplate object that represents the describing review template.

---

## getKeywords

```
public com.imis.util.IReadOnlyList getKeywords()  
    throws StorageConnectorException
```

Gets the describing IReview list of keywords.

**Returns:**

The review list of keywords.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- 

## getSubEntityTemplates

```
public com.imis.util.IReadOnlyList getSubEntityTemplates()
```

Gets the read-only list of templates that represent the describing IReview sub-entity creation templates.

**Returns:**

The read-only list of ITemplate objects that represent the review sub-entity creation templates.

---

## setStatus

```
public void setStatus(String status)  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Sets the status of the describing IReview.

**Note:** This operation allows user to pass a reason for changing review status. Reason can be passed through [AuditLog](#) message.

**Parameters:**

status - the describing IReview status or a null reference to reset status to inherited value.

**Throws:**

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Property value is not allowed based on the current template settings.
- Error occurred while getting properties.

IOException - if an I/O error occurs.

---

## getSystemProperties

```
public com.imis.util.IReadOnlyMap getSystemProperties()
```

Gets the read-only map of read-only system properties sorted by `SystemProperty` enum values.

**Returns:**

The read-only map of `IReadOnlyProperty` objects sorted by `SystemProperty` enum values.

---

## getSystemProperty

```
public IReadOnlyProperty getSystemProperty(SystemProperty systemProperty)  
throws StorageConnectorException
```

Gets the read-only property for the specified system property.

**Parameters:**

`systemProperty` - the system property.

**Returns:**

The read-only property for the specified system property.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- 

## areValuesInherited

```
public boolean areValuesInherited(SystemProperty systemProperty)  
throws StorageConnectorException
```

Gets the value indicating whether the specified system property values are inherited.

**Parameters:**

`systemProperty` - the system property.

**Returns:**

true if the specified system property values are inherited; otherwise false.

**Throws:**

`IllegalArgumentException` - if invalid system property.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- 

## getPickListValues

```
public com.imis.util.IReadOnlyList getPickListValues(SystemProperty systemProperty)  
throws StorageConnectorException
```

Gets the read-only list of the pick-list values for the specified system property.

**Parameters:**

`systemProperty` - the system property.

---

(continued from last page)

**Returns:**

The read-only list of `IPickListValue` objects representing the pick-list values for the specified system property.

**Throws:**

`IllegalArgumentException` - if invalid system property.

[`StorageConnectorException`](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Pick-list values are not supported for property type.
- Error occurred while verifying property value.
- Property value is not allowed based on the current template settings.

---

## getReport

```
public InputStream getReport(ReportType type)
```

Gets the specified report data stream for the describing `IReview`.

**Note:** Valid report type on `IReviewStub` is:

- [`ReportType.REVIEW\_STATUS`](#)

**Parameters:**

`type` - the report type.

**Returns:**

The report data input stream or `null` if report not available.

**Throws:**

`UnsupportedOperationException` - if report not supported.

---

## open

```
public IReview open(AccessMode mode)  
throws javax.security.auth.login.LoginException,  
       StorageConnectorException,  
       IOException
```

Opens a review described by this `IReviewStub` using a specified review access mode.

**Note:** This operation allows user to pass a reason for opening review. Reason can be passed through [AuditLog](#) message.

**Parameters:**

`mode` - the review access mode.

**Returns:**

An opened `IReview` instance.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Entity identifier is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while opening entity on archive.

`IOException` - if an I/O error occurs.

---

## hasAIP

```
public boolean hasAIP()
```

Gets a value indicating whether or not the describing `IReview` has AIP.

**Returns:**

`true` if the describing `IReview` has AIP; otherwise `false`.

## com.imis.storageconnector.entitymodel Interface ISearchedEntityStub

All Superinterfaces:

[IEntityStub](#)

public interface **ISearchedEntityStub**

extends [IEntityStub](#)

Defines an [IEntityStub](#) for a searched entity.

### Method Summary

com.imis.util.IReadOnlyList	<a href="#">getContentDescriptions()</a> Gets the describing searched <a href="#">IEntity</a> list of content descriptions.
-----------------------------	--

Methods inherited from interface [com.imis.storageconnector.entitymodel.IEntityStub](#)

[accessed](#), [areValuesInherited](#), [checkOut](#), [closed](#), [created](#), [delete](#), [delete](#), [getArchive](#), [getCategories](#), [getClassificationCode](#), [getContent](#), [getCreator](#), [getCustomContents](#), [getDescription](#), [getDraftCreator](#), [getDraftId](#), [getDraftInfo](#), [getEffectiveRights](#), [getExternalIds](#), [getId](#), [getKeywords](#), [getModifiedBy](#), [getOwner](#), [getParent](#), [getPickListValues](#), [getProperties](#), [getPublicClassificationCode](#), [getReferences](#), [getReport](#), [getRetention](#), [getSecurityClass](#), [getSignificance](#), [getStatus](#), [getSubEntities](#), [getSubEntityTemplates](#), [getSystemProperties](#), [getSystemProperty](#), [getTemplate](#), [getTitle](#), [getType](#), [getVersionComment](#), [getVersionLabel](#), [getVersions](#), [getVersionSequence](#), [hasAIP](#), [hasDraft](#), [hasReferences](#), [hasVersions](#), [isChildClassificationCodeGenerated](#), [isPermanent](#), [modified](#), [move](#), [open](#), [opened](#), [search](#), [setSecurityClass](#), [setStatus](#), [subEntityCount](#)

### Methods

#### getContentDescriptions

public com.imis.util.IReadOnlyList **getContentDescriptions()**  
throws [StorageConnectorException](#)

Gets the describing searched [IEntity](#) list of content descriptions.

#### Returns:

The describing searched [IEntity](#) list of content descriptions in case of full-text index searches and searches through content descriptions.

#### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

## com.imis.storageconnector.entitymodel Interface ISecurityClassChangeDetails

public interface **ISecurityClassChangeDetails**  
extends

Defines an IEntity security class change details.

An ISecurityClassChangeDetails defines an IEntity security class change details: agent, reason, date and time, and status attributes.

**See Also:**

[IEntity](#)

### Method Summary

<a href="#">IDirectoryEntity</a>	<a href="#">getAgent()</a> Gets the agent who changed the security class of the IEntity.
String	<a href="#">getReason()</a> Gets the reason for changing the security class of the IEntity.
String	<a href="#">getSecurityClassFrom()</a> Gets the IEntity security class before it was changed.
String	<a href="#">getSecurityClassTo()</a> Gets the IEntity security class after it was changed.
Calendar	<a href="#">modified()</a> Gets the IEntity security class modification date and time.

### Methods

#### getAgent

public [IDirectoryEntity](#) **getAgent()**

Gets the agent who changed the security class of the IEntity.

**Returns:**

The user who changed the security class of the IEntity.

#### getReason

public String **getReason()**

Gets the reason for changing the security class of the IEntity.

**Returns:**

A reason for changing the security class of the IEntity.

#### modified

public Calendar **modified()**

(continued from last page)

Gets the IEntity security class modification date and time.

**Returns:**

The date and time when the IEntity security class was modified.

---

## **getSecurityClassFrom**

```
public String getSecurityClassFrom()
```

Gets the IEntity security class before it was changed.

**Returns:**

The security class of the IEntity before it was changed.

---

## **getSecurityClassTo**

```
public String getSecurityClassTo()
```

Gets the IEntity security class after it was changed.

**Returns:**

The security class of the IEntity after it was changed.

## com.imis.storageconnector.entitymodel Interface IStringMaxValue

public interface **IStringMaxValue**  
extends

Represents an unlimited length string property, which has a read-write interface.

### Method Summary

String	<a href="#">getValue()</a> Gets the value of the IStringMaxValue as string.
InputStream	<a href="#">openDataInputStream()</a> Opens the string value data InputStream.
OutputStream	<a href="#">openDataOutputStream(boolean append)</a> Opens the string value data OutputStream.
void	<a href="#">setValue(String value)</a> Sets the value of the IStringMaxValue as string.

### Methods

#### getValue

```
public String getValue()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the value of the IStringMaxValue as string.

**Note:** This operation allows user to pass a reason for opening string value data input stream. Reason can be passed through [AuditLog](#) message.

#### Returns:

The string representing the value of the IStringMaxValue.

#### Throws:

[CancellationException](#) - if an operation was canceled by the user.  
[LoginException](#) - if a login error occurs.  
[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening string value data stream.

[IOException](#) - if an I/O error occurs.



(continued from last page)

## setValue

```
public void setValue(String value)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Sets the value of the IStringMaxValue as string.

**Note:** This operation allows user to pass a reason for opening string value data output stream. Reason can be passed through [AuditLog](#) message.

### Parameters:

value - the string representing the value of the IStringMaxValue.

### Throws:

NullPointerException - if value is null reference.

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening string value data stream.

IOException - if an I/O error occurs.

---

## openDataInputStream

```
public InputStream openDataInputStream()
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Opens the string value data InputStream.

This method enables the caller to open a data input stream from which value data can be read.

**Note:** This operation allows user to pass a reason for opening string value data input stream. Reason can be passed through [AuditLog](#) message.

### Returns:

A string value data InputStream.

### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Error occurred while opening property value data input stream.

IOException - if an I/O error occurs.

---

## openDataOutputStream

```
public OutputStream openDataOutputStream(boolean append)
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

(continued from last page)

Opens the string value data `OutputStream`.

This method enables the caller to open a data output stream to which value data can be written.

**Note:** This operation allows user to pass a reason for opening string value data output stream. Reason can be passed through [AuditLog](#) message.

**Parameters:**

`append` - `true` to append data to the end of the output stream; otherwise `false`.

**Returns:**

A string value data `OutputStream`.

**Throws:**

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Cannot open property value in read-write mode on a read-only property.
- Error occurred while opening property value data output stream.

`IOException` - if an I/O error occurs.

## com.imis.storageconnector.entitymodel Interface ITemplate

public interface **ITemplate**  
extends

Defines a template class on which an IEntity property structure is defined upon.

An ITemplate class defines and describes a property structure and constraints for an IEntity which has been based upon it.

Instance is automatically recycled after the last reference is released.

**See Also:**

[IEntity](#)

### Method Summary

long	<a href="#">entityCount()</a> Gets the template entity count.
<a href="#">IReadOnlyContent</a>	<a href="#">getContent()</a> Gets the read-only content assigned to the template.
com.imis.util.IReadOnlyMap	<a href="#">getCustomContents()</a> Gets the read-only map of custom read-only contents sorted by content identifiers assigned to the template.
String	<a href="#">getDescription()</a> Gets the template description.
<a href="#">EntityType</a>	<a href="#">getEntityType()</a> Gets the entity type which the template is defining.
String	<a href="#">getId()</a> Gets the template unique identifier.
String	<a href="#">getLabel()</a> Gets the template label.
com.imis.util.IReadOnlyList	<a href="#">getPickListValues(SystemProperty systemProperty)</a> Gets the read-only list of the pick-list values for the specified system property.
com.imis.util.IReadOnlyMap	<a href="#">getProperties()</a> Gets the read-only map of read-only properties sorted by property names representing the template metadata attributes.
com.imis.util.IReadOnlyList	<a href="#">getPropertyList()</a> Gets the read-only list of read-only properties representing the template metadata attributes.
com.imis.util.IReadOnlyMap	<a href="#">getSystemProperties()</a> Gets the read-only map of read-only system properties sorted by SystemProperty enum values representing the template system metadata attributes.
<a href="#">IReadOnlyProperty</a>	<a href="#">getSystemProperty(SystemProperty systemProperty)</a> Gets the read-only property for the specified system property.

boolean	<a href="#">isEmailArchivalCapable()</a> Gets the value indicating whether template can be used for email archival.
boolean	<a href="#">isVersionable()</a> Gets the value indicating whether template can be used for versioning.

## Methods

### getId

```
public String getId()
```

Gets the template unique identifier.

**Returns:**

A string that represents the template unique identifier.

### getDescription

```
public String getDescription()
```

Gets the template description.

**Returns:**

A string that represents the template description.

### getLabel

```
public String getLabel()
```

Gets the template label.

**Returns:**

A string that represents the template label.

### getEntityType

```
public EntityType getEntityType()
```

Gets the entity type which the template is defining.

Only IEntity objects of this type can be created using this template.

**Returns:**

An enumerator value that represents the entity type.

### entityCount

```
public long entityCount()
```

Gets the template entity count.

**Returns:**

The number of entities created with this template.

(continued from last page)

---

## isEmailArchivalCapable

```
public boolean isEmailArchivalCapable()
```

Gets the value indicating whether template can be used for email archival.

**Returns:**

true if template can be used for email archival; otherwise false

---

## isVersionable

```
public boolean isVersionable()
```

Gets the value indicating whether template can be used for versioning.

**Returns:**

true if template can be used for versioning; otherwise false

---

## getPropertyList

```
public com.imis.util.IReadOnlyList getPropertyList()
```

Gets the read-only list of read-only properties representing the template metadata attributes.

**Returns:**

The read-only list of IReadOnlyProperty objects.

---

## getProperties

```
public com.imis.util.IReadOnlyMap getProperties()
```

Gets the read-only map of read-only properties sorted by property names representing the template metadata attributes.

**Returns:**

The read-only map of IReadOnlyProperty objects sorted by property names.

---

## getContent

```
public IReadOnlyContent getContent()  
throws StorageConnectorException
```

Gets the read-only content assigned to the template. **Note:** This method is instantiated only for IDocument entities.

**Returns:**

A IReadOnlyContent object.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- 

## getCustomContents

```
public com.imis.util.IReadOnlyMap getCustomContents()
```

Gets the read-only map of custom read-only contents sorted by content identifiers assigned to the template. **Note:** This method is instantiated only for IDocument entities.

---

(continued from last page)

**Returns:**

A read-only map of `IReadOnlyContent` objects sorted by content identifiers.

---

## getSystemProperties

```
public com.imis.util.IReadOnlyMap getSystemProperties()
```

Gets the read-only map of read-only system properties sorted by `SystemProperty` enum values representing the template system metadata attributes.

**Returns:**

The read-only map of `IReadOnlyProperty` objects sorted by `SystemProperty` enum values.

---

## getSystemProperty

```
public IReadOnlyProperty getSystemProperty(SystemProperty systemProperty)  
    throws StorageConnectorException
```

Gets the read-only property for the specified system property.

**Parameters:**

`systemProperty` - the system property.

**Returns:**

The read-only property for the specified system property.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.

---

## getPickListValues

```
public com.imis.util.IReadOnlyList getPickListValues(SystemProperty systemProperty)  
    throws StorageConnectorException
```

Gets the read-only list of the pick-list values for the specified system property.

**Parameters:**

`systemProperty` - the system property.

**Returns:**

The read-only list of `IPickListValue` objects representing the pick-list values for the specified system property.

**Throws:**

`IllegalArgumentException` - if invalid system property.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Property is not present in template properties.
- Property is not a pick-list property.

## com.imis.storageconnector.entitymodel Interface ITransferDetails

public interface **ITransferDetails**  
extends

Describes a IEntity transfer details.

### Method Summary

String	<a href="#"><u>getAuditLog()</u></a> Gets the transferred entity audit log from previous archiving system.
String	<a href="#"><u>getClassificationCode()</u></a> Gets the transferred entity classification code from previous archiving system.
String	<a href="#"><u>getEvidence()</u></a> Gets the transferred entity evidence of authenticity.
com.imis.util.IReadOnlyList	<a href="#"><u>getMoveAgents()</u></a> Gets the list of transferred entity move agents.
com.imis.util.IReadOnlyList	<a href="#"><u>getMoveClassificationCodes()</u></a> Gets the list of transferred entity move classification codes.
com.imis.util.IReadOnlyList	<a href="#"><u>getMoveDateTimes()</u></a> Gets the list of transferred entity move date and times.
com.imis.util.IReadOnlyList	<a href="#"><u>getMoveReasons()</u></a> Gets the list of transferred entity move reasons.
String	<a href="#"><u>getSystemId()</u></a> Gets the transferred entity unique identifier from previous archiving system.
Calendar	<a href="#"><u>imported()</u></a> Gets the date and time when the transferred entity was imported into archive.

### Methods

#### getAuditLog

```
public String getAuditLog()
    throws javax.security.auth.login.LoginException,
           StorageConnectorException,
           IOException
```

Gets the transferred entity audit log from previous archiving system.

#### Returns:

A string that represents the transferred entity audit log.

#### Throws:

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while opening string value data stream.

[IOException](#) - if an I/O error occurs.

---

## getSystemId

```
public String getSystemId()  
    throws StorageConnectorException
```

Gets the transferred entity unique identifier from previous archiving system.

### Returns:

The transferred entity unique system identifier.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getClassificationCode

```
public String getClassificationCode()  
    throws StorageConnectorException
```

Gets the transferred entity classification code from previous archiving system.

### Returns:

The transferred entity classification code.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## imported

```
public Calendar imported()  
    throws StorageConnectorException
```

Gets the date and time when the transferred entity was imported into archive.

### Returns:

The transferred entity imported date and time.



(continued from last page)

**Throws:**[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getEvidence

```
public String getEvidence()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the transferred entity evidence of authenticity.

**Returns:**

A string that represents the transferred evidence record of authenticity.

**Throws:**

CancellationException - if an operation was canceled by the user.

LoginException - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.
- Error occurred while opening string value data stream.

IOException - if an I/O error occurs.

---

## getMoveReasons

```
public com.imis.util.IReadOnlyList getMoveReasons()  
    throws StorageConnectorException
```

Gets the list of transferred entity move reasons.

**Returns:**

A list of strings that represent the transferred entity move reasons.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getMoveAgents

```
public com.imis.util.IReadOnlyList getMoveAgents()  
    throws StorageConnectorException
```

(continued from last page)

Gets the list of transferred entity move agents.

**Returns:**

A list of strings that represent the transferred entity move agents.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getMoveDateTimes

```
public com.imis.util.IReadOnlyList getMoveDateTimes()  
throws StorageConnectorException
```

Gets the list of transferred entity move date and times.

**Returns:**

A list of strings that represent the transferred entity move date and times.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

---

## getMoveClassificationCodes

```
public com.imis.util.IReadOnlyList getMoveClassificationCodes()  
throws StorageConnectorException
```

Gets the list of transferred entity move classification codes.

**Returns:**

A list of strings that represent the transferred entity move classification codes.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed entity is not allowed.
- Operation on entity opened in read-only mode is not allowed.
- Error occurred while getting properties.
- Property is not present in template properties.

# com.imis.storageconnector.entitymodel

## Class LogType

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.entitymodel.LogType
  
```

**All Implemented Interfaces:**  
 Serializable, Comparable

```

public final class LogType
extends Enum
  
```

Defines constants for different types of logs.

### Field Summary

public static final	<a href="#">EXPORT</a> Export logs.
public static final	<a href="#">IMPORT</a> Import logs.

### Method Summary

static <a href="#">LogType</a>	<a href="#">valueOf</a> (String name)
static <a href="#">LogType[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields

#### IMPORT

```

public static final com.imis.storageconnector.entitymodel.LogType IMPORT

    Import logs.
  
```

---

## EXPORT

```
public static final com.imis.storageconnector.entitymodel.LogType EXPORT
```

Export logs.

## Methods

### values

```
public final static LogType[] values()
```

---

### valueOf

```
public static LogType valueOf(String name)
```

## com.imis.storageconnector.entitymodel Class PropertyType

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.entitymodel.PropertyType
  
```

**All Implemented Interfaces:**  
Serializable, Comparable

```

public final class PropertyType
extends Enum
  
```

Represents the type of the IProperty.

**See Also:**

[IProperty](#)

### Field Summary

public static final	<a href="#">BINARY</a> Binary property type.
public static final	<a href="#">BOOL</a> Logical boolean property type (true/false).
public static final	<a href="#">DATE</a> Date-only property type.
public static final	<a href="#">DATE_TIME</a> Date and time property type.
public static final	<a href="#">DECIMAL1</a> Signed 64-bit floating-point property type with 1 decimal place.
public static final	<a href="#">DECIMAL10</a> Signed 64-bit floating-point property type with 10 decimal places.
public static final	<a href="#">DECIMAL2</a> Signed 64-bit floating-point property type with 2 decimal places.
public static final	<a href="#">DECIMAL3</a> Signed 64-bit floating-point property type with 3 decimal places.
public static final	<a href="#">DECIMAL4</a> Signed 64-bit floating-point property type with 4 decimal places.
public static final	<a href="#">DECIMAL5</a> Signed 64-bit floating-point property type with 5 decimal places.
public static final	<a href="#">DECIMAL6</a> Signed 64-bit floating-point property type with 6 decimal places.
public static final	<a href="#">DECIMAL7</a> Signed 64-bit floating-point property type with 7 decimal places.

public static final	<a href="#">DECIMAL8</a> Signed 64-bit floating-point property type with 8 decimal places.
public static final	<a href="#">DECIMAL9</a> Signed 64-bit floating-point property type with 9 decimal places.
public static final	<a href="#">DIRECTORY_ENTITY</a> Directory entity property type.
public static final	<a href="#">DOUBLE</a> 64-bit floating-point property type.
public static final	<a href="#">ENTITY_REFERENCE</a> Entity reference property type.
public static final	<a href="#">INT128</a> Signed 128-bit integral property type.
public static final	<a href="#">INT16</a> Signed 16-bit integral property type.
public static final	<a href="#">INT32</a> Signed 32-bit integral property type.
public static final	<a href="#">INT64</a> Signed 64-bit integral property type.
public static final	<a href="#">INT8</a> Signed 8-bit integral property type.
public static final	<a href="#">STRING10</a> Up to 10 single byte (8-bit) character string property type.
public static final	<a href="#">STRING100</a> Up to 100 single byte (8-bit) character string property type.
public static final	<a href="#">STRING20</a> Up to 20 single byte (8-bit) character string property type.
public static final	<a href="#">STRING200</a> Up to 200 single byte (8-bit) character string property type.
public static final	<a href="#">STRING30</a> Up to 30 single byte (8-bit) character string property type.
public static final	<a href="#">STRING40</a> Up to 40 single byte (8-bit) character string property type.
public static final	<a href="#">STRING50</a> Up to 50 single byte (8-bit) character string property type.
public static final	<a href="#">STRINGMAX</a> Unlimited length single byte (8-bit) character string property type.
public static final	<a href="#">TIME</a> Time-only property type.
public static final	<a href="#">UINT128</a> Unsigned 128-bit integral property type.

<code>public static final</code>	<a href="#">UINT16</a> Unsigned 16-bit integral property type.
<code>public static final</code>	<a href="#">UINT32</a> Unsigned 32-bit integral property type.
<code>public static final</code>	<a href="#">UINT64</a> Unsigned 64-bit integral property type.
<code>public static final</code>	<a href="#">UINT8</a> Unsigned 8-bit integral property type.
<code>public static final</code>	<a href="#">UNKNOWN</a> Unknown property type.

## Method Summary

<code>static <a href="#">PropertyType</a></code>	<code><a href="#">valueOf</a>(String name)</code>
<code>static <a href="#">PropertyType[]</a></code>	<code><a href="#">values</a>()</code>

### Methods inherited from class `java.lang.Enum`

`clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

### Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods inherited from interface `java.lang.Comparable`

`compareTo`

## Fields

### UNKNOWN

`public static final com.imis.storageconnector.entitymodel.PropertyType UNKNOWN`

Unknown property type.

This value defines an invalid property type which is never returned on an coherent property.

### DIRECTORY\_ENTITY

`public static final com.imis.storageconnector.entitymodel.PropertyType DIRECTORY_ENTITY`

Directory entity property type.

Properties of this type use [IDirectoryEntity](#) to interface with its value.

(continued from last page)

## BOOL

```
public static final com.imis.storageconnector.entitymodel.PropertyType BOOL
```

Logical boolean property type (true/false).

Properties of this type use `java.lang.Boolean` to interface with its value.

---

## INT8

```
public static final com.imis.storageconnector.entitymodel.PropertyType INT8
```

Signed 8-bit integral property type.

Properties of this type use `java.lang.Byte` to interface with its value.

---

## UINT8

```
public static final com.imis.storageconnector.entitymodel.PropertyType UINT8
```

Unsigned 8-bit integral property type.

Properties of this type use `java.lang.Short` to interface with its value.

---

## INT16

```
public static final com.imis.storageconnector.entitymodel.PropertyType INT16
```

Signed 16-bit integral property type.

Properties of this type use `java.lang.Short` to interface with its value.

---

## UINT16

```
public static final com.imis.storageconnector.entitymodel.PropertyType UINT16
```

Unsigned 16-bit integral property type.

Properties of this type use `java.lang.Integer` to interface with its value.

---

## INT32

```
public static final com.imis.storageconnector.entitymodel.PropertyType INT32
```

Signed 32-bit integral property type.

Properties of this type use `java.lang.Integer` to interface with its value.

---

## UINT32

```
public static final com.imis.storageconnector.entitymodel.PropertyType UINT32
```

Unsigned 32-bit integral property type.

Properties of this type use `java.lang.Long` to interface with its value.

---

## INT64

```
public static final com.imis.storageconnector.entitymodel.PropertyType INT64
```



(continued from last page)

Signed 64-bit integral property type.

Properties of this type use `java.lang.Long` to interface with its value.

---

## UINT64

```
public static final com.imis.storageconnector.entitymodel.PropertyType UINT64
```

Unsigned 64-bit integral property type.

Properties of this type use `java.math.BigInteger` to interface with its value.

---

## INT128

```
public static final com.imis.storageconnector.entitymodel.PropertyType INT128
```

Signed 128-bit integral property type.

Properties of this type use `java.math.BigInteger` to interface with its value.

---

## UINT128

```
public static final com.imis.storageconnector.entitymodel.PropertyType UINT128
```

Unsigned 128-bit integral property type.

Properties of this type use `java.math.BigInteger` to interface with its value.

---

## DOUBLE

```
public static final com.imis.storageconnector.entitymodel.PropertyType DOUBLE
```

64-bit floating-point property type.

Properties of this type use `java.lang.Double` to interface with its value.

---

## DATE

```
public static final com.imis.storageconnector.entitymodel.PropertyType DATE
```

Date-only property type.

Properties of this type use `java.util.Calendar` to interface with its value.

---

## TIME

```
public static final com.imis.storageconnector.entitymodel.PropertyType TIME
```

Time-only property type.

Properties of this type use `java.util.Calendar` to interface with its value.

---

## DATE\_TIME

```
public static final com.imis.storageconnector.entitymodel.PropertyType DATE_TIME
```

Date and time property type.

Properties of this type use `java.util.Calendar` to interface with its value.

---

(continued from last page)

---

## STRINGMAX

```
public static final com.imis.storageconnector.entitymodel.PropertyType STRINGMAX
```

Unlimited length single byte (8-bit) character string property type.

Properties of this type use [IStringMaxValue](#) object to interface with its value.

---

## STRING10

```
public static final com.imis.storageconnector.entitymodel.PropertyType STRING10
```

Up to 10 single byte (8-bit) character string property type.

Properties of this type use `java.lang.String` to interface with its value.

---

## STRING20

```
public static final com.imis.storageconnector.entitymodel.PropertyType STRING20
```

Up to 20 single byte (8-bit) character string property type.

Properties of this type use `java.lang.String` to interface with its value.

---

## STRING30

```
public static final com.imis.storageconnector.entitymodel.PropertyType STRING30
```

Up to 30 single byte (8-bit) character string property type.

Properties of this type use `java.lang.String` to interface with its value.

---

## STRING40

```
public static final com.imis.storageconnector.entitymodel.PropertyType STRING40
```

Up to 40 single byte (8-bit) character string property type.

Properties of this type use `java.lang.String` to interface with its value.

---

## STRING50

```
public static final com.imis.storageconnector.entitymodel.PropertyType STRING50
```

Up to 50 single byte (8-bit) character string property type.

Properties of this type use `java.lang.String` to interface with its value.

---

## STRING100

```
public static final com.imis.storageconnector.entitymodel.PropertyType STRING100
```

Up to 100 single byte (8-bit) character string property type.

Properties of this type use `java.lang.String` to interface with its value.

---

## STRING200

```
public static final com.imis.storageconnector.entitymodel.PropertyType STRING200
```

---

(continued from last page)

Up to 200 single byte (8-bit) character string property type.

Properties of this type use `java.lang.String` to interface with its value.

---

## ENTITY\_REFERENCE

```
public static final com.imis.storageconnector.entitymodel.PropertyType  
ENTITY_REFERENCE
```

Entity reference property type.

Values of this type interface with `java.lang.String` type.

---

## DECIMAL1

```
public static final com.imis.storageconnector.entitymodel.PropertyType DECIMAL1
```

Signed 64-bit floating-point property type with 1 decimal place.

Properties of this type use `java.math.BigDecimal` to interface with its value.

---

## DECIMAL2

```
public static final com.imis.storageconnector.entitymodel.PropertyType DECIMAL2
```

Signed 64-bit floating-point property type with 2 decimal places.

Properties of this type use `java.math.BigDecimal` to interface with its value.

---

## DECIMAL3

```
public static final com.imis.storageconnector.entitymodel.PropertyType DECIMAL3
```

Signed 64-bit floating-point property type with 3 decimal places.

Properties of this type use `java.math.BigDecimal` to interface with its value.

---

## DECIMAL4

```
public static final com.imis.storageconnector.entitymodel.PropertyType DECIMAL4
```

Signed 64-bit floating-point property type with 4 decimal places.

Properties of this type use `java.math.BigDecimal` to interface with its value.

---

## DECIMAL5

```
public static final com.imis.storageconnector.entitymodel.PropertyType DECIMAL5
```

Signed 64-bit floating-point property type with 5 decimal places.

Properties of this type use `java.math.BigDecimal` to interface with its value.

---

## DECIMAL6

```
public static final com.imis.storageconnector.entitymodel.PropertyType DECIMAL6
```

Signed 64-bit floating-point property type with 6 decimal places.

Properties of this type use `java.math.BigDecimal` to interface with its value.

---

## DECIMAL7

```
public static final com.imis.storageconnector.entitymodel.PropertyType DECIMAL7
```

Signed 64-bit floating-point property type with 7 decimal places.

Properties of this type use `java.math.BigDecimal` to interface with its value.

---

## DECIMAL8

```
public static final com.imis.storageconnector.entitymodel.PropertyType DECIMAL8
```

Signed 64-bit floating-point property type with 8 decimal places.

Properties of this type use `java.math.BigDecimal` to interface with its value.

---

## DECIMAL9

```
public static final com.imis.storageconnector.entitymodel.PropertyType DECIMAL9
```

Signed 64-bit floating-point property type with 9 decimal places.

Properties of this type use `java.math.BigDecimal` to interface with its value.

---

## DECIMAL10

```
public static final com.imis.storageconnector.entitymodel.PropertyType DECIMAL10
```

Signed 64-bit floating-point property type with 10 decimal places.

Properties of this type use `java.math.BigDecimal` to interface with its value.

---

## BINARY

```
public static final com.imis.storageconnector.entitymodel.PropertyType BINARY
```

Binary property type.

Properties of this type use [IBinaryValue](#) object to interface with its value.

## Methods

### values

```
public final static PropertyType\[\] values()
```

### valueOf

```
public static PropertyType valueOf(String name)
```

## com.imis.storageconnector.entitymodel

### Class ReportType

java.lang.Object

└- java.lang.Enum

└- com.imis.storageconnector.entitymodel.ReportType

#### All Implemented Interfaces:

Serializable, Comparable

public final class **ReportType**  
extends Enum

Defines constants for different types of reports.

### Field Summary

public static final	<a href="#">ENTITY_MOVE</a> Entity move report in UTF8 format.
public static final	<a href="#">ENTITY_SAVE</a> Entity save report in UTF8 format.
public static final	<a href="#">ENTITY_SECURITY_CLASS</a> Entity security class report in UTF8 format.
public static final	<a href="#">ENTITY_STATUS</a> Entity status report in UTF8 format.
public static final	<a href="#">REVIEW_SAVE</a> Review save report in UTF8 format.
public static final	<a href="#">REVIEW_STATUS</a> Review status report in UTF8 format.

### Method Summary

static <a href="#">ReportType</a>	<a href="#">valueOf</a> (String name)
static <a href="#">ReportType[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

```
compareTo
```

## Fields

### ENTITY\_SAVE

```
public static final com.imis.storageconnector.entitymodel.ReportType ENTITY_SAVE
```

Entity save report in UTF8 format.

### ENTITY\_MOVE

```
public static final com.imis.storageconnector.entitymodel.ReportType ENTITY_MOVE
```

Entity move report in UTF8 format.

### ENTITY\_STATUS

```
public static final com.imis.storageconnector.entitymodel.ReportType ENTITY_STATUS
```

Entity status report in UTF8 format.

### ENTITY\_SECURITY\_CLASS

```
public static final com.imis.storageconnector.entitymodel.ReportType  
ENTITY_SECURITY_CLASS
```

Entity security class report in UTF8 format.

### REVIEW\_SAVE

```
public static final com.imis.storageconnector.entitymodel.ReportType REVIEW_SAVE
```

Review save report in UTF8 format.

### REVIEW\_STATUS

```
public static final com.imis.storageconnector.entitymodel.ReportType REVIEW_STATUS
```

Review status report in UTF8 format.

## Methods

### values

```
public final static ReportType[] values()
```

### valueOf

```
public static ReportType valueOf(String name)
```

## com.imis.storageconnector.entitymodel

### Class RetentionPolicyFilter

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.entitymodel.RetentionPolicyFilter
  
```

**All Implemented Interfaces:**  
 Serializable, Comparable

public final class **RetentionPolicyFilter**  
 extends Enum

Defines types of retention policy filter.

#### Field Summary

public static final	<a href="#">CLASSES</a> Retention has effect on classes.
public static final	<a href="#">DOCUMENTS</a> Retention has effect on documents under classes.
public static final	<a href="#">FOLDERS</a> Retention has effect on folders.

#### Method Summary

static <a href="#">RetentionPolicyFilter</a>	<a href="#">valueOf</a> (String name)
static <a href="#">RetentionPolicyFilter</a> []	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

#### Fields

(continued from last page)

---

## CLASSES

```
public static final com.imis.storageconnector.entitymodel.RetentionPolicyFilter  
CLASSES
```

Retention has effect on classes.

---

## FOLDERS

```
public static final com.imis.storageconnector.entitymodel.RetentionPolicyFilter  
FOLDERS
```

Retention has effect on folders.

---

## DOCUMENTS

```
public static final com.imis.storageconnector.entitymodel.RetentionPolicyFilter  
DOCUMENTS
```

Retention has effect on documents under classes.

## Methods

### values

```
public final static RetentionPolicyFilter\[\] values()
```

---

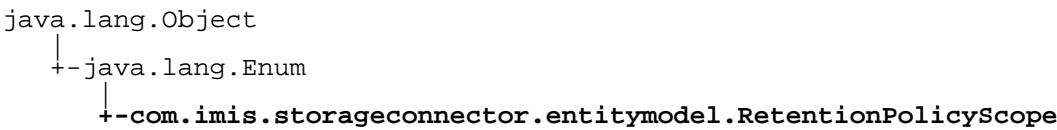
### valueOf

```
public static RetentionPolicyFilter valueOf(String name)
```



# com.imis.storageconnector.entitymodel

## Class RetentionPolicyScope



All Implemented Interfaces:  
Serializable, Comparable

public final class **RetentionPolicyScope**  
extends Enum

Defines enumeration for retention policy scope.

### Field Summary

public static final	<a href="#">ALLOW</a> Allowed.
public static final	<a href="#">DENY</a> Denied.

### Method Summary

<code>static</code> <a href="#">RetentionPolicyScope</a>	<a href="#">valueOf</a> (String name)
<code>static</code> <a href="#">RetentionPolicyScope</a> [ ]	<a href="#">values</a> ()

Methods inherited from class <code>java.lang.Enum</code>
<code>clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf</code>

Methods inherited from class <code>java.lang.Object</code>
<code>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

Methods inherited from interface <code>java.lang.Comparable</code>
<code>compareTo</code>

### Fields

#### ALLOW

public static final com.imis.storageconnector.entitymodel.RetentionPolicyScope **ALLOW**

(continued from last page)

Allowed.

**DENY**

public static final com.imis.storageconnector.entitymodel.RetentionPolicyScope **DENY**

Denied.

Methods

**values**

public final static [RetentionPolicyScope](#)[] **values**()

**valueOf**

public static [RetentionPolicyScope](#) **valueOf**(String name)

## com.imis.storageconnector.entitymodel Class SearchOptions

```

java.lang.Object
  |
  +- java.lang.Enum
        +- com.imis.storageconnector.entitymodel.SearchOptions

```

**All Implemented Interfaces:**  
Serializable, Comparable

public final class **SearchOptions**  
extends Enum

Defines constants for different search options.

Using empty set of SearchOptions the search is performed on first level child entities.

### Field Summary

public static final	<a href="#">INHERITED_VALUES</a> Search is performed on explicit and inherited values.
public static final	<a href="#">RECURSIVE</a> Search is performed recursively on all child entity levels.

### Method Summary

static <a href="#">SearchOptions</a>	<a href="#">valueOf</a> (String name)
static <a href="#">SearchOptions[]</a>	<a href="#">values</a> ()

#### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods inherited from interface java.lang.Comparable

compareTo

### Fields

(continued from last page)

## RECURSIVE

```
public static final com.imis.storageconnector.entitymodel.SearchOptions RECURSIVE
```

Search is performed recursively on all child entity levels.

---

## INHERITED\_VALUES

```
public static final com.imis.storageconnector.entitymodel.SearchOptions  
INHERITED_VALUES
```

Search is performed on explicit and inherited values. If not set search is performed on explicit values only.

## Methods

### values

```
public final static SearchOptions\[\] values()
```

---

### valueOf

```
public static SearchOptions valueOf(String name)
```

## com.imis.storageconnector.entitymodel

### Class SystemProperty

```

java.lang.Object
  |
  +- java.lang.Enum
        |
        +- com.imis.storageconnector.entitymodel.SystemProperty
  
```

**All Implemented Interfaces:**  
 Serializable, Comparable

```

public final class SystemProperty
extends Enum
  
```

Defines constants for different system properties.

Field Summary	
public static final	<a href="#">CATEGORY</a> Entity category.
public static final	<a href="#">CLASSIFICATION_CODE</a> Classification code.
public static final	<a href="#">CLOSED</a> Closed.
public static final	<a href="#">CONTENT</a> Content.
public static final	<a href="#">CREATOR</a> creator.
public static final	<a href="#">DELETED_ENTITY_AGENT</a> Deleted entity agent.
public static final	<a href="#">DELETED_ENTITY_CLASSIFICATION_CODE</a> Deleted entity classification code.
public static final	<a href="#">DELETED_ENTITY_DELETE_DATETIME</a> Deleted entity removed date and time.
public static final	<a href="#">DELETED_ENTITY_REASON</a> Deleted entity reason.
public static final	<a href="#">DELETED_ENTITY_REFERENCE</a> Deleted entity reference.
public static final	<a href="#">DESCRIPTION</a> Description.
public static final	<a href="#">DISPOSITION_HOLD_REASON</a> Retention policy reason.

public static final	<a href="#"><u>EMAIL_ENTITY_DATETIME</u></a> Email entity date and time.
public static final	<a href="#"><u>EMAIL_ENTITY_FROM</u></a> Email entity sender.
public static final	<a href="#"><u>EMAIL_ENTITY_IS_SIGNED</u></a> Email entity is signed value.
public static final	<a href="#"><u>EMAIL_ENTITY_MESSAGE_ID</u></a> Email entity message identifier.
public static final	<a href="#"><u>EMAIL_ENTITY_PRIORITY</u></a> Email entity delivery priority.
public static final	<a href="#"><u>EMAIL_ENTITY_SUBJECT</u></a> Email entity subject.
public static final	<a href="#"><u>EMAIL_ENTITY_TO</u></a> Email entity recipient.
public static final	<a href="#"><u>EMAIL_ENTITY_TOBCC</u></a> Email entity concealed recipients of copy.
public static final	<a href="#"><u>EMAIL_ENTITY_TOCC</u></a> Email entity recipients of copy.
public static final	<a href="#"><u>EXTERNAL_IDS</u></a> External identifiers.
public static final	<a href="#"><u>KEYWORDS</u></a> Keywords.
public static final	<a href="#"><u>MODIFIED_BY</u></a> Modified by.
public static final	<a href="#"><u>MOVED_ENTITY_AGENT</u></a> Moved entity agent.
public static final	<a href="#"><u>MOVED_ENTITY_CLASSIFICATION_CODE</u></a> Moved entity classification code.
public static final	<a href="#"><u>MOVED_ENTITY_MOVE_DATETIME</u></a> Moved entity moved date and time.
public static final	<a href="#"><u>MOVED_ENTITY_REASON</u></a> Moved entity reason.
public static final	<a href="#"><u>OPENED</u></a> Opened.
public static final	<a href="#"><u>OWNER</u></a> Owner.
public static final	<a href="#"><u>PHYSICAL_ENTITY_CURRENT_LOCATION</u></a> Physical entity current location.
public static final	<a href="#"><u>PHYSICAL_ENTITY_CUSTODIAN</u></a> Physical entity custodian.

public static final	<a href="#"><u>PHYSICAL_ENTITY_DESCRIPTION</u></a> Physical entity description.
public static final	<a href="#"><u>PHYSICAL_ENTITY_HOME_LOCATION</u></a> Physical entity home location.
public static final	<a href="#"><u>PHYSICAL_ENTITY_IDENTIFIER</u></a> Physical entity identifier.
public static final	<a href="#"><u>PHYSICAL_ENTITY_RETURN_DUE_DATETIME</u></a> Physical entity return due date and time.
public static final	<a href="#"><u>PHYSICAL_ENTITY_STATUS</u></a> Physical entity status.
public static final	<a href="#"><u>PHYSICAL_ENTITY_STATUS_CHANGE_DATETIME</u></a> Physical entity status changed date and time.
public static final	<a href="#"><u>REFERENCE</u></a> Entity reference.
public static final	<a href="#"><u>RETENTION_POLICY_ACTION</u></a> Retention policy action.
public static final	<a href="#"><u>RETENTION_POLICY_DETAILED_DESCRIPTION</u></a> Retention policy detailed description.
public static final	<a href="#"><u>RETENTION_POLICY_REASON</u></a> Retention policy reason.
public static final	<a href="#"><u>RETENTION_POLICY_REFERENCE</u></a> Retention policy reference.
public static final	<a href="#"><u>RETENTION_POLICY_TRIGGER</u></a> Retention policy trigger.
public static final	<a href="#"><u>REVIEW_ACTION</u></a> Review action.
public static final	<a href="#"><u>REVIEW_COMMENTS</u></a> Review comments.
public static final	<a href="#"><u>REVIEW_MEMBERS</u></a> Review members.
public static final	<a href="#"><u>REVIEW_MESSAGE</u></a> Review message.
public static final	<a href="#"><u>REVIEW_QUERY_EXPRESSION</u></a> Review query expression.
public static final	<a href="#"><u>REVIEW_SCHEDULED_ITEMS</u></a> Review entity lists.
public static final	<a href="#"><u>REVIEW_SCHEDULES</u></a> Review schedules.
public static final	<a href="#"><u>REVIEW_SCOPE</u></a> Review scope entity identifier.

public static final	<a href="#"><u>REVIEW_STATE</u></a> Review state.
public static final	<a href="#"><u>SAVE_LOG</u></a> Save log.
public static final	<a href="#"><u>SECURITY_CLASS</u></a> Security class.
public static final	<a href="#"><u>SECURITY_CLASS_CHANGE_AGENT</u></a> Security class change agent.
public static final	<a href="#"><u>SECURITY_CLASS_CHANGE_FROM</u></a> Security class prior to change.
public static final	<a href="#"><u>SECURITY_CLASS_CHANGE_MODIFIED_DATETIME</u></a> Security class modification date and time.
public static final	<a href="#"><u>SECURITY_CLASS_CHANGE_REASON</u></a> Security class change reason.
public static final	<a href="#"><u>SECURITY_CLASS_CHANGE_TO</u></a> Security class after the change.
public static final	<a href="#"><u>SIGNIFICANCE</u></a> Significance.
public static final	<a href="#"><u>STATUS</u></a> Status.
public static final	<a href="#"><u>TITLE</u></a> Title.
public static final	<a href="#"><u>TRANSFERRED_ENTITY_AUDIT_LOG</u></a> Transferred entity audit log.
public static final	<a href="#"><u>TRANSFERRED_ENTITY_CLASSIFICATION_CODE</u></a> Transferred entity classification code.
public static final	<a href="#"><u>TRANSFERRED_ENTITY_EVIDENCE</u></a> Transferred entity evidence of authenticity.
public static final	<a href="#"><u>TRANSFERRED_ENTITY_IMPORT_DATETIME</u></a> Transferred entity imported date and time.
public static final	<a href="#"><u>TRANSFERRED_ENTITY_MOVE_AGENT</u></a> Transferred entity move agent.
public static final	<a href="#"><u>TRANSFERRED_ENTITY_MOVE_CLASSIFICATION_CODE</u></a> Transferred entity move classification code.
public static final	<a href="#"><u>TRANSFERRED_ENTITY_MOVE_DATETIME</u></a>
public static final	<a href="#"><u>TRANSFERRED_ENTITY_MOVE_REASON</u></a> Transferred entity move reason.
public static final	<a href="#"><u>TRANSFERRED_ENTITY_SYSTEM_ID</u></a> Transferred entity system identifier.



public static final	<a href="#">VERSION_SERIES_ID</a> Version series identifier.
public static final	<a href="#">VERSIONING_COMMENT</a> Versioning check in comment.
public static final	<a href="#">VERSIONING_LABEL</a> Versioning series label.

## Method Summary

static <a href="#">SystemProperty</a>	<a href="#">valueOf</a> (String name)
static <a href="#">SystemProperty[]</a>	<a href="#">values</a> ()

### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods inherited from interface java.lang.Comparable

compareTo

## Fields

### CLASSIFICATION\_CODE

```
public static final com.imis.storageconnector.entitymodel.SystemProperty
CLASSIFICATION_CODE
```

Classification code.

### TITLE

```
public static final com.imis.storageconnector.entitymodel.SystemProperty TITLE
```

Title.

### DESCRIPTION

```
public static final com.imis.storageconnector.entitymodel.SystemProperty DESCRIPTION
```

Description.

### STATUS

```
public static final com.imis.storageconnector.entitymodel.SystemProperty STATUS
```

(continued from last page)

Status.

**OPENED**

public static final com.imis.storageconnector.entitymodel.SystemProperty **OPENED**

Opened.

**CLOSED**

public static final com.imis.storageconnector.entitymodel.SystemProperty **CLOSED**

Closed.

**SIGNIFICANCE**

public static final com.imis.storageconnector.entitymodel.SystemProperty **SIGNIFICANCE**

Significance.

**CREATOR**

public static final com.imis.storageconnector.entitymodel.SystemProperty **CREATOR**

creator.

**OWNER**

public static final com.imis.storageconnector.entitymodel.SystemProperty **OWNER**

Owner.

**MODIFIED\_BY**

public static final com.imis.storageconnector.entitymodel.SystemProperty **MODIFIED\_BY**

Modified by.

**KEYWORDS**

public static final com.imis.storageconnector.entitymodel.SystemProperty **KEYWORDS**

Keywords.

**EXTERNAL\_IDS**

public static final com.imis.storageconnector.entitymodel.SystemProperty **EXTERNAL\_IDS**

External identifiers.

**CONTENT**

public static final com.imis.storageconnector.entitymodel.SystemProperty **CONTENT**

Content.

(continued from last page)

---

## SAVE\_LOG

```
public static final com.imis.storageconnector.entitymodel.SystemProperty SAVE_LOG
```

Save log.

---

## SECURITY\_CLASS

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
SECURITY_CLASS
```

Security class.

---

## REFERENCE

```
public static final com.imis.storageconnector.entitymodel.SystemProperty REFERENCE
```

Entity reference.

---

## CATEGORY

```
public static final com.imis.storageconnector.entitymodel.SystemProperty CATEGORY
```

Entity category.

---

## SECURITY\_CLASS\_CHANGE\_AGENT

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
SECURITY_CLASS_CHANGE_AGENT
```

Security class change agent.

---

## SECURITY\_CLASS\_CHANGE\_REASON

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
SECURITY_CLASS_CHANGE_REASON
```

Security class change reason.

---

## SECURITY\_CLASS\_CHANGE\_MODIFIED\_DATETIME

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
SECURITY_CLASS_CHANGE_MODIFIED_DATETIME
```

Security class modification date and time.

---

## SECURITY\_CLASS\_CHANGE\_FROM

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
SECURITY_CLASS_CHANGE_FROM
```

Security class prior to change.

---

## SECURITY\_CLASS\_CHANGE\_TO

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
SECURITY_CLASS_CHANGE_TO
```

Security class after the change.

---

---

## MOVED\_ENTITY\_AGENT

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
MOVED_ENTITY_AGENT
```

Moved entity agent.

---

## MOVED\_ENTITY\_REASON

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
MOVED_ENTITY_REASON
```

Moved entity reason.

---

## MOVED\_ENTITY\_MOVE\_DATETIME

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
MOVED_ENTITY_MOVE_DATETIME
```

Moved entity moved date and time.

---

## MOVED\_ENTITY\_CLASSIFICATION\_CODE

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
MOVED_ENTITY_CLASSIFICATION_CODE
```

Moved entity classification code.

---

## DELETED\_ENTITY\_AGENT

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
DELETED_ENTITY_AGENT
```

Deleted entity agent.

---

## DELETED\_ENTITY\_REASON

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
DELETED_ENTITY_REASON
```

Deleted entity reason.

---

## DELETED\_ENTITY\_DELETE\_DATETIME

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
DELETED_ENTITY_DELETE_DATETIME
```

Deleted entity removed date and time.

---

## DELETED\_ENTITY\_CLASSIFICATION\_CODE

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
DELETED_ENTITY_CLASSIFICATION_CODE
```

Deleted entity classification code.

---

(continued from last page)

---

## DELETED\_ENTITY\_REFERENCE

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
DELETED_ENTITY_REFERENCE
```

Deleted entity reference.

---

## TRANSFERRED\_ENTITY\_SYSTEM\_ID

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
TRANSFERRED_ENTITY_SYSTEM_ID
```

Transferred entity system identifier.

---

## TRANSFERRED\_ENTITY\_CLASSIFICATION\_CODE

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
TRANSFERRED_ENTITY_CLASSIFICATION_CODE
```

Transferred entity classification code.

---

## TRANSFERRED\_ENTITY\_AUDIT\_LOG

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
TRANSFERRED_ENTITY_AUDIT_LOG
```

Transferred entity audit log.

---

## TRANSFERRED\_ENTITY\_IMPORT\_DATETIME

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
TRANSFERRED_ENTITY_IMPORT_DATETIME
```

Transferred entity imported date and time.

---

## TRANSFERRED\_ENTITY\_EVIDENCE

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
TRANSFERRED_ENTITY_EVIDENCE
```

Transferred entity evidence of authenticity.

---

## TRANSFERRED\_ENTITY\_MOVE\_REASON

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
TRANSFERRED_ENTITY_MOVE_REASON
```

Transferred entity move reason.

---

## TRANSFERRED\_ENTITY\_MOVE\_AGENT

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
TRANSFERRED_ENTITY_MOVE_AGENT
```

Transferred entity move agent.

---

(continued from last page)

## TRANSFERRED\_ENTITY\_MOVE\_DATETIME

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
TRANSFERRED_ENTITY_MOVE_DATETIME
```

---

## TRANSFERRED\_ENTITY\_MOVE\_CLASSIFICATION\_CODE

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
TRANSFERRED_ENTITY_MOVE_CLASSIFICATION_CODE
```

Transferred entity move classification code.

---

## PHYSICAL\_ENTITY\_IDENTIFIER

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
PHYSICAL_ENTITY_IDENTIFIER
```

Physical entity identifier.

---

## PHYSICAL\_ENTITY\_DESCRIPTION

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
PHYSICAL_ENTITY_DESCRIPTION
```

Physical entity description.

---

## PHYSICAL\_ENTITY\_STATUS

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
PHYSICAL_ENTITY_STATUS
```

Physical entity status.

---

## PHYSICAL\_ENTITY\_STATUS\_CHANGE\_DATETIME

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
PHYSICAL_ENTITY_STATUS_CHANGE_DATETIME
```

Physical entity status changed date and time.

---

## PHYSICAL\_ENTITY\_HOME\_LOCATION

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
PHYSICAL_ENTITY_HOME_LOCATION
```

Physical entity home location.

---

## PHYSICAL\_ENTITY\_CURRENT\_LOCATION

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
PHYSICAL_ENTITY_CURRENT_LOCATION
```

Physical entity current location.

(continued from last page)

---

## PHYSICAL\_ENTITY\_CUSTODIAN

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
PHYSICAL_ENTITY_CUSTODIAN
```

Physical entity custodian.

---

## PHYSICAL\_ENTITY\_RETURN\_DUE\_DATETIME

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
PHYSICAL_ENTITY_RETURN_DUE_DATETIME
```

Physical entity return due date and time.

---

## EMAIL\_ENTITY\_MESSAGE\_ID

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
EMAIL_ENTITY_MESSAGE_ID
```

Email entity message identifier.

---

## EMAIL\_ENTITY\_DATETIME

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
EMAIL_ENTITY_DATETIME
```

Email entity date and time.

---

## EMAIL\_ENTITY\_FROM

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
EMAIL_ENTITY_FROM
```

Email entity sender.

---

## EMAIL\_ENTITY\_TO

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
EMAIL_ENTITY_TO
```

Email entity recipient.

---

## EMAIL\_ENTITY\_TOCC

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
EMAIL_ENTITY_TOCC
```

Email entity recipients of copy.

---

## EMAIL\_ENTITY\_TOBCC

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
EMAIL_ENTITY_TOBCC
```

Email entity concealed recipients of copy.

---

(continued from last page)

---

## EMAIL\_ENTITY\_SUBJECT

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
EMAIL_ENTITY_SUBJECT
```

Email entity subject.

---

## EMAIL\_ENTITY\_PRIORITY

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
EMAIL_ENTITY_PRIORITY
```

Email entity delivery priority.

---

## EMAIL\_ENTITY\_IS\_SIGNED

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
EMAIL_ENTITY_IS_SIGNED
```

Email entity is signed value.

---

## RETENTION\_POLICY\_TRIGGER

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
RETENTION_POLICY_TRIGGER
```

Retention policy trigger.

---

## RETENTION\_POLICY\_ACTION

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
RETENTION_POLICY_ACTION
```

Retention policy action.

---

## RETENTION\_POLICY\_REASON

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
RETENTION_POLICY_REASON
```

Retention policy reason.

---

## RETENTION\_POLICY\_DETAILED\_DESCRIPTION

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
RETENTION_POLICY_DETAILED_DESCRIPTION
```

Retention policy detailed description.

---

## RETENTION\_POLICY\_REFERENCE

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
RETENTION_POLICY_REFERENCE
```

Retention policy reference.

---



(continued from last page)

---

## DISPOSITION\_HOLD\_REASON

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
DISPOSITION_HOLD_REASON
```

Retention policy reason.

---

## REVIEW\_MEMBERS

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
REVIEW_MEMBERS
```

Review members.

---

## REVIEW\_SCOPE

```
public static final com.imis.storageconnector.entitymodel.SystemProperty REVIEW_SCOPE
```

Review scope entity identifier.

---

## REVIEW\_COMMENTS

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
REVIEW_COMMENTS
```

Review comments.

---

## REVIEW\_SCHEDULES

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
REVIEW_SCHEDULES
```

Review schedules.

---

## REVIEW\_QUERY\_EXPRESSION

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
REVIEW_QUERY_EXPRESSION
```

Review query expression.

---

## REVIEW\_ACTION

```
public static final com.imis.storageconnector.entitymodel.SystemProperty REVIEW_ACTION
```

Review action.

---

## REVIEW\_STATE

```
public static final com.imis.storageconnector.entitymodel.SystemProperty REVIEW_STATE
```

Review state.

---

## REVIEW\_MESSAGE

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
REVIEW_MESSAGE
```

Review message.

---

---

## REVIEW\_SCHEDULED\_ITEMS

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
REVIEW_SCHEDULED_ITEMS
```

Review entity lists.

---

## VERSION\_SERIES\_ID

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
VERSION_SERIES_ID
```

Version series identifier.

---

## VERSIONING\_LABEL

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
VERSIONING_LABEL
```

Versioning series label.

---

## VERSIONING\_COMMENT

```
public static final com.imis.storageconnector.entitymodel.SystemProperty  
VERSIONING_COMMENT
```

Versioning check in comment.

---

## Methods

### values

```
public final static SystemProperty\[\] values()
```

---

### valueOf

```
public static SystemProperty valueOf(String name)
```

---

**Package**

**com.imis.storageconnector.services**

## com.imis.storageconnector.services

# Class DirectoryEntityType

```

java.lang.Object
  |
  +- java.lang.Enum
        +- com.imis.storageconnector.services.DirectoryEntityType

```

**All Implemented Interfaces:**  
 Serializable, Comparable

public final class **DirectoryEntityType**  
 extends Enum

Specifies the type of a IDirectoryEntity.

UNKNOWN directory entity type represents an invalid directory entity which is never returned on a coherent directory entity.

**See Also:**

[IDirectoryEntity](#)

## Field Summary

public static final	<a href="#">GROUP</a> Represents a group entity.
public static final	<a href="#">UNKNOWN</a> Represents an unknown entity.
public static final	<a href="#">USER</a> Represents a user entity.

## Method Summary

static <a href="#">DirectoryEntityType</a>	<a href="#">valueOf</a> (String name)
static <a href="#">DirectoryEntityType[]</a>	<a href="#">values</a> ()

### Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods inherited from interface java.lang.Comparable

compareTo

(continued from last page)

## Fields

### UNKNOWN

```
public static final com.imis.storageconnector.services.DirectoryEntityType UNKNOWN
```

Represents an unknown entity.

---

### USER

```
public static final com.imis.storageconnector.services.DirectoryEntityType USER
```

Represents a user entity.

---

### GROUP

```
public static final com.imis.storageconnector.services.DirectoryEntityType GROUP
```

Represents a group entity.

---

## Methods

### values

```
public final static DirectoryEntityType\[\] values()
```

---

### valueOf

```
public static DirectoryEntityType valueOf(String name)
```

## com.imis.storageconnector.services Class Discovery

java.lang.Object

└-com.imis.storageconnector.services.Discovery

public class **Discovery**  
extends Object

Provides discovery services such as archive discovery, etc.

See Also:

[IArchive](#)

### Constructor Summary

public	<a href="#">Discovery()</a>
--------	-----------------------------

### Method Summary

static void	<a href="#">addConnectingListener</a> ( <a href="#">IConnectingListener</a> listener) Adds an <a href="#">IConnectingListener</a> object.
static Collection	<a href="#">findArchives</a> (Map options) Gets all available archives.
static void	<a href="#">removeConnectingListener</a> ( <a href="#">IConnectingListener</a> listener) Removes an <a href="#">IConnectingListener</a> object.

#### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructors

### Discovery

public **Discovery**()

## Methods

### addConnectingListener

public static void **addConnectingListener**([IConnectingListener](#) listener)

Adds an [IConnectingListener](#) object.

#### Parameters:

listener - an IConnectingListener.

## removeConnectingListener

```
public static void removeConnectingListener(IConnectingListener listener)
```

Removes an [IConnectingListener](#) object.

**Parameters:**

listener - an IConnectingListener.

---

## findArchives

```
public static Collection findArchives(Map options)  
    throws StorageConnectorException,  
           IOException
```

Gets all available archives.

**Parameters:**

options - an archive options.

**Returns:**

A collection of [IArchiveDescriptor](#) objects.

**Throws:**

[StorageConnectorException](#) - if

- None of the specified discovery archive type is supported.
- Discovery hint is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while getting available archives on server.

IOException - if an I/O error occurs.

## com.imis.storageconnector.services Interface IArchiveDescriptor

public interface **IArchiveDescriptor**  
extends

Provides discovery services such as archive discovery, etc.

**See Also:**

[IArchive](#)

### Method Summary

String	<a href="#">getDescription()</a> Gets the archive description.
String	<a href="#">getHost()</a> Gets the archive information as host name or IP address and port separated by colon (:).
String	<a href="#">getId()</a> Gets the archive identifier.
String	<a href="#">getName()</a> Gets the archive name.
<a href="#">ArchiveType</a>	<a href="#">getType()</a> Gets the archive type.
<a href="#">IArchive</a>	<a href="#">open</a> (Map options) Opens an archive using an archive login and session options.

### Methods

#### getId

public String **getId()**

Gets the archive identifier.

**Returns:**

The archive identifier.

#### getType

public [ArchiveType](#) **getType()**

Gets the archive type.

**Returns:**

The archive type.

#### getName

public String **getName()**



(continued from last page)

Gets the archive name.

**Returns:**

The archive name.

---

## getDescription

```
public String getDescription()
```

Gets the archive description.

**Returns:**

The archive description.

---

## getHost

```
public String getHost()
```

Gets the archive information as host name or IP address and port separated by colon (:).

**Returns:**

A archive host name or IP address and port.

---

## open

```
public IArchive open(Map options)  
throws StorageConnectorException
```

Opens an archive using an archive login and session options.

**Parameters:**

`options` - an archive login and session options.

**Returns:**

An [IArchive](#) instance.

---

## com.imis.storageconnector.services Interface IDirectory

public interface **IDirectory**  
extends

Provides access to different directory services such as directory membership lookups, security related methods, etc.

**See Also:**

[IArchive](#)

### Method Summary

com.imis.util.IReadOnlyMap	<a href="#">getGroupMembers</a> (String groupName) Queries the IDirectory service for a collection of <a href="#">IDirectoryEntity</a> objects that are members of a specified group.
<a href="#">IDirectoryEntity</a>	<a href="#">getInvalidDirectoryEntity</a> () Gets the invalid <a href="#">IDirectoryEntity</a> .
com.imis.util.IReadOnlyMap	<a href="#">getMembers</a> () Queries the IDirectory service for a collection of all <a href="#">IDirectoryEntity</a> objects.
<a href="#">IUserProfile</a>	<a href="#">getUserProfile</a> () Gets the user profile.
com.imis.util.IReadOnlyList	<a href="#">search</a> (String query) Searches for directory entities in the IDirectory service using specified search query.

### Methods

#### getInvalidDirectoryEntity

```
public IDirectoryEntity getInvalidDirectoryEntity()
```

Gets the invalid [IDirectoryEntity](#).

**Returns:**

The invalid IDirectoryEntity.

#### getMembers

```
public com.imis.util.IReadOnlyMap getMembers()  
throws StorageConnectorException
```

Queries the IDirectory service for a collection of all [IDirectoryEntity](#) objects.

**Returns:**

A read-only map of IDirectoryEntity objects sorted by directory entity subjects.

**Throws:**

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from session pool.
- Error occurred while getting directory members on archive.

---

## search

```
public com.imis.util.IReadOnlyList search(String query)
    throws StorageConnectorException
```

Searches for directory entities in the IDirectory service using specified search query.

### Parameters:

query - the search query.

### Returns:

A read-only list of IDirectoryEntity objects.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from session pool.
- Error occurred while getting directory members on archive.

---

## getGroupMembers

```
public com.imis.util.IReadOnlyMap getGroupMembers(String groupName)
    throws StorageConnectorException
```

Queries the IDirectory service for a collection of [IDirectoryEntity](#) objects that are members of a specified group.

### Parameters:

groupName - the name of the group of directory entities.

### Returns:

A read-only map of IDirectoryEntity objects that are members of a specified group sorted by directory entity subjects.

### Throws:

NullPointerException - if groupName is a null reference.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Group name is required and cannot be empty.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from session pool.
- Error occurred while getting directory group members on archive.

## getUserProfile

```
public IUserProfile getUserProfile()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the user profile.

### Returns:

The `IUserProfile` object or a null reference.

### Throws:

`CancellationException` - if an operation was canceled by the user.

`LoginException` - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from session pool.
- Error occurred while getting user profile on archive.

`IOException` - if an I/O error occurs.

## com.imis.storageconnector.services Interface IDirectoryEntity

public interface **IDirectoryEntity**  
extends

Represents a directory entity, a group or a person registered in the IDirectory.

**See Also:**

[IDirectory](#)

### Method Summary

String	<a href="#">getDescription()</a> Gets the directory entity description.
String	<a href="#">getEmail()</a> Gets the directory entity email address.
String	<a href="#">getFirstName()</a> Gets the directory entity first (given) name.
<a href="#">IIcon</a>	<a href="#">getIcon()</a> Gets the directory entity icon.
String	<a href="#">getLastName()</a> Gets the directory entity last (surname) name.
String	<a href="#">getSecurityClass()</a> Gets the directory entity security class.
String	<a href="#">getSubject()</a> Gets the directory entity subject.
<a href="#">DirectoryEntityType</a>	<a href="#">getType()</a> Gets the directory entity type.
boolean	<a href="#">isDeleted()</a> Gets the value indicating whether or not the directory entity is deleted.
boolean	<a href="#">isEnabled()</a> Gets the value indicating whether or not the directory entity is enabled.

### Methods

#### getType

public [DirectoryEntityType](#) **getType()**

Gets the directory entity type.

**Returns:**

The directory entity type.

---

## getSubject

```
public String getSubject()
```

Gets the directory entity subject.

The subject represents a `IDirectoryEntity` unique identifier.

**Returns:**

The directory entity subject.

---

## getFirstName

```
public String getFirstName()
```

Gets the directory entity first (given) name.

The directory entity first (given) name is empty in case the entity is a group.

**Returns:**

A string that represents the directory entity first name.

---

## getLastName

```
public String getLastName()
```

Gets the directory entity last (surname) name.

The directory entity last (surname) name is empty in case the entity is a group.

**Returns:**

A string that represents the directory entity last name.

---

## getEmail

```
public String getEmail()
```

Gets the directory entity email address.

**Returns:**

The directory entity email address.

---

## getDescription

```
public String getDescription()
```

Gets the directory entity description.

**Returns:**

A string that represents the directory entity description.

---

## getSecurityClass

```
public String getSecurityClass()  
throws StorageConnectorException
```

Gets the directory entity security class.

The directory entity security class is undefined in case the entity is a group.

---

(continued from last page)

**Returns:**

The directory entity security class.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting templates on archive.

---

## getIcon

```
public IIcon getIcon()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Gets the directory entity icon.

**Returns:**

The directory entity icon or a null reference if no icon is set.

**Throws:**

[CancellationException](#) - if an operation was canceled by the user.

[LoginException](#) - if a login error occurs.

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from session pool.
- Error occurred while getting directory members on archive.

[IOException](#) - if an I/O error occurs.

---

## isEnabled

```
public boolean isEnabled()
```

Gets the value indicating whether or not the directory entity is enabled.

**Returns:**

true if the directory entity is enabled; otherwise false.

---

## isDeleted

```
public boolean isDeleted()
```

Gets the value indicating whether or not the directory entity is deleted.

**Returns:**

true if the directory entity is deleted; otherwise false.

## com.imis.storageconnector.services Interface IIcon

---

public interface **IIcon**  
extends

Represents an icon.

---

### Method Summary

String	<a href="#">getContentType()</a> Gets the icon content type.
byte[]	<a href="#">getData()</a> Gets the icon data.

---

### Methods

#### getContentType

public String **getContentType()**

Gets the icon content type.

**Returns:**

The icon content type or a null reference if the icon was changed.

---

#### getData

public byte[] **getData()**

Gets the icon data.

**Returns:**

The array of bytes containing icon data.



## com.imis.storageconnector.services Interface IUserProfile

public interface **IUserProfile**  
extends

Represents a profile owned by the authenticated user on the IArchive.

### Method Summary

void	<a href="#"><u>changePassword</u></a> (String oldPassword, String newPassword) Changes the user profile password.
String	<a href="#"><u>getDescription</u></a> () Gets the user profile description.
String	<a href="#"><u>getEmail</u></a> () Gets the user profile email address.
String	<a href="#"><u>getFirstName</u></a> () Gets the user profile first (given) name.
<a href="#"><u>IIcon</u></a>	<a href="#"><u>getIcon</u></a> () Gets the user profile icon.
String	<a href="#"><u>getLastName</u></a> () Gets the user profile last (surname) name.
String	<a href="#"><u>getSecurityClass</u></a> () Gets the user profile security class.
String	<a href="#"><u>getSubject</u></a> () Gets the user profile subject.
boolean	<a href="#"><u>isDescriptionEditable</u></a> () Gets the value indicating whether or not the user profile description is editable.
boolean	<a href="#"><u>isEmailEditable</u></a> () Gets the value indicating whether or not the user profile email address is editable.
boolean	<a href="#"><u>isFirstNameEditable</u></a> () Gets the value indicating whether or not the user profile first name is editable.
boolean	<a href="#"><u>isIconEditable</u></a> () Gets the value indicating whether or not the user profile icon is editable.
boolean	<a href="#"><u>isLastNameEditable</u></a> () Gets the value indicating whether or not the user profile last name is editable.
void	<a href="#"><u>save</u></a> () Saves the user profile changes.
void	<a href="#"><u>setDescription</u></a> (String value) Sets the user profile description.

void	<a href="#"><code>setEmail</code></a> (String value) Sets the user profile email address.
void	<a href="#"><code>setFirstName</code></a> (String value) Sets the user profile first (given) name.
void	<a href="#"><code>setIconData</code></a> (byte[] data) Sets the user profile icon data.
void	<a href="#"><code>setLastName</code></a> (String value) Sets the user profile last (surname) name.

---

## Methods

### **getSubject**

```
public String getSubject()
```

Gets the user profile subject.

**Returns:**

The user profile subject.

---

### **getFirstName**

```
public String getFirstName()
```

Gets the user profile first (given) name.

**Returns:**

The user profile first name.

---

### **setFirstName**

```
public void setFirstName(String value)  
throws StorageConnectorException
```

Sets the user profile first (given) name.

**Parameters:**

value - the user profile first name.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on non editable field is not allowed.
- 

### **getLastName**

```
public String getLastName()
```

Gets the user profile last (surname) name.

**Returns:**

The user profile last name.

---

---

## setLastName

```
public void setLastName(String value)  
    throws StorageConnectorException
```

Sets the user profile last (surname) name.

**Parameters:**

value - the user profile last name.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on non editable field is not allowed.

---

## getEmail

```
public String getEmail()
```

Gets the user profile email address.

**Returns:**

The user profile email address.

---

## setEmail

```
public void setEmail(String value)  
    throws StorageConnectorException
```

Sets the user profile email address.

**Parameters:**

value - the user profile email address.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on non editable field is not allowed.

---

## getDescription

```
public String getDescription()
```

Gets the user profile description.

**Returns:**

The user profile description.

---

## setDescription

```
public void setDescription(String value)  
    throws StorageConnectorException
```

Sets the user profile description.

(continued from last page)

**Parameters:**

value - the user profile description.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on non editable field is not allowed.

---

## getSecurityClass

```
public String getSecurityClass()  
    throws StorageConnectorException
```

Gets the user profile security class.

**Returns:**

The user profile security class.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while opening a session on server.
- Error occurred while authenticating a session on server.
- Error getting session from the session pool.
- Error occurred while getting templates on archive.

---

## getIcon

```
public IIcon getIcon()
```

Gets the user profile icon.

**Returns:**

The user profile icon or a null reference if no icon is set or icon was removed.

---

## setIconData

```
public void setIconData(byte[] data)  
    throws StorageConnectorException
```

Sets the user profile icon data.

**Parameters:**

data - the array of bytes containing user profile icon data or a null reference to remove the icon.

**Throws:**

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on non editable field is not allowed.

---

## isFirstNameEditable

```
public boolean isFirstNameEditable()
```

(continued from last page)

Gets the value indicating whether or not the user profile first name is editable.

**Returns:**

true if the user profile first name is editable; otherwise false.

---

## isLastNameEditable

```
public boolean isLastNameEditable()
```

Gets the value indicating whether or not the user profile last name is editable.

**Returns:**

true if the user profile last name is editable; otherwise false.

---

## isEmailEditable

```
public boolean isEmailEditable()
```

Gets the value indicating whether or not the user profile email address is editable.

**Returns:**

true if the user profile email address is editable; otherwise false.

---

## isDescriptionEditable

```
public boolean isDescriptionEditable()
```

Gets the value indicating whether or not the user profile description is editable.

**Returns:**

true if the user profile description is editable; otherwise false.

---

## isIconEditable

```
public boolean isIconEditable()
```

Gets the value indicating whether or not the user profile icon is editable.

**Returns:**

true if the user profile icon is editable; otherwise false.

---

## changePassword

```
public void changePassword(String oldPassword,  
                           String newPassword)  
throws StorageConnectorException
```

Changes the user profile password.

**Parameters:**

oldPassword - the old password.  
newPassword - the new password.

**Throws:**

NullPointerException - if oldPassword or newPassword is a null reference.

---

(continued from last page)

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Password is required and cannot be empty.

---

## save

```
public void save()  
    throws javax.security.auth.login.LoginException,  
           StorageConnectorException,  
           IOException
```

Saves the user profile changes.

### Throws:

[StorageConnectorException](#) - if an IMiS/Storage Connector error occurs:

- Operation on closed archive is not allowed.
- Error occurred while committing user profile changes on archive.

IOException - if an I/O error occurs.

# Index

## A

accessed 261, 273, 284, 298, 322, 371, 451, 471, 487  
ACLFILTERITEM 214  
acquire 69, 111  
acquireSession 112  
addAuthenticatingListener 75, 128  
addConnectingListener 75, 128, 542  
addDispositionHoldEntry 446  
addEntry 246  
addHandler 205  
addObjectRef 192  
addPolicyEntry 445, 446  
addPropertyEntry 246  
ADDRESSES 22  
addSession 111  
AES 31  
ALL 222  
ALLOW 521  
ANY 242  
areValuesInherited 291, 300, 340, 380, 433, 479, 490  
ASCENDING 239  
ATTRIBUTE 213  
AUDIT\_QUERY 4  
AuditLog 7  
auditLogQuery 104, 144, 145  
AuditQuery 12  
authenticate 79  
authenticating 115  
AuthenticatingEvent 37

## B

BASE16 120, 217, 226  
BASE64 120, 217, 227  
BASE85 120, 217, 227  
BINARY 516  
BOOL 511

## C

cancel 38, 46  
CATEGORY 531

CBC 34  
CFB 34  
CHANGE\_RETENTION 236  
CHANGE\_RIGHTS 235  
CHANGE\_SECURITY\_CLASS 236  
CHANGE\_STATUS 236  
changeEntityTemplate 86  
changePassword 557  
changeRetention 426  
changeRights 425  
changeSecurityClass 426  
changeStatus 426  
checkIn 302  
checkInDraft 89  
checkOut 384  
checkOutEntity 88  
CLASS 242  
CLASSES 29, 223, 243, 519  
CLASSIFICATION\_CODE 229, 529  
cleanSessions 111  
clear 9, 258, 406  
clone 61, 263  
cloneParts 255  
close 63, 69, 108, 113, 145, 292, 342, 357, 447, 482  
CLOSED 530  
closed 322, 371, 472, 487  
closeSessions 111  
committedValueCount 401  
COMPUTER\_NAMES 23  
connecting 116  
ConnectingEvent 43  
CONTENT 530  
CONTENT\_MANAGEMENT 4  
CONTENT\_MANAGER 207  
ContentTypeMapper 52  
CREATE\_REFERENCES 236  
CREATE\_SUB\_ENTITIES 235  
createAccess 440  
createBinaryValue 290, 337  
createClass 81, 251  
createContentPart 277, 293  
created 261, 272, 284, 297, 321, 353, 370, 419, 450, 458, 471, 486  
createDocument 83, 252, 388, 480  
createEntity 80

createEntityReference 79  
createFolder 82, 252, 387  
createObject 131  
createReferences 426  
createReview 100, 101  
createStringMaxValue 289, 336  
createSubEntities 425  
CREATOR 530

## D

DATE 513  
DATE\_TIME 513  
DECIMAL1 515  
DECIMAL10 516  
DECIMAL2 515  
DECIMAL3 515  
DECIMAL4 515  
DECIMAL5 515  
DECIMAL6 515  
DECIMAL7 516  
DECIMAL8 516  
DECIMAL9 516  
DEFAULT\_PAGE\_COUNT 174  
DEFAULT\_PAGE\_SIZE 173  
delete 62, 274, 347, 364, 383, 384, 410, 414, 421, 456  
DELETE\_ACCESS 235  
deleteAccess 425, 440  
DELETED 220  
deleted 268  
DELETED\_ENTITY\_AGENT 532  
DELETED\_ENTITY\_CLASSIFICATION\_CODE 532  
DELETED\_ENTITY\_DELETE\_DATETIME 532  
DELETED\_ENTITY\_REASON 532  
DELETED\_ENTITY\_REFERENCE 532  
deleteEntity 87, 90  
deleteObject 134  
DENY 522  
DESCENDING 240  
DESCRIPTION 529  
detach 264  
detachParts 256  
DIRECTORY\_ENTITY 213, 511  
discard 302  
discardDraft 89

Discovery 542  
DISPOSED 221  
DISPOSITION\_HOLD\_REASON 536  
DOCUMENT 242  
Document 56  
DOCUMENTS 30, 223, 244, 520  
DOMINO\_DOC 207  
DOUBLE 513  
DRAFT 233  
DRAFT\_MANAGEMENT 4

## E

ECB 34  
edit 421  
EMAIL\_ENTITY\_DATETIME 535  
EMAIL\_ENTITY\_FROM 535  
EMAIL\_ENTITY\_IS\_SIGNED 536  
EMAIL\_ENTITY\_MESSAGE\_ID 535  
EMAIL\_ENTITY\_PRIORITY 536  
EMAIL\_ENTITY\_SUBJECT 535  
EMAIL\_ENTITY\_TO 535  
EMAIL\_ENTITY\_TOBCC 535  
EMAIL\_ENTITY\_TOCC 535  
ENTITY\_MOVE 518  
ENTITY\_REFERENCE 515  
ENTITY\_SAVE 518  
ENTITY\_SECURITY\_CLASS 518  
ENTITY\_STATUS 518  
entityCount 500  
EntityFilter 223  
EntityFilterItem 225  
EntitySortKey 237  
EVENT\_IDS 25  
EXPORT 508  
EXTERNAL 122, 229  
EXTERNAL\_IDS 530

## F

FILE\_SYSTEM 207  
finalize 189, 204  
findArchives 543  
FOLDER 242  
FOLDERS 30, 223, 244, 520



freeInstance 189  
fromBackEndAuthenticationOptions 191  
fromBackEndAuthenticationType 191  
fromLegacyContentType 52

## G

getAcceptableIssuers 45  
getAccessibleCount 349  
getAccessMode 57  
getACL 285, 326, 355, 473  
getAction 451, 459, 468  
getAddresses 13  
getAddressFrom 13  
getAddressTo 14  
getAgent 268, 389, 494  
getArchivalInformationalPackage 331, 475  
getArchive 327, 374, 473, 488  
getArguments 8  
getAuditLog 77, 130, 331, 475, 503  
getAuthenticationType 45  
getAutoSave 58  
getBCC 308  
getCapacity 129  
getCategories 288, 299, 333, 349, 376  
getCategory 349  
getCC 307  
getClassificationCode 267, 316, 368, 389, 464, 485, 504  
getComments 470  
getComputerNameFrom 15  
getComputerNames 15  
getComputerNameTo 16  
getContent 276, 293, 300, 377, 501  
getContentDescriptions 493  
getContentType 118, 247, 261, 552  
getContentTypeResolver 192  
getContentTypeTypes 118  
getContext 68, 110  
getCreated 59  
getCreator 272, 285, 298, 328, 354, 375, 419, 450, 458, 473, 488  
getCurrentLocation 395  
getCustodian 395  
getCustomContents 277, 293, 300, 377, 501  
getCustomContentTypeResolver 192  
getData 552  
getDate 305  
getDateFrom 17  
getDateTo 18  
getDefaultExtension 56  
getDefaultMime 56  
getDefinition 274, 455  
getDeletedEntities 97  
getDeletedEntity 96  
getDescription 260, 268, 272, 283, 297, 318, 353, 369, 392, 419, 430, 450, 458, 466, 485, 500, 545, 550, 555  
getDetailedDescription 452, 459  
getDirection 238  
getDirectory 76  
getDispositionHoldEntries 445  
getDispositionHolds 78  
getDocuments 478  
getDraftCreator 324, 372  
getDraftId 323, 372  
getDraftInfo 107, 372  
getDrafts 97  
getEffectiveRights 103, 255, 285, 298, 326, 346, 354, 374, 409, 416, 420, 433, 473, 488  
getEmail 550, 555  
getEmailEntity 286, 330  
getEntityCollectionPageCount 77  
getEntityCollectionPageSize 77  
getEntityInfo 91, 92, 93  
getEntityReference 79  
getEntityType 500  
getEntries 245  
getEventIdFrom 17  
getEventIdQueryType 16  
getEventIds 17  
getEventIdTo 17  
getEvidence 505  
getEvidenceRecords 332, 476  
getExplicitAllowRights 346, 409  
getExplicitContext 456  
getExplicitDenyRights 346, 409  
getExtension 117, 261  
getExtensions 56, 117  
getExternalIds 333, 376  
getFilter 18, 442  
getFirstName 550, 554

getFrom 306  
getGroupMembers 547  
getHomeLocation 394  
getHost 76, 545  
getIcon 551, 556  
getId 59, 68, 260, 266, 271, 282, 296, 316, 352, 368, 392, 418, 449, 464, 484, 500, 544  
getInheritedAllowRights 345, 408  
getInheritedContext 455  
getInheritedDenyRights 345, 408  
getInputStream 60  
getInstance 113, 189, 204  
getInternalId 106  
getInternalLogging 204  
getInvalidDirectoryEntity 546  
getItem 223  
getKey 398, 399  
getKeyTypes 45  
getKeywords 287, 299, 332, 376, 476, 489  
getLabel 430, 500  
getLastAccessed 60  
getLastName 550, 554  
getLocalCertificate 46  
getLocalCertificateKey 46  
getLocalSocketAddress 43  
getLogEntities 98  
getMandate 452, 460  
getMembers 469, 546  
getMessage 7, 468, 486  
getMessageId 305  
getMimes 57  
getMode 320, 444, 467  
getModified 59  
getModifiedBy 284, 297, 321, 354, 370, 420, 472, 487  
getMoveAgents 505  
getMoveClassificationCodes 506  
getMoveDateTimes 506  
getMoveDetails 329  
getMoveReasons 505  
getName 271, 415, 429, 450, 457, 544  
getObjectIds 16  
getObjectScope 16  
getOptions 38  
getOriginal 460  
getOutputStream 60  
getOwner 68, 286, 298, 328, 344, 375, 407, 424, 440, 453, 474, 488  
getParent 262, 327, 375  
getParts 416  
getPasswordCallback 44  
getPersistent 110  
getPhysicalEntity 287, 330  
getPickListValues 291, 301, 340, 381, 433, 479, 490, 502  
getPolicyEntries 445  
getPriority 309  
getProfileCapacity 143  
getProfileObjectCount 144  
getProfileSpaceAvailable 143  
getProfileSpaceUsed 144  
getProperties 76, 289, 299, 334, 377, 501  
getPropertyEntries 245  
getPropertyList 501  
getPropertyName 238, 408  
getProtocol 43  
getPublicClassificationCode 105, 267, 317, 368, 389, 465, 485  
getQueryExpression 469  
getReason 269, 273, 390, 451, 459, 494  
getReference 269  
getReferenceIds 355, 420  
getReferences 335, 378  
getRemoteCertificateChain 45  
getRemoteHost 43  
getRemoteSocketAddress 43  
getReport 105, 341, 382, 480, 491  
getRetention 336, 378  
getRetentionPolicies 78, 478  
getReviewedItems 477  
getReviewInfo 102  
getReviews 102  
getRootClasses 95  
getSaveLog 329, 474  
getScheduledItems 477  
getScope 442, 469  
getSecurityClass 326, 374, 550, 556  
getSecurityClassChangeDetails 329  
getSecurityClassFrom 495  
getSecurityClassTo 495  
getSession 111  
getSessionPool 113  
getSessionQueryType 13

getSessionScope 13  
getSignificance 319, 369  
getSize 59, 261  
getSortOrder 18  
getSourceId 283, 297  
getSpaceAvailable 129  
getSpaceUsed 130  
getSslProtocols 44  
getStage 43  
getState 467, 486  
getStatus 319, 370, 393, 466, 485  
getStore 57  
getStoreInfo 129  
getStoreType 129  
getStub 316, 464  
getSubEntities 95, 269, 339, 379  
getSubEntityTemplates 335, 377, 477, 489  
getSubject 309, 345, 408, 550, 554  
getSubjectType 345, 408  
getSubParts 262  
getSystemId 504  
getSystemProperties 289, 299, 334, 376, 478, 490, 502  
getSystemProperty 290, 300, 339, 380, 479, 490, 502  
getSystemPropertyName 105  
getSystemPropertySortKey 106  
getTemplate 287, 299, 331, 375, 475, 488  
getTemplates 78, 348  
getTimeStamp 68  
getTitle 267, 282, 296, 317, 352, 369, 419, 465, 485  
getTo 307  
getTransferDetails 330  
getTrigger 451, 458  
getTrustAnchors 44  
getType 38, 223, 320, 370, 430, 544, 549  
getUser 76  
getUserNameFrom 14  
getUserNames 14  
getUserNameTo 15  
getUserProfile 548  
getValue 247, 398, 433, 434, 435, 496  
getValues 436, 437  
getVersionComment 325, 373  
getVersionInfo 194  
getVersionLabel 324, 373  
getVersions 374

getVersionSequence 324, 373  
getXmlValue 435  
getXmlValues 437  
GROUP 541

## H

hasAIP 323, 371, 482, 492  
hasDraft 323, 372  
hasReferences 335, 377  
hasVersions 324, 373

## I

IACSession 67  
IASessionPool 110  
ID192 123, 218, 230  
ID256 124, 219, 231  
IMIS\_ARC 207  
IMIS\_ARCHIVE 6  
IMPORT 507  
IMPORT\_EXPORT 4  
imported 504  
INHERITED\_VALUES 524  
INT128 513  
INT16 512  
INT32 512  
INT64 512  
INT8 512  
INTERNAL 122, 229  
isAppendOnly 432  
isChildClassificationCodeGenerated 251, 369, 386  
isClosed 57, 76, 128, 325, 472  
isContainer 267  
isDefault 398  
isDeleted 551  
isDescriptionEditable 557  
isEffective 442  
isEmailArchivalCapable 500  
isEmailEditable 557  
isEnabled 551  
isFirstNameEditable 556  
isIconEditable 557  
isIncludedInAIP 432  
isIndexed 262

isInherited 274, 432  
isLastNameEditable 557  
isModified 58  
isMultiValue 430  
isNew 58  
isNonEmpty 431  
isPermanent 322, 371  
isPickList 433  
isPropagated 426, 441  
isPublic 417, 430  
isReadOnly 416, 431  
isReadOnlyAfterCheckIn 416, 431  
isReadOnlyAfterCreate 416, 431  
isRecursive 19  
isRemoteCertificateAccepted 45  
isRequired 430  
isSearchable 432  
isSigned 262, 310  
isTruncated 350  
isUnique 432  
isValid 19  
isVersionable 417, 431, 501

## K

KERBEROS5\_SERVICE\_TICKET 40  
KEYWORDS 530  
KS128 36  
KS192 36  
KS256 36

## L

LDAP\_CREDENTIALS 40  
LIST 20  
LNDI\_MSS 207  
LOCAL\_CERTIFICATE\_SELECTION 49  
LOCAL\_CREDENTIALS 40  
lock 69  
logAddHandler 192  
logGetLevel 193  
logInternal 193  
logIsInternal 193  
logRemoveHandler 192  
logSetLevel 193

## M

MAX\_OPTION\_NAME\_LENGTH 155  
modified 261, 272, 284, 297, 321, 354, 370, 420, 451, 471, 487, 494  
MODIFIED\_BY 530  
move 61, 264, 383  
MOVE\_ACCESS 235  
moveAccess 425  
moved 390  
MOVED\_ENTITY\_AGENT 532  
MOVED\_ENTITY\_CLASSIFICATION\_CODE 532  
MOVED\_ENTITY\_MOVE\_DATETIME 532  
MOVED\_ENTITY\_REASON 532  
moveEntity 87  
moveObject 133  
moveParts 256, 257

## O

OBJECT\_IDS 24  
OBJECT\_SESSION\_TIMESTAMP 27  
OBJECT\_TIMESTAMP\_SESSION 27  
objectCount 130  
objectTransfer 198  
ODMA 207  
OFB 34  
open 301, 382, 491, 545  
openArchive 198  
openClass 84  
openContentManagerStorage 197  
openDataInputStream 248, 262, 497  
openDataOutputStream 248, 263, 497  
openDocument 85  
openDocumentManagerStorage 197  
openDraft 107  
OPENED 530  
opened 322, 371, 472, 487  
openEntity 83  
openFileSystemStorage 197  
openFolder 85  
openIMiSARCStorage 194, 195, 196  
openObject 132  
openReview 104

OPTION\_APPLICATION\_NAME 157  
OPTION\_ARCHIVE\_ID 174  
OPTION\_AUTH\_CRYPTO\_ALGORITHM 163  
OPTION\_AUTH\_CRYPTO\_CIPHERMODE 164  
OPTION\_AUTH\_CRYPTO\_KEYSIZE 162  
OPTION\_AUTH\_KEY 160  
OPTION\_AUTH\_TYPE 159  
OPTION\_AUTH\_TYPE\_DISPOSITION 176  
OPTION\_COMPUTER\_NAME 156  
OPTION\_CONTENT\_PART\_IDENCODING 171  
OPTION\_CONTENT\_PART\_IDTYPE 170  
OPTION\_DEFERRED\_COLLECTION\_FILTERING 186  
OPTION\_DISCOVERY\_ARCHIVETYPES 176  
OPTION\_DISCOVERY\_HINT 175  
OPTION\_ENTITY\_IDENCODING 169  
OPTION\_ENTITY\_IDTYPE 168  
OPTION\_ENTITYCOLLECTION\_PAGECOUNT 173  
OPTION\_ENTITYCOLLECTION\_PAGESIZE 172  
OPTION\_IMPERSONALIZATION\_USER\_NAME 184  
OPTION\_KERBEROS\_KDC 177  
OPTION\_KERBEROS\_REALM 179  
OPTION\_KERBEROS\_SPN 178  
OPTION\_KERBEROS\_TICKET 179  
OPTION\_LOCAL\_ADDRESS 158  
OPTION\_LOCAL\_CERTIFICATE 180  
OPTION\_LOCAL\_CERTIFICATE\_KEY 181  
OPTION\_MAX\_IO\_PAYLOAD\_SIZE 185  
OPTION\_MAX\_SESSIONS\_PER\_USER 153  
OPTION\_NODES 154  
OPTION\_OBJECT\_IDENCODING 166  
OPTION\_OBJECT\_IDKIND 167  
OPTION\_OBJECT\_IDTYPE 165  
OPTION\_OUTBOUND\_ADDRESS 182  
OPTION\_OUTBOUND\_ADDRESS\_V6 183  
OPTION\_PERSISTENT\_SESSIONS 152  
OPTION\_USER\_NAME 155  
OPTION\_USER\_PASSWORD 161  
optionsToString 190  
OWNER 530

## P

PARENT\_INFO 233  
partsCount 416  
PHYSICAL\_ENTITY\_CURRENT\_LOCATION 534

PHYSICAL\_ENTITY\_CUSTODIAN 534  
PHYSICAL\_ENTITY\_DESCRIPTION 534  
PHYSICAL\_ENTITY\_HOME\_LOCATION 534  
PHYSICAL\_ENTITY\_IDENTIFIER 534  
PHYSICAL\_ENTITY\_RETURN\_DUE\_DATETIME 535  
PHYSICAL\_ENTITY\_STATUS 534  
PHYSICAL\_ENTITY\_STATUS\_CHANGE\_DATETIME 534  
PRE\_AUTHENTICATE 40  
PRE\_SHARED\_KEY 40  
PROPERTIES 233  
PROPERTY\_AUDIT\_LOG\_REQUIRED 187  
PROPERTY\_AUDIT\_LOG\_SUPPORTED 187  
PROPERTY\_AUTHENTICATED\_USER\_ACCOUNT 189  
PROPERTY\_CAPABILITIES 187  
PROPERTY\_CLASS\_HIERARCHY\_DELIMITER 188  
PROPERTY\_DOCUMENT\_HIERARCHY\_DELIMITER 188  
PROPERTY\_FOLDER\_HIERARCHY\_DELIMITER 188  
PROPERTY\_ROOT\_EFFECTIVE\_RIGHTS 188  
PROPERTY\_SERVICE\_NAME 188  
PROPERTY\_SERVICE\_VERSION 188  
PROPERTY\_TRAFFIC\_ENCRYPTED 189

## Q

queueForConvert 265  
queueForIndex 265  
queuePartsForConvert 257  
queuePartsForIndex 258

## R

RANGE 21  
READ 64, 210  
READ\_ACCESS 235  
READ\_WRITE 65, 211  
readAccess 424, 440  
readOptionDiscoveryArchiveTypes 190  
readOptionDiscoveryHint 190  
readOptionEntityCollectionPageCount 190  
readOptionEntityCollectionPageSize 190  
RECURSIVE 523  
REFERENCE 531  
REFERENCES 233  
release 69, 111  
releaseSession 112

releaseSessionPool 114  
REMOTE\_CERTIFICATE\_VALIDATION 49  
removeAuthenticatingListener 75, 128  
removeConnectingListener 75, 128, 543  
removeHandler 205  
REPORTS 4  
resolveExternalObjectId 142  
RETENTION\_POLICY\_ACTION 536  
RETENTION\_POLICY\_DETAILED\_DESCRIPTION 536  
RETENTION\_POLICY\_REASON 536  
RETENTION\_POLICY\_REFERENCE 536  
RETENTION\_POLICY\_TRIGGER 536  
retrieveObject 140, 141  
returnDue 396  
REVIEW\_ACTION 537  
REVIEW\_COMMENTS 537  
REVIEW\_MEMBERS 537  
REVIEW\_MESSAGE 537  
REVIEW\_QUERY\_EXPRESSION 537  
REVIEW\_SAVE 518  
REVIEW\_SCHEDULED\_ITEMS 538  
REVIEW\_SCHEDULES 537  
REVIEW\_SCOPE 537  
REVIEW\_STATE 537  
REVIEW\_STATUS 518

**S**

save 62, 292, 342, 356, 447, 481, 558  
SAVE\_LOG 530  
search 99, 100, 340, 381, 547  
SECURED 51  
SECURITY\_CLASS 531  
SECURITY\_CLASS\_CHANGE\_AGENT 531  
SECURITY\_CLASS\_CHANGE\_FROM 531  
SECURITY\_CLASS\_CHANGE\_MODIFIED\_DATETIME 531  
SECURITY\_CLASS\_CHANGE\_REASON 531  
SECURITY\_CLASS\_CHANGE\_TO 531  
SESSION\_INITIALIZATION 48  
SESSION\_OBJECT\_TIMESTAMP 27  
SESSION\_TIMESTAMP\_OBJECT 27  
setAction 468  
setAddressFrom 14  
setAddressTo 14  
setArguments 8  
setAutoSave 58  
setBCC 308  
setCancel 38, 47  
setCategories 288, 334  
setCategory 349  
setCC 308  
setChangeRetention 362  
setChangeRights 361  
setChangeSecurityClass 361  
setChangeStatus 362  
setChildClassificationCodeGenerated 251, 387  
setClassificationCode 317  
setComments 470  
setComputerNameFrom 15  
setComputerNameTo 16  
setCreateAccess 412  
setCreateReferences 363  
setCreateSubEntities 361  
setCurrentLocation 395  
setCustodian 396  
setCustomContentTypeResolver 192  
setDate 306  
setDateFrom 18  
setDateTo 18  
setDeleteAccess 360, 413  
setDescription 260, 283, 318, 353, 393, 466, 555  
setDirection 238  
setEmail 555  
setEntityCollectionPageCount 77  
setEntityCollectionPageSize 77  
setEntitySecurityClass 94  
setEntityStatus 94  
setEventIdFrom 17  
setEventIdQueryType 16  
setEventIdTo 17  
setExtensions 56  
setExternalIds 333  
setExternalObjectId 141  
setFilter 18, 454  
setFirstName 554  
setFrom 306  
setHomeLocation 394  
setIconData 556  
setId 392

setInternalLogging 204  
setIsClosed 57, 129  
setIsModified 58  
setIsNew 58  
setKeywords 288, 332, 476  
setLastName 555  
setLevel 205  
setLocalCertificate 46  
setLocalCertificateKey 46  
setMembers 470  
setMessage 8  
setMessageId 305  
setMimes 57  
setMoveAccess 360  
setObjectScope 16  
setOwner 68, 286, 328, 474  
setParts 255  
setPersistent 110  
setPriority 310  
setPropagated 364, 413  
setPropertyName 238  
setReadAccess 359, 412  
setRecursive 19  
setReferenceIds 355  
setRemoteCertificateAccepted 45  
setReturnDue 396  
setScope 453  
setSecurityClass 327, 379  
setSessionQueryType 13  
setSessionScope 13  
setSigned 310  
setSignificance 319  
setSortOrder 18  
setSslProtocols 44  
setStatus 320, 379, 393, 467, 489  
setSubject 309  
setTimeStamp 69  
setTitle 282, 318, 352, 465  
setTo 307  
setTrustAnchors 44  
setUserNameFrom 15  
setUserNameTo 15  
setValidFrom 363, 413  
setValidTo 363, 414  
setValue 248, 401, 403, 496

setValues 403, 404, 405  
setWriteAccess 360, 412  
setXmlValue 402  
setXmlValues 404  
SIGNIFICANCE 530  
STATUS 529  
statusChange 394  
Storage 128  
StorageConnectorException 201, 202  
storeObject 135, 136, 137, 138, 139  
STRING10 514  
STRING100 514  
STRING20 514  
STRING200 514  
STRING30 514  
STRING40 514  
STRING50 514  
STRINGMAX 513  
subEntityCount 269, 325, 374

## T

TEMPLATES 233  
TIME 513  
TIMESTAMP\_OBJECT\_SESSION 27  
TIMESTAMP\_SESSION\_OBJECT 27  
TITLE 529  
toBackEndAuthenticationOptions 191  
toBackEndAuthenticationType 191  
toLegacyContentType 52  
toString 214, 223, 225  
TRANSFERRED\_ENTITY\_AUDIT\_LOG 533  
TRANSFERRED\_ENTITY\_CLASSIFICATION\_CODE 533  
TRANSFERRED\_ENTITY\_EVIDENCE 533  
TRANSFERRED\_ENTITY\_IMPORT\_DATETIME 533  
TRANSFERRED\_ENTITY\_MOVE\_AGENT 533  
TRANSFERRED\_ENTITY\_MOVE\_CLASSIFICATION\_CODE 534  
TRANSFERRED\_ENTITY\_MOVE\_DATETIME 533  
TRANSFERRED\_ENTITY\_MOVE\_REASON 533  
TRANSFERRED\_ENTITY\_SYSTEM\_ID 533

## U

UINT128 513

UINT16 512  
UINT32 512  
UINT64 513  
UINT8 512  
UNKNOWN 5, 50, 122, 212, 229, 242, 511, 540  
unlock 69  
UNSECURED 51  
USER 541  
USER\_NAMES 23

## V

validFrom 427, 441  
validTo 427, 441  
valueCount 433  
valueOf 4, 6, 21, 23, 25, 28, 30, 32, 34, 36, 40, 49, 51, 65, 120,  
122, 124, 208, 211, 213, 217, 219, 221, 227, 229, 231, 233,  
236, 240, 242, 244, 508, 516, 518, 520, 522, 524, 538, 541  
values 4, 6, 21, 23, 25, 27, 30, 32, 34, 36, 40, 49, 51, 65, 120,  
122, 124, 207, 211, 213, 217, 219, 221, 227, 229, 231, 233,  
236, 240, 242, 244, 508, 516, 518, 520, 522, 524, 538, 541  
VERSION\_SERIES\_ID 538  
VERSIONING\_COMMENT 538  
VERSIONING\_LABEL 538  
VERSIONS 233

## W

WRITE 65, 211  
WRITE\_ACCESS 235  
writeAccess 424, 440