Java API Reference

Solstice Enterprise Manager™ 4.1

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
U.S.A. 650-960-1300

Part No. 806-7973-10
October 2001, Revision A

Please
Recycle

Adobe PostScript™

Contents

Preface

This book provides an extensive list of the classes and member methods (functions) defined in the Portable Management Interface (PMI) used to communicate with the Solstice Enterprise Manager (Solstice EM) Management Information Server (MIS).

Who Should Use This Book

This document is intended for programmers who are developing applications that run on top of the Solstice EM MIS. A thorough working knowledge of Java™ and experience using complex programmatic interfaces is assumed. Familiarity with the C++ PMI API classes is also useful.

Before You Read This Book

If you have just acquired the Solstice EM product, read the *Managing Your Network* for an overview of the Solstice EM product functions, features, and components. Read the *Release Notes* for information on installing and starting the product, compatibility issues, minimum hardware and software requirements, known problems, an inventory of the product components, and late breaking information. Also refer to the books listed in “Related Books section for relevent information.

How This Book Is Organized

This book is organized as follows:

Chapter 1, "Common API Classes," describes the common set of objects that you can use with the Java PMI, Alarms, and Topology APIs.

Chapter 2, "Java PMI API," describes the PMI application programming interface.

Chapter 3, "Java Alarms API," describes the Alarms programming interface.

Chapter 4, "Java Topology API," describes the Topology application programming interface.

Related Books

Following is a list of related books:

- *Developing Java Applications*
- *Management Information Server (MIS) Guide*
- *Developing C++ Applications*
- *C++ API Reference*
- *Customizing Guide*

What Typographic Changes Mean

The following table describes the typographic changes used in this book.

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% You have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in *User's Guide*. These are called *class* options. You *must* be root to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P-2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

Accessing Sun Documentation Online

The docs.sun.com[sm] web site enables you to access Sun technical documentation on the Web. You can browse the docs.sun.com archive or search for a specific book title or subject at http://docs.sun.com.

Also, you can view the online documentation by pointing your browser to the following URL, file:/opt/SUNWconn/em/docs/SEMDOCHP/index.html

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can send your comments by email to docfeedback@sun.com.

Please include the part number of your document in the subject line of your email.

CHAPTER 1

Common API Classes

This chapter covers are the common set of objects you can use with the Java PMI, Alarms, and Topology Application Programming Interface (API).

This chapter comprises the following topics:

1.1 `EMObjectAttribute` Class

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMObjectAttribute
```

public final class `EMObjectAttribute`

extends `java.lang.Object`

implements `java.io.Serializable`

The abstract class `EMObjectAttribute` is the superclass of the classes `AlarmRecordAttribute`, `EMCmipAgentAttribute`, `EMRpcAgentAttribute`, `EMSnmpAgentAttribute`, `EMTopoNodeAttribute`, and `EMTopoTypeAttribute`.

1.1.1 Methods

equals

Compares this `EMObjectAttribute` against the specified object. The result is `true` if and only if the argument is not `null` and is an `EMObjectAttribute` object with the same identity as this `EMObjectAttribute`.

```
public boolean equals(java.lang.Object obj)
```

`obj` is the object to compare against.

Returns: `true` if the objects are equal, `false` otherwise.

Overrides: `equals` in class `java.lang.Object`

toString

Returns a String representation of `EMObjectAttribute`.

```
public java.lang.String toString()
```

Returns: String

Overrides: `toString` in class `java.lang.Object`

intValue

Returns a unique integer identifier for this attribute.

```
public int intValue()
```

1.2 EMAttributeSet Class

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMAttributeSet
```

public abstract class `EMAttributeSet`

extends `java.lang.Object`

implements java.lang.Cloneable, java.io.Serializable

The `EMAttributeSet` class is an abstract class that forms the basis for the attribute set classes of each API. `EMAttributeSet` is used with the Alarm and Topology APIs to communicate which attributes of the Alarm or Topology object, the API method should operate.

1.2.1 Methods

add

Adds an attribute to the set.

```
public void add(EMObjectAttribute attribute)
```

Parameters: *`attribute`* is the attribute to add to the set

Throws `ClassCastException` if the attribute is not a valid attribute of the object

addAll

Adds all attributes to the set.

```
public void addAll()
```

and

Performs a logical AND of this set with the argument set. This set contains attributes which are members of both this set and the argument set. This set also has its upper bound adjusted to the greater of its upper bound and the upper bound of the argument set.

```
public void and(EMAttributeSet set)
```

clone

Returns a clone of this EMAttributeSet object.

```
public java.lang.Object clone()
```

Returns: A clone of this EMAttributeSet

elements

Returns an enumeration of the attributes in this set. Enumeration.nextElement() will return an object whose type is a subclass of EMObjectAttribute, e.g. void enumeration(EMAttributeSet set) {Enumeration e=set.elements(); while (e.hasMoreElements() {EMObjectAttribute attribute=(EMObjectAttribute) e.nextElement(); System.out.println(attribute.toString());}}

```
public abstract java.util.Enumeration elements
```

Returns: An enumeration of the attributes in this set

equals

Compares this object against the specified object. The result is `true` if and only if the argument is not `null` and is an `EMAttributeSet` object that contains the same set of attributes as this `EMAttributeSet`.

```
public boolean equals(java.lang.Object obj)
```

Parameters: *`obj`* is the object to compare with

Returns: `true` if the objects are the same; `false` otherwise

Overrides: `equals` in class `java.lang.Object`

getNumMembers

Returns the number of attributes which are members of this set.

```
public int getNumMembers()
```

isMember

Returns whether the argument attribute is a member of this set.

```
public boolean isMember (EMObjectAttribute attributeID)
```

Parameters: `attributeID` is the attribute to test for membership in the set

Returns: `true` if the argument attribute is in the set, otherwise `false`

Throws `ClassCastException` if the attribute is not a valid attribute of the object

or

Performs a logical OR of this set with the argument set. This set contains attributes that are members of this set, the argument set, or both. This set also has its upper bound adjusted to the greater of its upper bound and the upper bound of the argument set.

```
public void or(EMAttributeSet set)
```

Parameters: The argument set

remove

Removes an attribute from the set.

```
public void remove(EMObjectAttribute attribute)
```

Parameters: `attribute` the attribute to remove from the set.

Throws `ClassCastException` if the attribute is not an attribute of the object

removeAll

Removes all attributes from the set.

```
public void removeAll
```

xor

Performs a logical XOR of this set with the argument set. This set contains attributes that are members of either this set or the argument set, but not both. This set also has its upper bound adjusted to the greater of its upper bound and the upper bound of the argument set.

```
public void xor(EMAttributeSet set)
```

Parameters: The argument set

1.3 MOName Class

```
java.lang.Object
   |
   +----com.sun.em.api.common.MOName
```

public final class `MOName`

extends `java.lang.Object`

implements java.io.Serializable

An instance of `MOName` represents a Managed Object Instance name. `findMOsByNodes` method of `EMTopoNode` class returns instances of this class.

1.3.1 Constructor

MOName

Creates an instance of `MOName` given the fully distinguished name (*fdn*) of a Managed Object Instance.

```
public MOName(String moiName)
```

Parameters: *moiName* is the fully distinguished name

1.3.2 Methods

getMOId

Gets the internal ID for the managed object.

```
public int getMOId()
```

getMOName

Gets the fully distinguished name of a managed object instance.

```
public String getMOName()
```

setMOId

Sets the internal ID for the managed object. This is generally not set by the API user, rather it is set internally.

```
public void setMOId(int id)
```

setMOName

Sets the fully distinguished name of a managed object instance.

```
public void setMOName(String moiName)
```

1.4 EMSeverity Class

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMSeverity
```

public final class `EMSeverity`

extends `java.lang.Object`

An instance of `com.sun.em.api.common.EMSeverity` represents one of the six possible alarm severity levels.

1.4.1 Alarm Severity Levels

CLEARED

```
public static final EMSeverity CLEARED
```

CRITICAL

```
public static final EMSeverity CRITICAL
```

INDETERMINATE

```
public static final EMSeverity INDETERMINATE
```

MAJOR

```
public static final EMSeverity MAJOR
```

MINOR

```
public static final EMSeverity MINOR
```

WARNING

```
public static final EMSeverity WARNING
```

1.4.2 Methods

equals

Compares this `EMSeverity` against the specified object. The result is `true` if, and only if, the argument is not `null` and is a `EMSeverity` object with the same value as this `EMSeverity`.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns: `true` if the objects are equal, `false` otherwise.

Overrides: `equals` in class `java.lang.Object`

getId

Returns an integer representation of the `EMSeverity`.

```
public int getId()
```

toString

Returns a String representation of `EMSeverity`.

```
public String toString()
```

Returns: String

Overrides: `toString` in class `java.lang.Object`

CHAPTER 2

Java PMI API

This chapter introduces the classes that make up the Java Portable Management Interface (PMI) APIs.

This chapter comprises the following topics:

2.1 Overview

The Java PMI is a Java based API for manipulating managed objects similar to the C++ PMI API. Java applications can use the Java PMI to communicate with the MIS. JMI is used as an abbreviation for Java PMI throughout this book.

While functionally similar to the C++ PMI API, Java PMI offers a subset of the PMI features. Java PMI includes five major classes:

- `Album` maps to either of `MOHCollectioByRule` or `MOHCollectionEnum`.
- `Platform` maps to the PMI `Platform`
- `MOHandle` maps to PMI `Image`
- `MOHandleCollection` is abstract and maps to PMI `Album`
- `AbstractData` maps to PMI `Morf`
- `EventReport` maps to PMI `CurrentEvent`

Note – The `AbstractData` class can represent more than just an attribute; it can also represent `EventInfo` and so forth. `AbstractData` encapsulates any ASN1 syntax type and corresponding ASN1 value, not necessarily just an attribute.

Additionally, some classes used in the PMI are not reflected in the JMI:

- `Error` class is provided through Java Exceptions.
- `Waiter` class is not provided because JMI only supports Synchronous calls.
- `Coder` class is not defined since BER coding is assumed and is most likely transparent.
- `Syntax` class is not provided as it is fairly low-level and it is not expected that developers will be modifying or accessing ASN type information.
- `AlbumImage` is dispensed with, a single call, `getMOHandles()`, returns an array of `MOHandles`.

For more detailed information on how to use this API, refer to the *Developing Java Applications* book.

2.2 `AbstractData` Class

public class `AbstractData`

extends `Object` implements `Serializable`

The `com.sun.em.api.pmi.AbstractData` class represents an attribute for the managed object. It encapsulates the syntax information along with the value for any attribute. `AbstractData` corresponds to the `Morf` class available under C++ PMI API. This class provides many utility methods regarding the attribute value and also allows the user to break a complex attribute into its constituents.

2.2.1 Constructors

AbstractData

Constructs an instance of the `AbstractData` class.

```
public AbstractData(String attrName, Platform platform)
throws JmiException
```

attrName is the name of the attribute.

platform is the instance of `Platform`.

Throws `JmiException` if there is an error in the construction of `AbstractData`, or a fault in the communication link.

AbstractData

Constructs an instance of the `AbstractData` class.

```
public AbstractData(String attrName,AbstractData list[],
                    Platform platform)
throws JmiException
```

attrName is the name of the attribute.

list is the array of `AbstractData`.

platform is the instance of `Platform` used for establishing connection to MIS.

Throws `JmiException` if there is an error in the construction of `AbstractData`, processing errors are encountered at the server, or a fault in the communication link.

2.2.2 Methods

For error handling, all these methods throw `JmiException`:

- An error occurs in executing the method
- Processing errors are encountered at the server
- An error occurs in the communication link

getStr

Provides the attribute value in `String` form.

```
public String getStr() throws JmiException
```

Returns the attribute value in `String` form.

getLong

Provides the attribute value as `Long`.

```
public long getLong() throws JmiException
```

Returns the attribute value as `Long`.

getDouble

Provides the attribute value as `Double`.

```
public double getDouble() throws JmiException
```

Returns the attribute value as `Double`.

getBigInteger

Provides the attribute value as `BigInteger`.

```
public BigInteger getBigInteger() throws JmiException
```

Returns the attribute value as `BigInteger`.

setStr

Sets the attribute value.

```
public void setStr(String attrValue) throws JmiException
```

attrValue is the value of attribute in `String` form.

setLong

Sets the attribute value with a `Long`.

```
public void setLong(long attrValue) throws JmiException
```

attrValue is the value of attribute as `Long`.

setDouble

Set the attribute value with a `Double`.

```
public void setDouble(double attrValue) throws JmiException
```

attrValue is the value of attribute as `Double`.

setAny

Sets the attribute value with an `AbstractData`.

```
public void setAny(AbstractData data) throws JmiException
```

attrValue is the value of attribute as `AbstractData`.

setBigInteger

Sets the attribute value with a BigInteger.

```
public void setBigInteger(BigInteger attrValue) throws JmiException
```

attrValue is the value of attribute as BigInteger.

isList

Finds out if the attribute is either of SET/SET_OF/SEQUENCE/SEQUENCE_OF.

```
public boolean isList() throws JmiException
```

Returns true if the attribute is either of SET/SET_OF/SEQUENCE/SEQUENCE_OF, otherwise returns false.

isChoice

Indicates if the attribute type is CHOICE or not.

```
public boolean isChoice() throws JmiException
```

Returns true if the attribute type is CHOICE, otherwise returns false.

isSet

Indicates if the attribute type is SET or not.

```
public boolean isSet() throws JmiException
```

Returns true if the attribute is a SET, otherwise returns false.

isSequence

Indicates if the attribute type is SEQUENCE or not.

```
public boolean isSequence() throws JmiException
```

Returns true if the attribute is a SEQUENCE, otherwise returns false.

isAny

Indicates if the attribute type is ANY or not.

```
public boolean isAny() throws JmiException
```

Returns true if the attribute type is ANY, otherwise returns false.

splitList

Creates an AbstractData for each member of the AbstractData and returns an array of all the members. This method works only on AbstractData that are of list type.

```
public AbstractData[] splitList() throws JmiException
```

Returns an array of the members of AbstractData.

extract

Extracts the member of the composite AbstractData using the navigational string.

```
public AbstractData extract(String navigation) throws JmiException
```

navigation is the navigational string.

Returns AbstractData which is a member of the composite AbstractData.

numElements

Returns the number of elements in the list. This method works only on `AbstractData` which are of list type.

```
public int numElements() throws JmiException
```

Returns the number of members for this list type `AbstractData`.

setMemberName

Sets the member in the attribute of `CHOICE` type.

```
public void setMemberName(String str) throws JmiException
```

str is the name of the member syntax. Calling this method sets the behavior of the `AbstractData` according to the member syntax. This method is applicable only on choice types.

getMemberName

Returns the name of member syntax. This method is applicable on choice types only.

```
public String getMemberName() throws JmiException
```

Returns the name of member syntax.

getMemberNames

Returns an array of member names. This method is applicable on choice types only.

```
public String[] getMemberNames() throws JmiException
```

Returns an array of member names for this particular choice type.

2.3 `AuthList` Class

public class `AuthList`

extends `Object` implements `Serializable`

The `com.sun.em.api.pmi.AuthList` class represents a list of authorized applications or features of a given application. This list is determined by access control setup of MIS server. `AuthList` corresponds to two C++ PMI classes: `AuthFeatures` and `AuthApps`, for application/feature level access control purposes.

2.3.1 Constructors

`AuthList`

Constructs an instance of the `AuthList` class.

```
public AuthList(String list[])
```

list is the list of names of applications or features.

2.3.2 Methods

`isAuthorized`

Checks to see if the application is authorized.

```
public boolean isAuthorized(String name)
```

name is the name of the application.

Returns true if the application is authorized, otherwise returns false.

hasFullAccess

Checks to see if the user has full access.

```
public boolean hasFullAccess()
```

Returns true if the user has full access, otherwise returns false.

getAuthorizedList

Returns the authorized list.

```
public String[] getAuthorizedList()
```

Returns an empty array (zero length) if the user has full access. If the user doesn't have any access, the return value will be a null object. Otherwise, the array containing applications or features will be returned.

2.4 EventReport Class

public class `EventReport`

extends `Object` implements `Serializable`

The `com.sun.em.api.pmi.`The `EventReport` class provides information for any event received from the MIS. Applications can register for events using the appropriate methods in the various classes for example, `Platform`, `MOHandle`, `MOHcollection` (`ByRule/Enum`) and so forth. The `EventReport` class is similar to the `CurrentEvent` class available under C++ PMI API.

2.4.1 Methods

For error handling, most of these methods throw `JmiException` when there is an error in executing this method or a fault in the communication link.

getName

Returns the event type in String form.

```
public String getName()
```

getInfo

Returns the event information in String form.

```
public String getInfo() throws JmiException
```

getInfoRaw

Returns the event information in `AbstractData` form.

```
public AbstractData getInfoRaw() throws JmiException
```

getMOName

Returns the managed object name for the event in String form.

```
public String getMOName()
```

getMOClass

Returns the managed object class for the event in String form.

```
public String getMOClass()
```

2.5 `EventReportListener` Interface

public interface `EventReportListener`

The `com.sun.em.api.pmi.EventReportListener` interface will be implemented by any user interested in receiving `EventReports`. This interface defines the method `handler` which will be called by JMI when the event arrives. The abstract `handler` in the `EventReportListener` needs to be implemented for receiving and processing events.

2.5.1 Methods

`handler`

This method is called by JMI when an event arrives.

```
public void handler(EventReport ind)
```

ind is an instance of `EventReport` which provides the event information.

2.6 `JmiException` Class

public class `JmiException`

extends `Exception`

The `com.sun.em.api.pmi.JmiException` class is used to represent errors encountered in method execution, broken communication links, or processing errors encountered at the server.

2.6.1 Constructors

JmiException

Constructs an instance of the `JmiException` class.

```
JmiException(String msg)
```

msg is the message provided by the object that is throwing the exception.

2.7 MOHandle Class

public class `MOHandle`

extends `Object`

The `com.sun.em.api.pmi.MOHandle` class represents the Managed Object Handle. This class is similar to the `Image` class from C++ PMI API. It provides methods to operate on the object over the network and the copy held in the server.

In order to deliver events, you need to define and instantiate a class that implements the interface `EventReportListener`. An instance of the class implementing this interface is passed as a parameter to the event registration methods (`addAttributeValueChangeListener`, `addObjectCreationListener` and so forth). On event delivery, Java PMI invokes the *handler* method of this interface.

Similar to event registration, callbacks for events can be deregistered by calling methods such as `removeAttributeValueChangeListener`, `removeObjectCreationListener` and so forth. Java PMI expects the same listener arguments passed as the one used when registering the event.

2.7.1 Constructors

MOHandle

Constructs an instance of the `MOHandle` class.

```
public MOHandle(String instance,Platform platform)
throws JmiException
```

instance is the name of the object in DN form.

platform is the instance of the `Platform` used for establishing connection to MIS.

Throws `JmiException` if there is an error in the construction of `MOHandle`, or a fault in the communication link.

MOHandle

Constructs an instance of the `MOHandle` class.

```
public MOHandle(String instance,String objclass,Platform platform)
throws JmiException
```

instance is the name of the object in DN form.

objclass is the class name of the object.

platform is the instance of the `Platform` used for establishing connection to MIS.

Throws `JmiException` if there is an error in the construction of `MOHandle`, processing errors occur at the server, or a fault occurs in the communication link.

2.7.2 Methods

Unless otherwise noted, all these methods throw `JmiException` if an error occurs in executing the method, a fault occurs in the communication link, or processing errors are encountered at the server.

addAttributeValueChangeListener

Registers a callback for Attribute Value Change of the managed object represented by this `MOHandle`.

```
public void addAttributeValueChangeListener
                    (EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which implements handler method.

Throws `JmiException` if there is an error in registering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

addObjectCreationListener

Registers a callback for Object Creation of a managed object represented by this `MOHandle`.

```
public void addObjectCreationListener(EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which implements handler method.

Throws `JmiException` if there is an error in registering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

addObjectDeletionListener

Registers a callback for Object Deletion of a managed object represented by this `MOHandle`.

```
public void addObjectDeletionListener(EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` that implements handler method.

Throws `JmiException` if there is an error in registering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

addRawEventListener

Registers a callback for any event for the managed object represented by this `MOHandle`.

```
public void addRawEventListener(EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` that implements handler method.

Throws `JmiException` if there is an error in registering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

removeAttributeValueChangeListener

Removes the callback for Attribute Value Change for this `MOHandle`.

```
public void
removeAttributeValueChangeListener(EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` that implements handler method.

Throws `JmiException` if there is an error in deregistering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

removeObjectCreationListener

Removes the callback for Object Creation for this `MOHandle`.

```
public void removeObjectCreationListener
                    (EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` that was used for registering the callback.

Throws `JmiException` if there is an error in deregistering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

removeObjectDeletionListener

Removes the callback for Object Deletion for this `MOHandle`.

```
public void removeObjectDeletionListener
                    (EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` that was used for registering the callback.

Throws `JmiException` if there is an error in deregistering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

removeRawEventListener

Removes the callback for registered event using `addRawEventListener`.

```
public void removeRawEventListener(EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` that was used for registering the callback.

Throws `JmiException` if there is an error in deregistering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

isEmpty

Indicates whether the server holds an underlying data object for this `MOHandle`.

```
public boolean isEmpty() throws JmiException
```

Returns `true` or `false`.

getTrackIdList

Returns an array of attribute names. The array contains the names of the attributes that are tracked by the `MOHandle` for any changes in their values (in the network).

```
public String[] getTrackIdList() throws JmiException
```

Returns an array of zero or more attribute names.

setTrackIdList

Sets the attributes of the `MOHandle` to track changes in their values in the network based on the `trackIdList` supplied.

```
public void setTrackIdList(String trackIdList[]) throws JmiException
```

trackIdList is an array of attribute names that are expected to be tracked.

setTracking

This method enables or disables tracking of the attributes for this `MOHandle`.

```
public void setTracking(boolean tracking) throws JmiException
```

tracking is a boolean flag. A true value sets tracking on and false turns it off.

removeFromTrackIdList

Removes the attributes from *trackIdList* when in tracking mode.

```
public void removeFromTrackIdList(String trackIdList[])
throws JmiException
```

trackIdList is a list of attribute names that are removed from the tracking mode.

getStr

Gets the value of the attribute in `String` form.

```
public String getStr(String attrName) throws JmiException
```

attrName is the name of the attribute.

getStr

Gets the values of more than one attribute at a time.

```
public String[] getStr(String attrList[]) throws JmiException
```

attrList is the list of attribute names for which values are returned.

Returns attribute values in an array of `Strings`.

getAttrNames

Returns all the attribute names for the `MOHandle` as an array of `Strings`.

```
public String[] getAttrNames() throws JmiException
```

Returns attribute names in an array of `Strings`.

getLong

Gets the value of the attribute as a `Long`.

```
public long getLong(String attrName) throws JmiException
```

attrName is the name of the attribute.

Returns an attribute value as a `Long`.

getDouble

Gets the value of the attribute as a `Double`.

```
public double getDouble(String attrName) throws JmiException
```

attrName is the name of the attribute.

Returns an attribute value as a `Double`.

getBigInteger

Gets the value of the attribute as a `BigInteger`.

```
public BigInteger getBigInteger(String attrName)
throws JmiException
```

attrName is the name of the attribute.

Returns an attribute value as a `BigInteger`.

getRaw

Gets the value of the attribute as an `AbstractData`.

```
public AbstractData getRaw(String attrName) throws JmiException
```

attrName is the name of the attribute.

Returns an attribute value as `AbstractData`.

setStr

Sets the value of the attribute.

```
public void setStr(String attrName,String value)
throws JmiException
```

attrName is the name of the attribute.

value is the value of the attribute as a String.

setStr

Sets the values of more than one attribute at a time.

```
public void setStr(String attrList[],String values[])
throws JmiException
```

attrList is the names of the attribute.

values is the corresponding values of the attributes.

exists

Indicates whether or not the object represented by this `MOHandle` already exists.

```
public boolean exists() throws JmiException
```

Returns a boolean value. If `MOHandle` exists, it returns `true`, otherwise it returns `false`.

setLong

Sets the value of the attribute as a `Long`.

```
public void setLong(String attrName,long value) throws JmiException
```

attrName is the name of the attribute.

value is the value of the attribute as a `Long`.

setDouble

Sets the value of the attribute as a `Double`.

```
public void setDouble(String attrName,double value)
throws JmiException
```

attrName is the name of the attribute.

value is the value of the attribute as a `Double`.

setBigInteger

Sets the value of the attribute as `BigInteger`.

```
public void setBigInteger(String attrName,BigInteger value)
throws JmiException
```

attrName is the name of the attribute.

value is the value of the attribute as a `BigInteger`.

setRaw

Sets the value of the attribute as AbstractData.

```
public void setRaw(String attrName,AbstractData rawAttr)
throws JmiException
```

attrName is the name of the attribute.

value is the value of the attribute in AbstractData form.

getObjectName

Returns the name of the object in distinguished form.

```
public String getObjectName() throws JmiException
```

Returns the object name in String form.

getObjectClass

Returns the name of the object class for this MOHandle.

```
public String getObjectClass() throws JmiException
```

Returns the object class.

getObjectState

Returns the state of this MOHandle. If the state is up, the MOhandle is "booted", that is, the MOHandle has all the metadata information and attribute values (if the object exists). If the state is down, the MOHandle is not booted.

```
public String getObjectState() throws JmiException
```

Returns the up or down state of MOHandle.

MGet

Executes a management operation of getting all the attribute values from the network for this `MOHandle`.

```
public void MGet(double to) throws JmiException
```

to is the time out value provided for this operation to finish.

MGet

Executes a management operation of getting some of the attribute values from the network for this `MOHandle`.

```
public void MGet(String attrlist[],double to) throws JmiException
```

attrlist is the attribute list.

to is the time out value provided for this operation to finish.

MSet

Executes a management operation of setting all the attribute values from the `MOHandle` over the network (set locally by calling *setStr*).

```
public void MSet(double to) throws JmiException
```

to is the time out value provided for this operation to finish.

MCreate

Executes a management operation of creating the object in the network represented by this `MOHandle`.

```
public void MCreate(double to) throws JmiException
```

to is the time out value provided for this operation to finish.

MCreateWithin

Executes a management operation of creating the object in the network represented by this MOHandle with the superior object as in the argument.

```
public void MCreateWithin(String superiorObj,double to)
throws JmiException
```

superiorObj is the name of the object that will contain the new object.

to is the time out value provided for this operation to finish.

MDelete

Executes a management operation of deleting the object in the network represented by this MOHandle.

```
public void MDelete(double to) throws JmiException
```

to is the time out value provided for this operation to finish.

MAction

Executes a management operation of calling an action on the object in the network represented by this MOHandle.

```
public String MAction(String actionName,String param,double to)
throws JmiException
```

actionName is the name of the action supported by the object.

param is the parameter required to execute this action.

to is the time out value provided for this operation to finish.

MEventReport

Sends an event with the type and the event information provided.

```
public void MEventReport(String eventName,String eventInfo)
throws JmiException
```

eventName is the name of the event type.

eventInfo is the event information in the String form.

2.8 MOHCollectionByRule Class

public class MOHCollectionByRule

extends Object

The `com.sun.em.api.pmi.MOHCollectionByRule` class represents a group of `MOHandles` which satisfy the rule specified by base object (the distinguished name of the base object), scope (the scope of the collection in the MIT), and filter (the filter applied as a criteria for membership). The `MOHCollectionByRule` class does not allow user manipulation of its membership. This class will track all the `MOHandles` within the collection, based on changes in the network for these `MOHandles`.

In order for events to be delivered, users of `MOHCollectionByRule` need to define and instantiate a class that implements the `EventReportListener` interface. An instance of the class implementing this interface is passed as a parameter to these event registration methods (`addAttributeValueChangeListener`, `addObjectCreationListener` and so forth). Upon event delivery, Java PMI invokes the handler method of this interface.

Two additional events are supported by `MOHCollectionByRule`, namely, `MOHandleIncluded` and `MOHandleExcluded`. These events are delivered when a `MOHandle` is included in the collection (because it satisfies the rule for collection) or excluded.

Similar to event registration, callbacks for events can be deregistered by calling methods such as `removeAttributeValueChangeListener`, `removeObjectCreationListener` and so forth. The user of Java PMI is expected to pass the same listener argument as the one used when registering earlier.

2.8.1 Constructors

MOHCollectionByRule

Constructs an instance of the `MOHCollectionByRule` class.

```
public MOHCollectionByRule(String baseobj,String scope,
                           String filter,Platform platform)
throws JmiException
```

baseObj is the distinguished name of the base object.

scope is the cope of the collection in the MIT.

filter is the filter applied as a criteria for membership.

platform is the instance of `Platform`.

Throws `JmiException` if there is an error in the construction of `MOHCollectionByRule`, or a fault in the communication link.

2.8.2 Methods

Note – For error handling, all these methods throw `JmiException` if an error occurs while executing, processing errors are encountered at the server, or a fault occurs in the communication link.

---

allSetProp

Sets the specified property of every image currently in the existing `MOHCollectionByRule` instance to the specified value.

```
public void allSetProp(String property,String value)
throws JmiException
```

property is the name of the `MOHandle` property.

value is the value to which the specified property is set.

setTracking

Modifies tracking mode of the collection . The collection will get updated whenever the managed objects, represented by member `MOHandles`, are changed in the network when tracking is set to on.

```
public void setTracking(boolean tracking) throws JmiException
```

tracking is a boolean flag. If it is set to `true`, collection is in tracking mode or it does not track.

getTracking

Returns the tracking mode for the collection.

```
public boolean getTracking() throws JmiException
```

Returns a boolean value to indicate whether or not the collection is tracking.

populate

Populates the collection from the network based on the base object, scope, and the filter defined in the constructor.

```
public void populate(double to) throws JmiException
```

to is the time out value provided for this operation to finish.

getScope

Returns the scope for this collection.

```
public String getScope() throws JmiException
```

Returns the `String` value of the scope.

getFilter

Returns the filter for this collection.

```
public String getFilter() throws JmiException
```

Returns the `String` value of the filter.

getBaseManagedObject

Returns the name of the base object for this collection.

```
public String getBaseManagedObject() throws JmiException
```

Returns the name of the base object for this collection.

setScope

Modifies the scope for this collection.

```
public void setScope(String scope) throws JmiException
```

scope is the scope for the collection.

setFilter

Modifies the filter for this collection.

```
public void setFilter(String filter) throws JmiException
```

filter is the filter for the collection.

setBaseManagedObject

Modifies the base managed object for this collection.

```
public void setBaseManagedObject(String baseobj)
throws JmiException
```

baseObj is the name of the base object.

getMOHandles

Returns the members of this collection as an array of `MOHandles`.

```
public MOHandle[] getMOHandles() throws JmiException
```

Returns the `MOHandles` in the collection as an array.

MGet

Management operation to obtain information of all the `MOHandles` within the collection.

```
public void MGet(double to) throws JmiException
```

to is the time out value for the operation to finish.

MSet

Management operation to store information of all the `MOHandles` within the collection.

```
public void MSet(double to) throws JmiException
```

to is the time out value for the operation to finish.

2.9 MOHCollectionEnum Class

public class MOHCollectionEnum

extends Object

The com.sun.em.api.pmi.MOHCollectionEnum class represents a group of MOHandles in the Java PMI. The MOHCollectionEnum class allows a user to manipulate its membership.

In order for events to be delivered, users of MOHCollectionEnum must define and instantiate a class that implements the interface EventReportListener. An instance of the class implementing this interface is passed as a parameter to the event registration methods (addAttributeValueChangeListener, addObjectCreationListener and so forth). Upon event delivery, Java PMI invokes the handler method of this interface.

Similar to event registration, callbacks for events can be deregistered by calling methods such as removeAttributeValueChangeListener, removeObjectCreationListener etc. The user of Java PMI is expected to pass the same listener argument as the one used used while registering the event earlier.

2.9.1 Constructors

MOHCollectionEnum

Constructs an instance of the MOHCollectionEnum class from a MOHcollectionByRule instance. The instance of Platform in the constructor ensures that this instance of collection is created in the same process as that of the Platform.

```
public MOHCollectionEnum(MOHCollectionByRule collection,
                         Platform platform)
throws JmiException
```

collection is an instance of MOHCollectionByRule.

platform is an instance of Platform.

Throws JmiException if there is an error in the construction of MOHCollectionEnum, or a fault in the communication link.

MOHCollectionEnum

Constructs an instance of the `MOHCollectionEnum` class. The instance of `Platform` in the constructor ensures that this instance of collection is created in the same process as that of the Platform.

```
public MOHCollectionEnum(Platform platform) throws JmiException
```

platform is an instance of `Platform`.

Throws `JmiException` if there is an error in the construction of `MOHCollectionEnum`, or a fault in the communication link.

2.9.2 Methods

Unless otherwise noted, all these methods throw `JmiException` if an error occurs in the execution of this method, processing errors are encountered at the server, or a fault occurs in the communication link.

allSetProp

Sets the specified property of every image currently in the existing `MOHCollectionEnum` instance to the specified value.

```
public void allSetProp(String property, String value)
throws JmiException
```

property is the name of the `MOHandle` property.

value is the value you set for the specified property.

getMOHandles

Returns an array of `MOHandles` within the collection.

```
public MOHandle[] getMOHandles() throws JmiException
```

Returns an array of `MOHandles`.

Mget

This is a management operation to obtain information on all the `MOHandles` within the collection.

```
public void MGet(double to) throws JmiException
```

to is the time out for the operation to finish.

MSet

This is a management operation to store information of all the `MOHandles` within the collection.

```
public void MSet(double to) throws JmiException
```

to is time out for the operation to finish.

Mcreate

This is a management operation to create all the `MOHandles` within the collection.

```
public void MCreate(double to) throws JmiException
```

to is the time out for the operation to finish.

MCreateWithin

This is a management operation to create all the `MOHandles` within the collection under a superior object.

```
public void MCreateWithin(String superiorObj, double to)
throws JmiException
```

superiorObj is the name of the superior object.

to is the time out for the operation to finish.

MDelete

This is a management operation to delete all the `MOHandles` within the collection.

```
public void MDelete(double to) throws JmiException
```

to is the time out for the operation to finish.

MAction

This is a management operation to call an action on all the `MOHandles` within the collection.

```
public void MAction(String actionName, String parameter, double to)
throws JmiException
```

actionName is the name of the action.

parameter is the argument required for the action.

to is the time out for the operation to finish.

exclude

Excludes a `MOHandle` from this collection.

```
public void exclude(MOHandle mh) throws JmiException
```

mh is the MOHandle instance.

exclude

Excludes a collection of `MOHandles` from this collection.

```
public void exclude(MOHCollectionEnum collection)
throws JmiException
```

collection is the instance of a `MOHCollectionEnum`.

include

Adds a `MOHandle` to this collection.

```
public void include(MOHandle mh) throws JmiException
```

mh is the `MOHandle` instance.

include

Adds a collection of `MOHandles` to this collection.

```
public void include(MOHCollectionEnum collection)
throws JmiException
```

collection is the instance of a `MOHCollectionEnum`.

addAttributeValueChangeListener

Registers a callback for Attribute Value Change of any object in the collection in the network.

```
public void addAttributeValueChangeListener
                    (EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which implements handler method.

Throws `JmiException` if there is an error in registering the callback or a fault in the communication link.

addObjectCreationListener

Registers a callback for Object Creation of any managed object in the collection in the network.

```
public void addObjectCreationListener(EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which implements handler method.

Throws `JmiException` if there is an error in registering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

addObjectDeletionListener

Registers a callback for Object Deletion of any managed object in the collection in the network.

```
public void addObjectDeletionListener(EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which implements handler method.

Throws `JmiException` if there is an error in registering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

addRawEventListener

Registers a callback for any raw event forwarded by the MIS for a MOHandle in the collection.

```
public void addRawEventListener(EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which implements handler method.

Throws `JmiException` if there is an error in registering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

removeAttributeValueChangeListener

Removes the callback for Attribute Value Change of a managed object in the collection.

```
public void removeAttributeValueChangeListener
                          (EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which was used for registering callback.

Throws `JmiException` if there is an error in deregistering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

removeObjectCreationListener

Removes the callback for Object Creation of a managed object in the collection.

```
public void removeObjectCreationListener
                      (EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which was used for registering the callback.

Throws `JmiException` if there is an error in deregistering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

removeObjectDeletionListener

Removes the callback for Object Deletion of a managed object in the collection.

```
public void removeObjectDeletionListener
                      (EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which was used for registering this callback.

Throws `JmiException` if there is an error in deregistering the callback, processing errors are encountered at the server, or a fault occurs in the communication link.

removeRawEventListener

Removes the callback for the registered event using *addRawEventListener*.

```
public void removeRawEventListener(EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which was used for registering the callback.

Throws `JmiException` if there is an error in deregistering the callback processing errors are encountered at the server, or a fault occurs in the communication link.

2.10 `Platform` Class

public class Platform

extends Object

The `com.sun.em.api.pmi.Platform` class represents the Agent Handle for JMI. It needs to be instantiated to set up a connection with the MIS. Additionally, it spawns a remote server process which acts as the middle tier in the client-server model.

All operations happen through the Java server process. Each platform instance is associated with a unique server instance. All the major classes take a `Platform` instance as a parameter.

In order for events to be delivered, users of `Platform` must define and instantiate a class that implements the `EventReportListener` interface. An instance of the class implementing this interface is passed as a parameter to the event registration methods (`addAttributeValueChangeListener`, `addObjectCreationListener` and so forth). Upon event delivery, `Platform` invokes the handler method of this interface.

Similar to event registration, callbacks for events can be deregistered by calling methods such as `removeAttributeValueChangeListener`, `removeObjectCreationListener` and so forth. The user of `Platform` is expected to pass the same listener argument as the one used used while registering the event earlier.

2.10.1 Constructors

Platform

Constructs an instance of the `Platform` class.

```
public Platform(String host,String misName,String user,
                        String passwd)
throws JmiException
```

host is the host name where the server resides. This is the host where `jme_services` was started.

misName is the name of MIS host.

user is the user name.

passwd is the password. The password is supplied in clear-text but is not transmitted in clear-text to the server, except if the client program is an applet running in Netscape. The encryption is export quality.

Throws `JmiException` if there is an error in the construction of `Platform`, or a fault in the communication link.

2.10.2 Methods

addAttributeValueChangeListener

Registers a callback for Attribute Value Change of any object in the network.

```
public void addAttributeValueChangeListener
                        (EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which implements handler method.

Throws `JmiException` if there is an error in registering the callback or a fault in the communication link.

addObjectCreationListener

Registers a callback for Object Creation of any managed object in the network.

```
public void addObjectCreationListener(EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which implements handler method.

Throws `JmiException` if there is an error in registering the callback or a fault in the communication link.

addObjectDeletionListener

Registers a callback for Object Deletion of any managed object in the network.

```
public void addObjectDeletionListener(EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which implements handler method.

Throws `JmiException` if there is an error in registering the callback or a fault in the communication link.

addRawEventListener

Registers a callback for any event forwarded by the MIS.

```
public void addRawEventListener(EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which implements handler method.

Throws `JmiException` if there is an error in registering the callback or a fault in the communication link.

removeAttributeValueChangeListener

Removes the callback for Attribute Value Change of a managed object in the network.

```
public void removeAttributeValueChangeListener
                          (EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which was used for registering callback.

Throws `JmiException` if there is an error in registering the callback or a fault in the communication link.

removeObjectCreationListener

Removes the callback for Object Creation of a managed object in the network.

```
public void removeObjectCreationListener
                          (EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which was used for registering the callback.

Throws `JmiException` if there is an error in deregistering the callback or a fault in the communication link.

removeObjectDeletionListener

Removes the callback for Object Deletion of a managed object in the network.

```
public void removeObjectDeletionListener
                          (EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which was used for registering this callback.

Throws `JmiException` if there is an error in deregistering the callback or a fault in the communication link.

removeRawEventListener

Removes the callback for registered event using `addRawEventListener`.

```
public void removeRawEventListener(EventReportListener listener)
throws JmiException
```

listener is the instance of `EventReportListener` which was used for registering the callback.

Throws `JmiException` if there is an error in deregistering the callback or a fault in the communication link.

getAuthorizedFeatures

Returns a list of authorized features for an application for a particular user. It is possible through Solstice EM, to make some features of an application accessible/inaccessible to a user. The instance of `AuthList` returned provides a method (`isAuthorized`) which indicates whether or not the user is authorized to use that feature.

```
public AuthList getAuthorizedFeatures(String user,String appName)
throws JmiException
```

user is the user name.

appName is the application name.

Returns the instance of `AuthList` which the user can further query for additional information.

getAuthorizedApps

Returns a list of authorized applications for a particular user. It is possible through Solstice EM, to make some applications accessible/inaccessible to a user. The instance of `AuthList` returned provides a method (`isAuthorized`) which indicates whether or not the user is authorized to use that application.

```
public AuthList getAuthorizedApps(String user) throws JmiException
```

user is the user name.

Returns the instance of `AuthList` which the user can further query for additional information.

getUserName

Returns the user name.

```
public String getUserName()
```

Returns the user name used for the creation of `Platform` and log in.

getMisName

Returns the MIS name.

```
public String getMisName()
```

Returns the host name of MIS to which the server is connected.

CHAPTER 3

Java Alarms API

This chapter presents the classes that make up the Java Alarms Application Programming Interface.

This chapter comprises the following topics:

3.1 Overview

The Java Alarms API enables you to:

- Query alarms under a specific log
- Clear/acknowledge and set the attributes of a given alarm
- Query alarms in batches (to return alarms asynchronously)
- Delete alarms
- Register for alarm creation, modification, and deletion events

For information on how to use these APIs, see the *Developing Java Applications* book.

3.2 `AlarmAttributeNotSetException` Class

public class `AlarmAttributeNotSetException`

extends `throwable`

The `com.sun.em.api.alarm.AlarmAttributeNotSetException` class is used for exceptions thrown when the user tries to access fields in `AlarmRecord` that were not set.

3.2.1 Constructors

`AlarmAttributeNotSetException`

This is the class constructor for the exception.

```
public AlarmAttributeNotSetException(String message)
```

AlarmAttributeNotSetException

This is the Class Constructor for the exception.

```
public AlarmAttributeNotSetException(Throwable t)
```

3.3 AlarmsBatchListener Interface

public interface AlarmsBatchListener

The com.sun.em.api.alarm.AlarmsBatchListener interface provides notifications whenever a new batch is received or when the query is completed (that is, all batches have been received).

3.3.1 Methods

batchReceived

This method is called when a new batch is being received.

```
public void batchReceived(int callId,AlarmRecord record[])
```

callId is the Integer returned when the batch call was initiated.

record is an array of AlarmRecord returned in this batch.

Returns none.

batchDone

This method is called when there are no more alarms to receive for that batch query.

```
public void batchDone(int callId)
```

callId is the `BatchId` referencing the given batch.

Returns none.

3.4 `AlarmException` Class

public class `AlarmException`

extends `RemoteException`

The `com.sun.em.api.alarm.AlarmException` class wraps and hides Remote Method Invocation (RMI) exceptions to hide the transport layer. The exception is thrown when an error occurs in the underlying service or an exceptional condition exists on the server.

3.4.1 Inheritance

```
java.lang.Object
   |
   +--java.lang.Throwable
           |
           +--java.lang.Exception
                   |
                   +--java.io.IOException
                           |
                           +--java.rmi.RemoteException
                                   |
                                   +--com.sun.em.api.alarm.AlarmException
```

3.4.2 Constructors

AlarmException

```
public AlarmException(String message)
```

AlarmException

```
public AlarmException(Throwable t)
```

3.5 AlarmLogCreationListener Interface

public interface `AlarmLogCreationListener`

An `AlarmLogEvent` event occurs whenever an alarm is being created, deleted, or modified. The `com.sun.em.api.alarm.AlarmLogCreationListener` interface allows you to be notified whenever an alarm is being created.

See Also: `AlarmLogEvent`

3.5.1 Methods

alarmRecordCreated

```
public void alarmRecordCreated(AlarmLogEvent event)
```

3.6 AlarmLogDeletionListener Interface

public interface AlarmLogDeletionListener

An AlarmLogEvent event occurs whenever an alarm is being created, deleted or modified. The com.sun.em.api.alarm.AlarmLogDeletionListener interface allows you to be notified whenever an alarm is being deleted.

See Also: AlarmLogEvent

3.6.1 Methods

alarmRecordDeleted

```
public void alarmRecordDeleted(AlarmLogEvent event)
```

3.7 AlarmLogEvent Class

public class AlarmLogEvent

extends Object implements Serializable

The com.sun.em.api.alarm.AlarmLogEvent class implements the events associated with creation, deletion of alarm records, and the modification of alarm record attributes.

3.7.1 Variables

OBJECT_CREATED

Type of the event being returned: OBJECT_CREATED for alarm creation.

```
public static final int OBJECT_CREATED
```

OBJECT_DELETED

Type of the event being returned: OBJECT_DELETED for alarm deletion.

```
public static final int OBJECT_DELETED
```

ATTR_VALUE_CHANGED

Type of the event being returned: ATTR_VALUE_CHANGED for alarm modification.

```
public static final int ATTR_VALUE_CHANGED
```

ALARM_EVENT_ID

String identifying the type of event.

```
public static final String ALARM_EVENT_ID
```

3.7.2 Methods

getEventType

Returns the type of the event: OBJECT_CREATED, OBJECT_DELETED, ATTR_VALUE_CHANGED.

```
public int getEventType()
```

getAlarmRecord

Returns the alarm record corresponding to the event.

```
public AlarmRecord getAlarmRecord()
```

See Also: `AlarmRecord`

getAlarmRecordId

Returns the alarm record ID corresponding to the event.

```
public AlarmRecordId getAlarmRecordId()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if `AlarmRecordId` was not set in the initial `AlarmRecordAttributeSet` list.

See Also: `AlarmRecordId`

toString

Returns a String containing String representation of the event type and the alarm record.

```
public String toString()
```

Overrides *toString* in class `Object`.

toTypeString

Returns a String representation of the event type.

```
public String toTypeString()
```

3.8 AlarmLogListener Interface

public interface AlarmLogListener

An AlarmLogEvent event occurs whenever an alarm is being created, deleted or modified. The com.sun.em.api.alarm.AlarmLogListener interface allows you to be notified whenever an alarm is being created, deleted, or modified.

See Also: AlarmLogEvent

3.8.1 Methods

alarmRecordCreated

```
public void alarmRecordCreated(AlarmLogEvent event)
```

alarmRecordDeleted

```
public void alarmRecordDeleted(AlarmLogEvent event)
```

alarmRecordModified

```
public void alarmRecordModified(AlarmLogEvent event)
```

3.9 AlarmLogModificationListener Interface

public interface AlarmLogModificationListener

An AlarmLogEvent event occurs whenever an alarm is being created, deleted or modified. The com.sun.em.api.alarm.AlarmLogModificationListener interface allows you to be notified whenever an alarm is being modified.

See Also: AlarmLogEvent

3.9.1 Methods

alarmRecordModified

```
public void alarmRecordModified(AlarmLogEvent event)
```

3.10 AlarmLog Class

public class AlarmLogSvc

extends Log

The com.sun.em.api.alarm.AlarmLog class represents an alarm instantiation which enables the user to query, clear, acknowledge, delete alarms and receive event notifications.

3.10.1 Constructors

AlarmLog

Creates an instance of AlarmLog by specifying the log where the alarms are stored.

```
public AlarmLog(Platform platform,LogName logN)
    throws AlarmException
```

platform is the platform for the alarm Service.

logN is the object that contains the name of the log.

Throws AlarmException if there is an internal error in the alarm service.

See Also: LogName

3.10.2 Methods

getLogName

Returns under which log the alarms are stored.

```
public LogName getLogName() throws AlarmException
```

Returns the LogName.

setEventAttrSet

Specify which attributes are to be returned when receiving events.

```
public void setEventAttrSet(AlarmRecordAttributeSet attrSet1)
    throws AlarmException
```

attrSet1 is the list of attributes to be returned.

Returns nothing.

See Also: AlarmRecordAttributeSet

getAlarmsInBatches

Gets a batch of alarms in the Alarm Log. The user will receive all alarms matching the given query in chunks. Each chunk will contain the *batchSize* number of alarms. The *attrSet* variable specifies which attributes to get back in the `alarmRecord`.

```
public int getAlarmsInBatches
                (Query query,int batchSize,AlarmsBatchListener listener,
                AlarmRecordAttributeSet attrSet)
    throws AlarmException
```

query finds the alarms that match this query.

batchSize gets at most this many alarms in a chunk.

listener applies to the batch response.

attrSet specifies which attributes to get back in the `alarmRecord`.

Returns a handle that can be used to stop this call.

See Also: `AlarmsBatchListener`, `AlarmRecordAttributeSet`

stopGetAlarmsInBatches

Stops a call to get a batch of alarms in the Alarm Log.

```
public void stopGetAlarmsInBatches(int callId)
    throws AlarmException
```

callId is the handle from `getAlarmsInBatches`.

Returns nothing.

getAlarms

Returns all alarms that match the query.

```
public AlarmRecord[]
        getAlarms(Query query,AlarmRecordAttributeSet attrSet)
    throws AlarmException
```

query is used to find the alarms that match this query.

attrSet specifies which `AlarmRecord` attributes to retrieve.

Returns an array of `AlarmRecord`. It can be null if no records are found.

See Also: `AlarmRecord`

getAlarmCount

Returns the count of alarms that match the query.

```
public int getAlarmCount(Query query) throws AlarmException
```

query is used to find the alarms that match this query.

attrSet specifies which `AlarmRecord` attributes to retrieve.

Returns array of `AlarmRecord`. It can be null if no records are found.

See Also: `AlarmRecord`

getAlarmCountBySeverity

Returns an Integer array that contains the count of alarm per severity.

```
public int[] getAlarmCountBySeverity
                          (MOName moName[],EMSeverity severity[])
    throws AlarmException
```

MOName gets the alarm severities for these managed objects.

severity specifies the severities.

Returns: Array of Integer containing the count of alarm per severity.

See Also: MOName, EMSeverity

setClearAlarms

Clears the alarms against the given alarm records.

```
public void setClearAlarms
               (AlarmRecordId alarmRecordIds[], String clearText[])
    throws AlarmException
```

alarmRecordIds is the array of alarms to clear.

String is the text to display in the *ClearText* attribute field of the cleared alarm.

See Also: AlarmRecordId

setAckAlarms

Acknowledges the alarms corresponding to the given alarm records.

```
public void setAckAlarms
               (AlarmRecordId alarmRecordIds[],String ackText[])
    throws AlarmException
```

alarmRecordIds is the array of alarms to acknowledge.

String is the text to display in the *ackText* attribute field of the acknowledged alarm.

See Also: AlarmRecordId

setDisplayAlarms

Indicates which alarm records should be displayed.

```
public void setDisplayAlarms
            (AlarmRecordId alarmRecordIds[],String displayText[])
    throws AlarmException
```

alarmRecordIds is the array of alarms to display.

String is the text to display in the *ackText* attribute field of the acknowledged alarm.

See Also: `AlarmRecordId`

setUnAckAlarms

Unacknowledge the alarms corresponding to the given alarm records.

```
public void setUnAckAlarms(AlarmRecordId alarmRecordIds[])
 throws AlarmException
```

alarmRecordIds is the array of alarms to unacknowledged.

See Also: `AlarmRecordId`

setUnClearAlarms

Unclears the alarms corresponding to the given alarm records.

```
public void setUnClearAlarms(AlarmRecordId alarmRecordIds[])
 throws AlarmException
```

alarmRecordIds is the array of alarms to unclear.

See Also: `AlarmRecordId`

setUnDisplayAlarms

Undisplays the alarms corresponding to the given alarm records.

```
public void setUnDisplayAlarms(AlarmRecordId alarmRecordIds[])
throws AlarmException
```

alarmRecordIds is the array of alarms to undisplay.

See Also: `AlarmRecordId`

sendClearAlarmsEvent

Sends a clear alarm event for the given alarm records.

```
public void sendClearAlarmsEvent(AlarmRecordId alarmRecordIds[])
    throws AlarmException
```

alarmRecordIds is the array of alarms to clear.

See Also: `AlarmRecordId`

deleteAlarms

Delete the given alarm records from the persistent store.

```
public void deleteAlarms(AlarmRecordId alarmRecordIds[])
    throws AlarmException
```

alarmRecordIds is the array of alarms to delete.

See Also: `AlarmRecordId`

addAlarmLogCreationListener

Registers for alarm log creation events.

```
public void addAlarmLogCreationListener
            (AlarmLogCreationListener l)throws AlarmException
```

l is the listener for `AlarmLogEvent` creation.

See Also: `AlarmLogCreationListener`

removeAlarmLogCreationListener

Deregisters for alarm log creation events.

```
public void removeAlarmLogCreationListener
            (AlarmLogCreationListener l) throws AlarmException
```

l is the listener for `AlarmLogEvent` creation to unregister.

See Also: `AlarmLogCreationListener`

addAlarmLogDeletionListener

Registers to receive an event when the alarm log is deleted.

```
public void addAlarmLogDeletionListener
            (AlarmLogDeletionListener l)throws AlarmException
```

l is the listener for `AlarmLogEvent` deletion to be removed.

See Also: `AlarmLogDeletionListener`

removeAlarmLogDeletionListener

Deregisters from receiving an event when the alarm log is deleted.

```
public void removeAlarmLogDeletionListener
           (AlarmLogDeletionListener l)throws AlarmException
```

l is the listener for `AlarmLogEvent` deletion to be removed.

See Also: `AlarmLogDeletionListener`

addAlarmLogModificationListener

Registers to receive an event when the alarm log is modified.

```
public void addAlarmLogModificationListener
           (AlarmLogModificationListener l)throws AlarmException
```

l is the listener for `AlarmLogEvent` modification.

See Also: `AlarmLogEvent`

removeAlarmLogModificationListener

Deregisters from receiving an event when the alarm log is modified.

```
public void removeAlarmLogModificationListener
           (AlarmLogModificationListener l)throws AlarmException
```

l is the listener for `AlarmLogEvent` modification to be removed.

See Also: `AlarmLogEvent`

addAlarmLogListener

Register to receive an event when the alarm log is modified.

```
public void addAlarmLogListener(AlarmLogListener l)
    throws AlarmException
```

l is the listener for `AlarmLogEvent`. Can be creation, deletion or modification.

See Also: `AlarmLogEvent`

removeAlarmLogListener

Deregisters to receive an event when the alarm log is modified.

```
public void removeAlarmLogListener(AlarmLogListener l)
    throws AlarmException
```

l is the listener for `AlarmLogEvent` to be removed.

See Also: `AlarmLogEvent`

3.11 `AlarmRecord` Class

public class `AlarmRecord`

extends `LogRecord` implements `Serializable`

The `com.sun.em.api.alarm.AlarmRecord` class implements the alarm record. It provides methods to retrieve the attribute values of an alarm record.

3.11.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.log.LogRecord
           |
           +----com.sun.em.api.alarm.AlarmRecord
```

Note – `LogRecord` is an empty class. Basically an `AlarmRecord` inherits from `LogRecord` for some basic attributes but right now this is an empty class.

3.11.2 Methods

getAttrSet

Returns the attributes that have been set in the `AlarmRecord`. This method is used in case of events. Some events do not contain all attributes. This flag indicates which attributes are present.

```
public AlarmRecordAttributeSet getAttrSet()
```

See Also: `AlarmRecordAttributeSet`

getLogName

Returns the container alarm log of the alarm record.

```
public LogName getLogName() throws AlarmAttributeNotSetException
```

Returns `LogName`.

Throws `AlarmAttributeNotSetException` if *LogName* was not set in the initial `AlarmRecordAttributeSet`.

getAckOperator

Returns the user ID of the operator who acknowledged the alarm.

```
public String getAckOperator()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *AckOperator* was not set in the initial `AlarmRecordAttributeSet`.

getAckState

Returns a boolean indicating whether the alarm has been acknowledged or not. If the boolean is set to true, the alarm has been acknowledged.

```
public boolean getAckState()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *AckState* was not set in the initial `AlarmRecordAttributeSet`.

getAckText

Returns any description entered by the operator who acknowledged the alarm.

```
public String getAckText() throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *AckText* was not set in the initial `AlarmRecordAttributeSet`.

getAdditionalText

Returns the additionalText attribute which is part of the AlarmRecord.

```
public String getAdditionalText()
     throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *AdditionalText* was not set in the initial `AlarmRecordAttributeSet`.

getAckTime

Returns the date and time when the alarm was acknowledged.

```
public Date getAckTime() throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *AckTime* was not set in the initial `AlarmRecordAttributeSet`.

getClearOperator

Returns the user ID of the operator who cleared the alarm.

```
public String getClearOperator()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *ClearOperator* was not set in the initial `AlarmRecordAttributeSet`.

getClearState

Returns a boolean indicating whether the alarm has been cleared or not. If the boolean is set to true, the alarm has been cleared.

```
public boolean getClearState()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *ClearState* was not set in the initial `AlarmRecordAttributeSet`.

getClearText

Returns any description entered by the operator when the alarm was cleared.

```
public String getClearText()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *ClearText* was not set in the initial `AlarmRecordAttributeSet`.

getClearTime

Returns the date and time when the alarm was cleared.

```
public Date getClearTime() throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *ClearTime* was not set in the initial `AlarmRecordAttributeSet`.

getDisplayOperator

Returns the user ID of the operator who changed the display state of the alarm.

```
public String getDisplayOperator()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *DisplayOperator* was not set in the initial `AlarmRecordAttributeSet`.

getDisplayState

Returns a boolean indicating whether the alarm displayed or not. If the boolean has been set to true, the alarm has been set to displayed.

```
public boolean getDisplayState()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *DisplayState* was not set in the initial `AlarmRecordAttributeSet`.

getDisplayText

Returns any description entered by the operator when the alarm display state is changed.

```
public String getDisplayText()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *DisplayText* was not set in the initial `AlarmRecordAttributeSet`.

getDisplayTime

Returns the date and time when the operator last changed the display state of the alarm.

```
public Date getDisplayTime()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *DisplayTime* was not set in the initial `AlarmRecordAttributeSet`.

getEventTime

Returns the date and time when the corresponding alarm was generated.

```
public Date getEventTime() throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *EventTime* was not set in the initial `AlarmRecordAttributeSet`.

getEventType

Returns the event type that created this alarm record.

```
public String getEventType()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *EventType* was not set in the initial `AlarmRecordAttributeSet`.

getLogRecordId

Returns the ID of this alarm record.

```
public AlarmRecordId getLogRecordId()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *LogRecordId* was not set in the initial `AlarmRecordAttributeSet`.

See Also: *AlarmRecordId*

getLoggingTime

Returns the date and time when the event was logged.

```
public Date getLoggingTime()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *loggingTime* was not set in the initial `AlarmRecordAttributeSet`.

getManagedObjectInstance

Returns the managed object instance of the device that generated the alarm.

```
public MOName getManagedObjectInstance()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if `ManagedObjectInstance` was not set in the initial `AlarmRecordAttributeSet`.

See Also: `MOName`

getPerceivedSeverity

Returns the severity of the alarm.

```
public EMSeverity getPerceivedSeverity()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *PerceivedSeverity* was not set in the initial `AlarmRecordAttributeSet`.

See Also: `EMSeverity`

getProbableCause

Returns the probable cause of the alarm.

```
public String getProbableCause()
    throws AlarmAttributeNotSetException
```

Throws `AlarmAttributeNotSetException` if *ProbableCause* was not set in the initial `AlarmRecordAttributeSet` .

toString

Returns a String representation of the fields that have been set in the alarm record. Note that this prints out only the fields that have been requested.

```
public String toString()
```

Overrides *toString* in class `Object`

See Also: `AlarmRecordAttributeSet`

3.12 `AlarmRecordAttribute` Class

public class `AlarmRecordAttribute`

extends `EMObjectAttribute` implements Serializable

The `com.sun.em.api.alarm.AlarmRecordAttribute` class represents a single attribute of the `AlarmRecord` object.

3.12.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMObjectAttribute
           |
           +----com.sun.em.api.alarm.AlarmRecordAttribute
```

Note – `EMObjectAttribute` is also being used in the Topology API. It is part of the common API directory. It represents any attribute of a given object. An attribute is represented by an ID.

3.12.2 Variables

ACK_OPERATOR

`AlarmRecordAttribute` for `ACK_OPERATOR` field

```
public static final AlarmRecordAttribute ACK_OPERATOR
```

See Also: `LogRecord`

ACK_STATE

`AlarmRecordAttribute` for `ACK_STATE` field

```
public static final AlarmRecordAttribute ACK_STATE
```

See Also: `LogRecord`

ACK_TEXT

`AlarmRecordAttribute` for `ACK_TEXT` field

```
public static final AlarmRecordAttribute ACK_TEXT
```

See Also: `LogRecord`

ACK_TIME

`AlarmRecordAttribute` for `ACK_TIME` field

```
public static final AlarmRecordAttribute ACK_TIME
```

See Also: `LogRecord`

CLEAR_OPERATOR

`AlarmRecordAttribute` for `CLEAR_OPERATOR` field

```
public static final AlarmRecordAttribute CLEAR_OPERATOR
```

See Also: `LogRecord`

CLEAR_STATE

`AlarmRecordAttribute` for `CLEAR_STATE` field

```
public static final AlarmRecordAttribute CLEAR_STATE
```

See Also: `LogRecord`

CLEAR_TEXT

`AlarmRecordAttribute` for `CLEAR_TEXT` field

```
public static final AlarmRecordAttribute CLEAR_TEXT
```

See Also: `LogRecord`

CLEAR_TIME

`AlarmRecordAttribute` for `CLEAR_TIME` field

```
public static final AlarmRecordAttribute CLEAR_TIME
```

See Also: `LogRecord`

DISPLAY_OPERATOR

`AlarmRecordAttribute` for `DISPLAY_OPERATOR` field

```
public static final AlarmRecordAttribute DISPLAY_OPERATOR
```

See Also: `LogRecord`

DISPLAY_STATE

`AlarmRecordAttribute` for `DISPLAY_STATE` field

```
public static final AlarmRecordAttribute DISPLAY_STATE
```

See Also: `LogRecord`

DISPLAY_TEXT

`AlarmRecordAttribute` for `DISPLAY_TEXT` field

```
public static final AlarmRecordAttribute DISPLAY_TEXT
```

See Also: `LogRecord`

DISPLAY_TIME

`AlarmRecordAttribute` for `DISPLAY_TIME` field

```
public static final AlarmRecordAttribute DISPLAY_TIME
```

See Also: `LogRecord`

EVENT_TIME

`AlarmRecordAttribute` for `EVENT_TIME` field

```
public static final AlarmRecordAttribute EVENT_TIME
```

See Also: `LogRecord`

EVENT_TYPE

AlarmRecordAttribute for `EVENT_TYPE` field

```
public static final AlarmRecordAttribute EVENT_TYPE
```

See Also: `LogRecord`

LOGGING_TIME

`AlarmRecordAttribute` for `LOGGING_TIME` field

```
public static final AlarmRecordAttribute LOGGING_TIME
```

See Also: `LogRecord`

LOG_RECORD_ID

`AlarmRecordAttribute` for `LOG_RECORD_ID` field

```
public static final AlarmRecordAttribute LOG_RECORD_ID
```

See Also: `LogRecord`, `LogRecordId`

MANAGED_OBJECT_INSTANCE

AlarmRecordAttribute for MANAGED_OBJECT_INSTANCE field

```
public static final AlarmRecordAttribute MANAGED_OBJECT_INSTANCE
```

See Also: LogRecord, MOName

PERCEIVED_SEVERITY

AlarmRecordAttribute for PERCEIVED_SEVERITY field

```
public static final AlarmRecordAttribute PERCEIVED_SEVERITY
```

See Also: LogRecord, EMSeverity

PROBABLE_CAUSE

AlarmRecordAttribute for PROBABLE_CAUSE field

```
public static final AlarmRecordAttribute PROBABLE_CAUSE
```

See Also: LogRecord

ADDITIONAL_TEXT

AlarmRecordAttribute for ADDITIONAL_TEXT field

```
public static final AlarmRecordAttribute ADDITIONAL_TEXT
```

See Also: LogRecord

LOG_NAME

`AlarmRecordAttribute` for `LOG_NAME` field

```
public static final AlarmRecordAttribute LOG_NAME
```

See Also: `LogRecord`, `LogName`

MIS_NAME

`AlarmRecordAttribute` for `MIS_NAME` field

```
public static final AlarmRecordAttribute MIS_NAME
```

See Also: `LogRecord`

3.12.3 Methods

toString

Returns a String representation of the `AlarmRecordAttribute`.

```
public String toString()
```

Returns the String representation.

Overrides *toString* in class `EMObjectAttribute`.

getColumnName

Returns a String representation of the `AlarmRecordAttribute`.

```
public String getColumnName()
```

Returns the String representation.

3.13 AlarmRecordAttributeSet Class

public class AlarmRecordAttributeSet

extends EMAttributeSet implements Cloneable, Serializable

The com.sun.em.api.alarm.AlarmRecordAttributeSet class implements an abstract class which forms the basis for the attributes set classes of each AlarmRecord. AlarmRecordAttributeSet is used in the Alarm API to communicate which attributes of the AlarmRecord must be returned.

3.13.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMAttributeSet
           |
           +----com.sun.em.api.alarm.AlarmRecordAttributeSet
```

3.13.2 Constructors

AlarmRecordAttributeSet

Creates an AlarmRecordAttributeSet which contains no attributes.

```
public AlarmRecordAttributeSet()
```

3.13.3 Methods

clone

Returns a clone of this `AlarmRecordAttributeSet` object.

```
public Object clone()
```

Returns a clone of this `AlarmRecordAttributeSet`.

Overrides `clone` in class `EMAttributeSet.`

elements

Returns an enumeration of the attributes in this set.

```
public Enumeration elements()
```

Overrides `elements` in class `EMAttributeSet.`

3.14 `AlarmRecordId` Class

public class `AlarmRecordId`

Extends `Object` implements `Cloneable` and `Serializable`.

An `alarmRecord` is uniquely identified with its `AlarmRecordId` and the `LogName` associated to it. The `com.sun.em.api.alarm.AlarmRecordId` is a unique identifier.

3.14.1 Constructors

AlarmRecordId

Construct using an Integer identifier.

```
public AlarmRecordId(int alarmId)
```

3.14.2 Methods

getAlarmFullName

Returns the name of the `logRecordId` prepended with the choice String.

```
public String getAlarmFullName()
```

clone

```
public Object clone()
```

Overrides `clone` in class `Object`.

toString

Returns the name of the `logRecordId`.

```
public String toString()
```

Overrides `toString` in class `Object`.

equals

Returns true if *obj* matches self, otherwise returns false.

```
public boolean equals(Object obj)
```

Overrides `equals` in class `Object`.

3.15 `Filter` Class

public class `Filter`

extends `Object` implements `Serializable`

The `com.sun.em.api.alarm.Filter` class implements an alarm filter. An alarm filter is constructed by combining `LogicalCriteria` with a list of `FilterItem`.

See Also: `LogicalCriteria`, `FilterItem`

3.15.1 Constructors

Filter

This is the class constructor. An alarm filter is constructed by combining a `LogicalCriteria` with a *FilterItem*.

```
public Filter(LogicalCriteria op,FilterItem filterItem)
```

Filter

This is the class constructor. An alarm filter is constructed by combining the default logical operator with the *filterItem*. The default logical operator is `AND`.

```
public Filter(FilterItem filterItem)
```

3.15.2 Methods

setLogicalCriteria

Specifies the default logical operator to use in the query.

```
public void setLogicalCriteria(LogicalCriteria op)
```

getLogicalCriteria

Get the default logical operator used in the query.

```
public LogicalCriteria getLogicalCriteria()
```

logicalOp is the *LogicalCriteria*.

addFilterItem

Adds a filter element to the query.

```
public void addFilterItem(FilterItem filterItem)
```

See Also: `FilterItem`

removeFilterItem

Removes a filter element from the query.

```
public boolean removeFilterItem(FilterItem filterItem)
```

See Also: `FilterItem`

clear

Clears the query by removing all filter elements.

```
public void clear()
```

elements

Returns an enumeration of the filter elements present in the query.

```
public Enumeration elements()
```

toString

Returns a String representation of the query.

```
public String toString()
```

Overrides `toString` in class `Object`.

3.16 `FilterItem` Class

public class `FilterItem`

extends Object implements Serializable

The `com.sun.em.api.alarm.FilterItem` class implements a `FilterItem` that represents a condition in a query. A condition comprises an attribute name, a relational operator, and one or more attribute values. Conditions are combined with a logical operator (`AND` or `OR`) to build a query.

3.16.1 Constructors

FilterItem

Constructs a `FilterItem` to filter alarms based on alarm severity attribute.

```
public FilterItem
    (AlarmRecordAttribute name,RelationCriteria rel,
                                        EMSeverity value[])
    throws AlarmException
```

name is the `AlarmRecordAttribute.PERCEIVED_SEVERITY`.

rel is the `RelationCriteria`.

See Also: `AlarmRecordAttribute`, `RelationCriteria`, `EMSeverity`

FilterItem

Constructs a `FilterItem` to filter alarms based on event time, log time, clear time, acknowledge time, or display time attribute.

```
public FilterItem
        (AlarmRecordAttribute name,RelationCriteria rel,Date value)
    throws AlarmException
```

name is the `AlarmRecordAttribute`: `eventTime`, `loggingTime`, `clearTime`, `ackTime`, `displayTime`.

rel is the `RelationCriteria`.

value is the date to query on.

See Also: `AlarmRecordAttribute`, `RelationCriteria`

FilterItem

Constructs a FilterItem to filter alarms based on the Managed Object Name attribute.

```
public FilterItem(AlarmRecordAttribute name,RelationCriteria
                    rel,MOName value[])
    throws AlarmException
```

name is the AlarmRecordAttribute.MANAGED_OBJECT_INSTANCE.

rel is the RelationCriteria: EQUAL or NOT_EQUAL.

value is the list of MOName to query on.

See Also: AlarmRecordAttribute, RelationCriteria, MOName

FilterItem

Constructs a FilterItem to filter alarms based on any String-format attribute, (should be part of DISPLAY_OPERATOR, ACK_OPERATOR, CLEAR_OPERATOR, DISPLAY_TEXT, ACK_TEXT, or CLEAR_TEXT).

```
public FilterItem
    (AlarmRecordAttribute name,RelationCriteria rel,String value[])
    throws AlarmException
```

rel is the RelationCriteria: EQUAL or NOT_EQUAL.

value is the list of value in String format.

See Also: AlarmRecordAttribute, RelationCriteria

FilterItem

Constructs a `FilterItem` to filter alarms based on the Log Record ID.

```
public FilterItem
    (AlarmRecordAttribute name,RelationCriteria rel,
                                    AlarmRecordId value[])
    throws AlarmException
```

name is the `AlarmRecordAttribute.LOG_RECORD_ID`.

rel is the `RelationCriteria`.

value is the list of `AlarmRecordIds`.

See Also: `AlarmRecordAttribute`, `RelationCriteria`, `AlarmRecordId`

FilterItem

Constructs a `FilterItem` to filter alarms based on the acknowledge, clear, or display state.

```
public FilterItem
    (AlarmRecordAttribute name,RelationCriteria rel,
                                    Boolean value)
    throws AlarmException
```

name is the `AlarmRecordAttribute`: `ACK_STATE`, `CLEAR_STATE`, or `DISPLAY_STATE`.

rel is the `RelationCriteria`.

value is the boolean value.

See Also: `AlarmRecordAttribute`, `RelationCriteria`

3.16.2 Methods

getAttrName

Returns the attribute name of the `FilterItem`.

```
public AlarmRecordAttribute getAttrName()
```

See Also: `AlarmRecordAttribute`

getRelation

Returns the relational operator of the `FilterItem`.

```
public RelationCriteria getRelation()
```

See Also: `RelationCriteria`

getAttrValue

Returns an array of the attribute values of the `FilterItem`.

```
public Object[] getAttrValue()
```

toString

Returns a String representation of the attribute name, the relation operator, and the attribute value(s).

```
public String toString()
```

Overrides `toString` in class `Object`.

3.17 GenericQuery Class

public class GenericQuery

extends Object implements Serializable, Query

The com.sun.em.api.alarm.GenericQuery class implements a generic alarm filter or query. An alarm filter is constructed by combining LogicalCriteria with a Filter.

3.17.1 Constructors

GenericQuery

The default class constructor using the AND logical operator for the query.

```
public GenericQuery()
```

GenericQuery

The class constructor. A generic query is constructed by combining a LogicalCriteria with a filter.

```
public GenericQuery(LogicalCriteria op,Filter filter)
```

See Also: LogicalCriteria, Filter

GenericQuery

The class constructor. A generic query is constructed by combining the default logical operator with the filter. The default logical operator is AND.

```
public GenericQuery(Filter filter)
```

See Also: Filter

3.17.2 Methods

addFilter

Adds a filter element to the query.

```
public void addFilter(Filter filter)
```

See Also: `FilterItem`

setLogicalCriteria

Specifies the default logical operator to use in the query.

```
public void setLogicalCriteria(LogicalCriteria log)
```

See Also: `LogicalCriteria`

getLogicalCriteria

Gets the default logical operator used in the query.

```
public LogicalCriteria getLogicalCriteria()
```

Returns the `LogicalCriteria`.

removeFilter

Removes a filter element from the query.

```
public boolean removeFilter(Filter filter)
```

See Also: `Filter`

clear

Clears the query by removing all filter elements.

```
public void clear()
```

elements

Returns an enumeration of the filter elements present in the query.

```
public Enumeration elements()
```

toString

Returns a String representation of the query.

```
public String toString()
```

Overrides `toString` in class `Object`.

3.18 `LogicalCriteria` Class

public class `LogicalCriteria`

extends `EMObjectAttribute` implements Serializable

The `com.sun.em.api.alarm.LogicalCriteria` class implements the logical operator used to combine the `Filter` conditions. The logical operators are `AND` and `OR`.

3.18.1 Variables

OR

Logical Criteria can be OR.

```
public static final LogicalCriteria OR
```

AND

Logical Criteria can be AND.

```
public static final LogicalCriteria AND
```

3.18.2 Methods

toString

Gets the String representation of the logical operator indexed by ID.

```
public String toString()
```

Returns the String representation of the logical operator.

Overrides toString in class EMObjectAttribute.

3.19 LogName Class

public class LogName

extends Object implements Serializable

The com.sun.em.api.alarm.LogName class implements the name of the alarm log.

3.19.1 Variables

STRING

```
public static final String STRING
```

NUMBER

```
public static final String NUMBER
```

3.19.2 Constructors

LogName

Creates an instance of LogName by specifying the MIS name and the log name as a String format.

```
public LogName(String mis,String log)
```

mis is the name of the MIS that contains the log.

log is the name of the log. The log name is a choice between a String and an Integer format. This example is the String format.

LogName

Creates an instance of `LogName` by specifying the MIS name and the log name as an Integer format.

```
public LogName(String mis,Integer log)
```

mis is the name of the MIS that contains the log.

log is the name of the log. The `LogName` is a choice between a String and an Integer format. This example is the Integer format.

LogName

```
public LogName(String mis,int log)
```

3.19.3 Methods

getMisName

Returns the name of the MIS of the alarm log.

```
public String getMisName()
```

getLogName

Returns the name of the alarm log.

```
public Object getLogName()
```

getFullLogName

Returns the name of the alarm log as a fully distinguished name (fdn).

```
public String getFullLogName()
```

equals

Tests for equality of alarm logs.

```
public boolean equals(Object obj)
```

Returns true if *obj* matches self, otherwise returns false.

Overrides *equals* in class `Object`.

toString

Returns the name of the log in String format. This does not contain the MIS name.

```
public String toString()
```

Overrides `toString` in class `Object`.

3.20 Query Interface

public interface `Query`

The `com.sun.em.api.alarm.Query` interface is empty.

See Also: `GenericQuery`

3.21 RelationCriteria Class

public class RelationCriteria

extends EMObjectAttribute implements Serializable

The com.sun.em.api.alarm.RelationCriteria class implements a RelationCriteria or relation operator. RelationCriteria is used in building a FilterItem.

See Also: FilterItem

3.21.1 Variables

EQUAL

RelationCriteria can be EQUAL.

```
public static final RelationCriteria EQUAL
```

NOT_EQUAL

RelationCriteria can be NOT_EQUAL.

```
public static final RelationCriteria NOT_EQUAL
```

GREATER_THAN

RelationCriteria can be GREATER_THAN.

```
public static final RelationCriteria GREATER_THAN
```

GREATER_THAN_OR_EQUAL

`RelationCriteria` can be `GREATER_THAN_OR_EQUAL`.

```
public static final RelationCriteria GREATER_THAN_OR_EQUAL
```

LESS_THAN

`RelationCriteria` can be `LESS_THAN`.

```
public static final RelationCriteria LESS_THAN
```

LESS_THAN_OR_EQUAL

`RelationCriteria` can be `LESS_THAN_OR_EQUAL`.

```
public static final RelationCriteria LESS_THAN_OR_EQUAL
```

3.21.2 Methods

toString

Returns a String representation of the relational operator.

```
public String toString()
```

Overrides `toString` in class `EMObjectAttribute`.

CHAPTER 4

Java Topology API

This chapter descibes the classes that make up the Java Topology Application Programming Interface.

This chapter comprises the following topics:

4.1 Overview

For information on how to use the Java Topology API, refer to the *Developing Java Applications* book.

4.2 `EMAgent` Class

public abstract class `EMAgent`

extends `EMObject`

4.2.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.topology.EMObject
           |
           +----com.sun.em.api.topology.EMAgent
```

4.2.2 Constructors

EMAgent

```
public EMAgent()
```

4.2.3 Methods

getOperationalState

```
public abstract EMAgentOperationalState getOperationalState()
```

Returns the `EMAgentAttribute.OPERATIONAL_STATE` attribute.

See Also: `OPERATIONAL_STATE`

getAdministrativeState

Returns the `EMAgentAttribute.ADMINISTRATIVE_STATE` attribute.

```
public abstract EMAgentAdministrativeState
                                    getAdministrativeState()
```

See Also: `ADMINISTRATIVE_STATE`

setAdministrativeState

Sets the `EMAgentAttribute.ADMINISTRATIVE_STATE` attribute.

```
public abstract void setAdministrativeState
                   (EMAgentAdministrativeState administrativeState)
```

administrativeState is the administrative state.

See Also: `ADMINISTRATIVE_STATE`

4.3 EMAgentAdministrativeState Class

public class EMAgentAdministrativeState

extends Object implements Serializable

The com.sun.em.api.topology.EMAgentAdministrativeState class represents the possible administrative states of a proxy agent. It is the same as the administrativeState attribute defined in [ISO10165-2].

4.3.1 Variables

LOCKED

The LOCKED state means that the proxy must not perform, or suspend performing, proxy activities on behalf of the Internet agent.

```
public static final EMAgentAdministrativeState LOCKED
```

UNLOCKED

The UNLOCKED state means that proxy must continue to perform, or resume performing, proxy activities on behalf of the Internet agent.

```
public static final EMAgentAdministrativeState UNLOCKED
```

SHUTTING_DOWN

```
public static final EMAgentAdministrativeState SHUTTING_DOWN
```

4.3.2 Methods

equals

Compares this `EMAgentAdministrativeState` against the specified object. The result is true if, and only if, the argument is not null and is a `EMAgentAdministrativeState` object with the same value as this `EMAgentAdministrativeState`.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, otherwise returns false.

Overrides *equals* in class `Object`

toString

Returns a string representation of the `EMAgentAdministrativeState`

```
public String toString()
```

Returns the string representation.

Overrides `toString` in class `Object`

4.4 EMAgentAttribute Class

public class `EMAgentAttribute`

extends `EMTopoAttribute` implements `Serializable`

The `com.sun.em.api.topology.EMAgentAttribute` class represents a single attribute of the `EMAgent` persistent object class (POC).

4.4.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMObjectAttribute
           |
           +----com.sun.em.api.topology.EMTopoAttribute
                   |
                   +----com.sun.em.api.topology.EMAgentAttribute
```

4.4.2 Variables

OPERATIONAL_STATE

The `OPERATIONAL_STATE` attribute indicates the perceived state of the agent. It is the same as the `operationalState` attribute defined in ISO10165-2.

```
public static final EMAgentAttribute OPERATIONAL_STATE
```

ADMINISTRATIVE_STATE

The `ADMINISTRATIVE_STATE` attribute is used to suspend and resume the proxy activity relative to the agent. It is the same as the `administrativeState` attribute defined in ISO10165-2.

```
public static final EMAgentAttribute ADMINISTRATIVE_STATE
```

4.4.3 Methods

toString

Returns a String representation of the EMAgentAttribute.

```
public String toString()
```

Returns the String representation.

Overrides toString in class EMTopoAttribute.

4.5 EMAgentOperationalState Class

public class EMAgentOperationalState

extends Object implements Serializable

The com.sun.em.api.topology.EMAgentOperationalState class represents the possible operational states of a proxy agent. It is the same as the operationalState attribute defined in ISO10165-2.

4.5.1 Variables

DISABLED

The DISABLED state means that the agent is not operational, as perceived by the proxy, that is, it cannot be reached.

```
public static final EMAgentOperationalState DISABLED
```

ENABLED

The ENABLED state means that the Internet agent is operational as perceived by the proxy, that is, it can be reached.

```
public static final EMAgentOperationalState ENABLED
```

4.5.2 Methods

equals

Compares this EMAgentOperationalState against the specified object. The result is true if and only if the argument is not null and is a EMAgentOperationalState object, with the same value as this EMAgentOperationalState.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, otherwise returns false.

Overrides *equals* in class Object.

toString

Returns a String representation of the EMAgentOperationalState.

```
public String toString()
```

Returns the String representation.

Overrides *toString* in class Object.

4.6 EMAttributeDecodeException Class

public class `EMAttributeDecodeException`

extends `RuntimeException`

The `com.sun.em.api.topology.EMAttributeDecodeException` class is the exception returned when an error occured in converting the attribute values received in ASN.1 format into the local cache data format.

4.6.1 Inheritance

```
java.lang.Object
   |
   +--java.lang.Throwable
           |
           +--java.lang.Exception
                   |
                   +--java.lang.RuntimeException
                           |
                           +-com.sun.em.api.topology.EMAttributeDecodeException
```

4.6.2 Constructors

EMAttributeDecodeException

Constructs a new `EMAttributeDecodeException` with the specified detail message.

```
public EMAttributeDecodeException(String message)
```

message is the detail message.

EMAttributeDecodeException

Constructs a new EMAttributeDecodeException with no detail message.

```
public EMAttributeDecodeException()
```

4.7 EMAttributeEncodeException Class

public class EMAttributeEncodeException

extends RuntimeException

The com.sun.em.api.topology.EMAttributeEncodeException class is used for exceptions returned when an error occured in converting the attribute values to ASN.1 for transmission to the MIS.

4.7.1 Inheritance

```
java.lang.Object
   |
   +--java.lang.Throwable
           |
           +--java.lang.Exception
                   |
                   +--java.lang.RuntimeException
                           |
                           +-com.sun.em.api.topology.EMAttributeEncodeException
```

4.7.2 Constructs

EMAttributeEncodeException

Constructs a new EMAttributeEncodeException with the specified detail message.

```
public EMAttributeEncodeException(String message)
```

message is the detail message.

EMAttributeEncodeException

Constructs a new EMAttributeEncodeException with no detail message.

```
public EMAttributeEncodeException
```

4.8 EMAttributeNotCreatableException Class

public class EMAttributeNotCreatableException

extends RuntimeException

The com.sun.em.api.topology.EMAttributeNotCreatableException class is thrown if the method createWithSomeAttributes() of a persistent object class (POC) attempts to set the initial value of an attribute, which is not allowed to be set at creation time. This exception is a RuntimeException since it is caused by a programming error.

4.8.1 Inheritance

```
java.lang.Object
   |
   +--java.lang.Throwable
           |
           +--java.lang.Exception
                   |
                   +--java.lang.RuntimeException
                       |
                        +-com.sun.em.api.topology.EMAttributeNotCreatableException
```

4.8.2 Constructs

EMAttributeNotCreatableException

Constructs a new `EMAttributeNotCreatableException` with the specified detail message.

```
public EMAttributeNotCreatableException(String message)
```

message is the detail message.

EMAttributeNotCreatableException

Constructs a new `EMAttributeNotCreatableException` with no detail message.

```
public EMAttributeNotCreatableException()
```

4.9 EMAttributeNotSetException Class

public class EMAttributeNotSetException

extends RuntimeException

The com.sun.em.api.topology.EMAttributeNotSetException class is used for exceptions returned when an attribute of a persistent object class (POC) instance is being accessed but has not been set yet in the cache, or when create or store is invoked without certain mandatory attribute(s) being set.

4.9.1 Inheritance

```
java.lang.Object
   |
   +--java.lang.Throwable
           |
           +--java.lang.Exception
                   |
                   +--java.lang.RuntimeException
                           |
                           +-com.sun.em.api.topology.EMAttributeNotSetException
```

4.9.2 Constructs

EMAttributeNotSetException

Constructs a new EMAttributeNotSetException with the specified detail message.

```
public EMAttributeNotSetException(String message)
```

message is the detail message.

EMAttributeNotSetException

Constructs a new EMAttributeNotSetException with no detail message.

```
public EMAttributeNotSetException()
```

4.10 EMAttributeNotStoreableException Class

public class EMAttributeNotStoreableException

extends RuntimeException

The com.sun.em.api.topology.EMAttributeNotStoreableException class is thrown if the method storeWithSomeAttributes(), of a Persistent Object Class (POC), attempts to store a read-only attribute. This exception is a RuntimeException since it is caused by a programming error.

4.10.1 Inheritance

```
java.lang.Object
   |
   +--java.lang.Throwable
           |
           +--java.lang.Exception
                   |
                   +--java.lang.RuntimeException
                           |
                       +-com.sun.em.api.topology.EMAttributeNotStoreableException
```

4.10.2 Constructs

EMAttributeNotStoreableException

Constructs a new `EMAttributeNotStoreableException` with the specified detail message.

```
public EMAttributeNotStoreableException(String message)
```

message is the detail message.

EMAttributeNotStoreableException

Constructs a new `EMAttributeNotStoreableException` with no detail message.

```
public EMAttributeNotStoreableException()
```

4.11 EMCmipAgent Class

public class `EMCmipAgent`

extends `EMObject` implements `Cloneable`

An instance of the `com.sun.em.api.topology.EMCmipAgent` class represents the MIS object which contains configuration information for a CMIP agent. The configuration information includes the CMIP MPA hostname and port number, a list of managed objects DNs, a network SAP, a transport selector, a presentation selector, a session selector, and an application entity title (AET).

Note – This class does not provide an interface to the agent's managed objects, but only to Solstice EM's configuration information for the agent.

4.11.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.topology.EMObject
           |
           +----com.sun.em.api.topology.EMCmipAgent
```

4.11.2 Constructors

EMCmipAgent

Constructs an instance of `EMCmipAgent` representing a particular `CMIPagent` configuration object.

```
public EMCmipAgent(EMTopoPlatform platform, EMCmipAgentDn dn)
```

platform is the Topology API platform.

dn is the unique identifier of the `CMIPagent` configuration object.

4.11.3 Methods

getCmipAgentFromManagedObject

Returns the unique identifier of the CMIP agent configuration object whose management domain includes the specified managed object.

```
public static EMCmipAgentDn getCmipAgentFromManagedObject
           (EMTopoPlatform platform,String managedObjectDnSlashForm)
    throws EMTopoServiceException
```

platform is the Topology API platform.

`managedObjectDnSlashForm` is the fully distinguished name (fdn) of the managed object.

Returns the CMIP agent configuration object unique identifier; null if the managed object is not managed by a CMIP agent.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

clearAllAttributes

Clears the cached values for all object attributes, resetting them back to their default values (usually null or empty string or 0). This includes the *EMCmipAgent.DN* attribute which associates the `EMCmipAgent` instance with a particular CMIP agent object in the MIS. This method is useful when you want to reuse the `EMCmipAgent` instance to access a different CMIP agent object and don't want the previous values to remain in effect.

```
public void clearAllAttributes()
```

Overrides *clearAllAttributes* in class `EMObject`.

clearSomeAttributes

Clears the cached values for the specified object attributes, resetting them back to their default values (usually null or empty string or 0). This includes the *EMCmipAgent.DN* attribute which associates the `EMCmipAgent` instance with a particular CMIP agent object in the MIS. This method is useful when you want to reuse a `EMCmipAgent` instance to access a different proxy agent object and do not want the previous values to remain in effect.

```
public void clearSomeAttributes(EMAttributeSet attributes)
```

attributes is the object attributes to have their cached values cleared.

Overrides *clearSomeAttributes* in class `EMObject`.

clone

Return a clone of this `EMCmipAgent` object.

```
public Object clone()
```

Returns a clone of this `EMCmipAgent`.

Overrides `clone` in class `Object`.

createWithAllAttributes

Creates a new CMIP agent object in the MIS, storing all active attribute values in the object. Any attribute which was not given a value will take on a default value defined by the GDMO (generally null or empty string). In order for the create to succeed, the following attributes must be set:

`EMTopoAttribute.DN`

`EMAgentAttribute.ADMINISTRATIVE_STATE`

`EMCmipAgentAttribute.MPA_ADDRESS_INFO`

`EMCmipAgentAttribute.AGENT_ADDRESS_INFO`

`EMCmipAgentAttribute.MANAGED_OBJECTS`

`EMCmipAgentAttribute.APPLICATION_ENTITY_TITLE`

`EMCmipAgentAttribute.PRESENTATION_SELECTOR`

`EMCmipAgentAttribute.SESSION_SELECTOR`

`EMCmipAgentAttribute.TRANSPORT_SELECTOR`

`EMCmipAgentAttribute.NETWORK_SAP`

The following attributes cannot be set at creation time, and therefore are ignored regardless of whether they have a cached value:

`EMAgentAttribute.OPERATIONAL_STATE`

These attributes are ignored even if they have cached values.

```
public void createWithAllAttributes()
    throws EMTopoServiceException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the any of the mandatory attributes are not set.

Overrides `createWithAllAttributes` in class `EMObject`.

`createWithSomeAttributes`

Creates a new object in the MIS, storing a subset of the active attribute values in the object. Only attributes specified in the parameter *attributes* will be stored in the new object, and then only if the attribute is active (that is, it has been given a value). All other attributes will be given a default value defined by the GDMO (generally null or empty string). In order for the create to succeed, the following attributes must be set and must be members of the `EMAttributeSet` attributes:

`EMTopoAttribute.DN`

`EMAgentAttribute.ADMINISTRATIVE_STATE`

`EMCmipAgentAttribute.MPA_ADDRESS_INFO`

`EMCmipAgentAttribute.AGENT_ADDRESS_INFO`

`EMCmipAgentAttribute.MANAGED_OBJECTS`

`EMCmipAgentAttribute.APPLICATION_ENTITY_TITLE`

`EMCmipAgentAttribute.PRESENTATION_SELECTOR`

`EMCmipAgentAttribute.SESSION_SELECTOR`

`EMCmipAgentAttribute.TRANSPORT_SELECTOR`

`EMCmipAgentAttribute.NETWORK_SAP`

The following attributes cannot be set at creation time, and therefore are ignored regardless of whether they have a cached value:

`EMAgentAttribute.OPERATIONAL_STATE`

If any of these attributes are members of the `EMAttributeSet` attributes, then an `EMAttributeNotCreatableException` will be thrown.

```
public void createWithSomeAttributes(EMAttributeSet attributes
    throws EMTopoServiceException
```

attributes is subset of the `EMCmipAgent`'s attributes to store in the new object.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if any of the mandatory attributes are not set.

`EMAttributeNotCreatableException` if attributes contains any attributes which cannot be set at creation time.

Overrides `createWithAllAttributes` in class `EMObject`.

destroy

Deletes the object identified by `EMCmipAgent.DN` from the MIS. This is a permanent, non-reversible operation, so some care should be taken when using this method.

```
public void destroy() throws EMTopoServiceException,
    EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMCmipAgent.DN` does not exist.

Overrides *destroy* in class `EMObject`.

differences

Compares this EMCmipAgent against the specified EMCmipAgent and returns either the set of attributes for which the EMCmipAgent objects have different values or null if the EMCmipAgent objects are equal. If the argument *obj* is null or not an instance of EMCmipAgent, then all active attributes of this EMCmipAgent are considered to be differences. Otherwise, an attribute has differing values if the attribute is active for one EMCmipAgent but not the other, or if the attribute is active for both EMCmipAgent objects but the values of the attribute are not equal.

```
public EMAttributeSet differences(EMObject obj)
```

obj is the object to compare against.

Returns the set of attributes for which the EMCmipAgent objects have different values or null if the EMCmipAgent objects are equal.

Overrides *differences* in class EMObject.

differencesSubset

Compares this EMCmipAgent only for those attributes specified by the second input parameter, and returns the set of attributes for which the EMCmipAgents have different values or null if the EMCmipAgents are equal. If the argument *obj* is null or not an instance of EMCmipAgent, then all active attributes of this EMCmipAgent are considered to be differences. Otherwise, an attribute has differing values if the attribute is active for one EMCmipAgent but not the other, or if the attribute is active for both EMCmipAgents but the values of the attribute are not equal.

```
public EMAttributeSet differencesSubset
                          (EMObject obj,EMAttributeSet attributes)
```

obj is the object to compare against.

attributes is the set of attributes to compare.

Returns the set of attributes for which the EMCmipAgent objects have different values or null if the EMCmipAgent objects are equal.

Overrides *differencesSubset* in class EMObject.

equals

Compares this `EMCmipAgent` against the specified object. If the argument *obj* is null or not an instance of `EMCmipAgent`, then the two objects are not equal. Otherwise, if the two `EMCmipAgent` objects have the same set of active attributes, and the same value for each active attribute, then the `EMCmipAgent` objects are equal.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, otherwise returns false.

Overrides *equals* in class `Object`.

equalsSubset

Compares this `EMCmipAgent` against the specified object only for those attributes specified by the second parameter, restricting the equality check to the specified attributes. If the argument *obj* is null or not an instance of `EMCmipAgent`, then the two objects are not equal. Otherwise, if the two `EMCmipAgents` have the same set of active attributes out of the specified attributes, and the same value for each active attribute, then the `EMCmipAgents` are equal.

```
public boolean equalsSubset(EMObject obj,EMAttributeSet attributes)
```

obj is the object to compare against.

attributes is the subset of attributes.

Returns true if the objects are equal, otherwise returns false.

Overrides *equalsSubset* in class `EMObject`.

exists

Checks to see if the object identified by `EMCmipAgent.DN` exists.

```
public boolean exists() throws EMTopoServiceException
```

Returns true if the object exists.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

Overrides exists in class `EMObject`.

getActiveAttributes

Returns the set of object attributes which have been given a value.

```
public EMAttributeSet getActiveAttributes()
```

Returns the set of active attributes.

Overrides *getActiveAttributes* in class `EMObject`.

loadAllAttributes

Loads all attributes of the object identified by *EMCmipAgent.DN* from the MIS into the object's attribute cache. These attributes are now considered to be active and can be retrieved with the appropriate getter methods.

```
public void loadAllAttributes()throws EMTopoServiceException,
    EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMCmipAgent.DN` does not exist.

`EMAttributeDecodeException` if an error occured in converting the attribute values received in ASN.1 format into the local cache data format.

Overrides `loadAllAttributes` in class `EMObject`.

loadSomeAttributes

Loads the specified attributes of the object identified by `EMCmipAgent.DN` from the MIS into the object's attribute cache. These attributes are now considered to be active and can be retrieved with the appropriate get methods.

```
public void loadSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException, EMUnknownObjectException
```

attributes is the object attributes to load from the MIS.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the *EMTopoAttribute.DN* attribute is not set.

`EMUnknownObjectException` if the object *EMCmipAgent.DN* does not exist.

`EMAttributeDecodeException` if an error occured in converting the attribute values received in ASN.1 format into the local cache data format.

Overrides `loadSomeAttributes` in class `EMObject`.

newInstance

Returns a new instance of the `EMCmipAgent` class without any attributes set.

```
public EMObject newInstance()
```

Returns the new object.

Overrides `newInstance` in class `EMObject`.

storeAllAttributes

Stores all attributes into the object identified by *EMCmipAgent.DN*. For attributes which have not been given a value by calling load or setter methods, a default value, (usually null or empty string or 0) will be stored.

The `EMAgentAttribute.OPERATIONAL_STATE` attribute is read-only and therefore cannot be stored in the MIS. This attribute is ignored even if they have cached values.

```
public void storeAllAttributes() throws
EMTopoServiceException,EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the *EMTopoAttribute.DN* attribute is not set.

`EMUnknownObjectException` if the object *EMCmipAgent.DN* does not exist.

`EMAttributeEncodeException` if an error ocurred in converting the attribute values to ASN.1 for transmission to MIS.

Overrides *storeAllAttributes* in class `EMObject`.

storeSomeAttributes

Stores the specified attributes into the object identified by `EMCmipAgent.DN`. For attributes which have not been given a value by calling load or setter methods, a default value (usually null or empty string or 0) will be stored.

The `EMAgentAttribute.OPERATIONAL_STATE` attribute is read-only and therefore cannot be stored in the MIS. If this attribute is a member of the `EMAttributeSet` attributes, then an `EMAttributeNotStoreableException` will be thrown.

```
public void storeSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException, EMUnknownObjectException
```

attributes is the object attributes to store in the MIS.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the *EMTopoAttribute.DN* attribute is not set.

`EMUnknownObjectException` if the object *EMCmipAgent.DN* does not exist.

`EMAttributeNotStoreableException` if attributes contains any attributes which cannot be set at creation time.

`EMAttributeEncodeException` if an error occured in converting the attribute values to ASN.1 for transmission to MIS.

Overrides `storeSomeAttributes` in class `EMObject`.

getAdministrativeState

Returns the `EMAgentAttribute.ADMINISTRATIVE_STATE` attribute.

```
public EMAgentAdministrativeState getAdministrativeState()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `ADMINISTRATIVE_STATE`

getAgentAddressInfo

Returns the `EMCmipAgentAttribute.AGENT_ADDRESS_INFO` attribute.

```
public String getAgentAddressInfo()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `AGENT_ADDRESS_INFO`

getAgentAddressTag

Returns the *E*MCmipAgentAttribute.AGENT_ADDRESS_TAG attribute.

```
public Integer getAgentAddressTag()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: AGENT_ADDRESS_TAG

getApplicationEntityTitle

Returns the EMCmipAgentAttribute.APPLICATION_ENTITY_TITLE attribute.

```
public String getApplicationEntityTitle()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: APPLICATION_ENTITY_TITLE

getDn

Returns the EMTopoAttribute.DN attribute.

```
public EMObjectDn getDn()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

Overrides *getDn* in class EMObject.

See Also: *DN*

getMpaAddressInfo

Returns the EMCmipAgentAttribute.MPA_ADDRESS_INFO attribute.

```
public EMCmipAgentMpaAddressInfo getMpaAddressInfo()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: MPA_ADDRESS_INFO

getManagedObjects

Returns the EMCmipAgentAttribute.MANAGED_OBJECTS attribute.

```
public String[] getManagedObjects()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: MANAGED_OBJECTS

getNetworkSAP

Returns the EMCmipAgentAttribute.NETWORK_SAP attribute.

```
public String getNetworkSAP()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: NETWORK_SAP

getOperationalState

Returns the `EMAgentAttribute.OPERATIONAL_STATE` attribute.

```
public EMAgentOperationalState getOperationalState()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `OPERATIONAL_STATE`

getPresentationSelector

Returns the `EMCmipAgentAttribute.PRESENTATION_SELECTOR` attribute.

```
public String getPresentationSelector()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: *PRESENTATION_SELECTOR*

getSessionSelector

Returns the `EMCmipAgentAttribute.SESSION_SELECTOR` attribute.

```
public String getSessionSelector()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `SESSION_SELECTOR`

getTransportSelector

Returns the `EMCmipAgentAttribute.TRANSPORT_SELECTOR` attribute.

```
public String getTransportSelector()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `TRANSPORT_SELECTOR`

setAdministrativeState

Sets the `EMAgentAttribute.ADMINISTRATIVE_STATE` attribute.

```
public void setAdministrativeState
               (EMAgentAdministrativeState administrativeState)
```

administrativeState is the attribute value.

Throws `NullPointerException` if *administrativeState* is null.

See Also: `ADMINISTRATIVE_STATE`

setAgentAddressInfo

Sets the `EMCmipAgentAttribute.AGENT_ADDRESS_INFO` attribute.

```
public void setAgentAddressInfo(String agentAddressInfo)
```

agentAddressInfo is the attribute value.

Throws `NullPointerException` if *agentAddressInfo* is null.

See Also: *A*`GENT_ADDRESS_INFO`

setAgentAddressTag

Sets the `EMCmipAgentAttribute.AGENT_ADDRESS_TAG` attribute.

```
public void setAgentAddressTag(Integer agentAddressTag)
```

agentAddressTag is the attribute value.

Throws `NullPointerException` if *agentAddressTag* is null.

See Also: `AGENT_ADDRESS_TAG`

setApplicationEntityTitle

Sets the `EMCmipAgentAttribute.APPLICATION_ENTITY_TITLE` attribute.

```
public void setApplicationEntityTitle(String applicationEntityTitle)
```

applicationEntityTitle is the attribute value.

Throws `NullPointerException` if applicationEntityTitle is null.

See Also: `APPLICATION_ENTITY_TITLE`

setDn

Sets the `EMTopoAttribute.DN` attribute.

```
public void setDn(EMObjectDn dn)
```

dn is the attribute value.

Throws:

`NullPointerException` if *dn* is null.

`ClassCastException` if *dn* is not an instance of `EMCmipAgentDn`.

Overrides *setDn* in class `EMObject`.

See Also: `DN`

setMpaAddressInfo

Sets the *EMCmipAgentAttribute.MPA_ADDRESS_INFO* attribute.

```
public void setMpaAddressInfo
                  (EMCmipAgentMpaAddressInfo mpaAddressInfo)
```

mpaAddressInfo is the attribute value.

Throws `NullPointerException` if *mpaAddressInfo* is null.

See Also: `MPA_ADDRESS_INFO`

setManagedObjects

Sets the *EMCmipAgentAttribute.MANAGED_OBJECTS* attribute.

```
public void setManagedObjects(String managedObjects[])
```

managedObjects is the attribute value.

Throws `NullPointerException` if *managedObjects* is null.

See Also: `MANAGED_OBJECTS`

addManagedObject

Adds the specified *managedObject* to the `EMCmipAgentAttribute.MANAGED_OBJECTS` attribute.

```
public void addManagedObject(String managedObject)
```

managedObject is the attribute value.

See Also: `MANAGED_OBJECTS`

removeManagedObject

Removes the specified managedObject from the EMCmipAgentAttribute.MANAGED_OBJECTS attribute.

```
public void removeManagedObject(String managedObject)
```

managedObject is the attribute value.

See Also: MANAGED_OBJECTS

setNetworkSAP

Sets the EMCmipAgentAttribute.NETWORK_SAP attribute.

```
public void setNetworkSAP(String networkSAP)
```

networkSAP is the attribute value.

Throws NullPointerException if *networkSAP* is null.

See Also: NETWORK_SAP

setPresentationSelector

Sets the EMCmipAgentAttribute.PRESENTATION_SELECTOR attribute.

```
public void setPresentationSelector(String presentationSelector)
```

presentationSelector is the attribute value.

Throws NullPointerException if *presentationSelector* is null.

See Also: PRESENTATION_SELECTOR

setSessionSelector

Sets the `EMCmipAgentAttribute.SESSION_SELECTOR` attribute.

```
public void setSessionSelector(String sessionSelector)
```

sessionSelector is the attribute value.

Throws `NullPointerException` if *sessionSelector* is null.

See Also: `SESSION_SELECTOR`

setTransportSelector

Sets the `EMCmipAgentAttribute.TRANSPORT_SELECTOR` attribute.

```
public void setTransportSelector(String transportSelector)
```

transportSelector is the attribute value.

Throws `NullPointerException` if *transportSelector* is null.

See Also: `TRANSPORT_SELECTOR`

toString

Returns a String representation of the `EMCmipAgent.`

```
public String toString()
```

Returns the String representation.

Overrides *toString* in class `Object`.

4.12 EMCmipAgentAttribute Class

public class EMCmipAgentAttribute

extends EMTopoAttribute implements Serializable

The com.sun.em.api.topology.EMCmipAgentAttribute class represents a single attribute of the EMCmipAgent persistent object class POC.

4.12.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMObjectAttribute
           |
           +----com.sun.em.api.topology.EMTopoAttribute
                   |
                   +----com.sun.em.api.topology.EMCmipAgentAttribute
```

4.12.2 Variables

MPA_ADDRESS_INFO_ID

```
public static final int MPA_ADDRESS_INFO_ID
```

MPA_ADDRESS_INFO

This variable is the MPA hostname and port number.

```
public static final EMCmipAgentAttribute MPA_ADDRESS_INFO
```

AGENT_ADDRESS_INFO_ID

```
public static final int AGENT_ADDRESS_INFO_ID
```

AGENT_ADDRESS_INFO

This variable is the agent address information in the format defined by AGENT_ADDRESS_TAG.

```
public static final EMCmipAgentAttribute AGENT_ADDRESS_INFO
```

AGENT_ADDRESS_TAG_ID

```
public static final int AGENT_ADDRESS_TAG_ID
```

AGENT_ADDRESS_TAG

This variable defines format of AGENT_ADDRESS_INFO.

```
public static final EMCmipAgentAttribute AGENT_ADDRESS_TAG
```

MANAGED_OBJECTS_ID

```
public static final int MANAGED_OBJECTS_ID
```

MANAGED_OBJECTS

This variable is the list of DNs in slash format of managed objects located on agent. Note that the multiple CMIP agent configurations can be created for the same CMIP MPA but with a different set of managed objects for each configuration.

```
public static final EMCmipAgentAttribute MANAGED_OBJECTS
```

APPLICATION_ENTITY_TITLE_ID

```
public static final int APPLICATION_ENTITY_TITLE_ID
```

APPLICATION_ENTITY_TITLE

This variable is the Application Entity Title (AET).

```
public static final EMCmipAgentAttribute
                                APPLICATION_ENTITY_TITLE
```

PRESENTATION_SELECTOR_ID

```
public static final int PRESENTATION_SELECTOR_ID
```

PRESENTATION_SELECTOR

This variable is the OSI presentation selector.

```
public static final EMCmipAgentAttribute PRESENTATION_SELECTOR
```

SESSION_SELECTOR_ID

```
public static final int SESSION_SELECTOR_ID
```

SESSION_SELECTOR

This variable is the OSI session selector.

```
public static final EMCmipAgentAttribute SESSION_SELECTOR
```

TRANSPORT_SELECTOR_ID

```
public static final int TRANSPORT_SELECTOR_ID
```

TRANSPORT_SELECTOR

This variable is the OSI transport selector.

```
public static final EMCmipAgentAttribute TRANSPORT_SELECTOR
```

NETWORK_SAP_ID

```
public static final int NETWORK_SAP_ID
```

NETWORK_SAP

This variable is the OSI network SAP.

```
public static final EMCmipAgentAttribute NETWORK_SAP
```

4.12.3 Methods

toString

Returns a String representation of the `EMCmipAgentAttribute`.

```
public String toString()
```

Returns the String representation.

Overrides *toString* in class `EMTopoAttribute`.

4.13 EMCmipAgentAttributeSet Class

public class EMCmipAgentAttributeSet

extends EMAttributeSet implements Cloneable, Serializable

The com.sun.em.api.topology.EMCmipAgentAttributeSet class implements an abstract class which forms the basis for the attributes set classes of each EMCmipAgent. EMCmipAgentAttributeSet is used in the Topology API to communicate which attributes of a EMCmipAgent an API method should operate on.

4.13.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMAttributeSet
           |
           +----com.sun.em.api.topology.EMCmipAgentAttributeSet
```

4.13.2 Constructors

EMCmipAgentAttributeSet

Creates an EMCmipAgentAttributeSet which contains no attributes.

```
public EMCmipAgentAttributeSet()
```

4.13.3 Methods

clone

Returns a clone of this `EMCmipAgentAttributeSet` object.

```
public Object clone()
```

Returns a clone of this `EMCmipAgentAttributeSet`.

Overrides *clone* in class `EMAttributeSet.`

elements

Returns an enumeration of the attributes in this set.

```
public Enumeration elements()
```

Returns an enumeration of the attributes in this set.

Overrides *elements* in class `EMAttributeSet`.

4.14 `EMCmipAgentDn` Class

public final class `EMCmipAgentDn`

extends `EMObjectDn` and implements `Serializable`, `Comparable`.

An instance of the `com.sun.em.api.topology.EMCmipAgentDn` class uniquely identifies a CMIP agent.

4.14.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.topology.EMObjectDn
           |
           +----com.sun.em.api.topology.EMCmipAgentDn
```

4.14.2 Constructors

EMCmipAgentDn

Creates an `EMCmipAgentDn` with the specified system name and agent name.

```
public EMCmipAgentDn(String systemName, String uniqueName)
```

systemName is the name of the MIS where the CMIP agent is stored.

uniqueName is the unique name of the CMIP agent within the MIS.

4.14.3 Methods

equals

Compares this object against the specified object. The result is true if, and only if, the argument is not null and is a `EMCmipAgentDn` object which identifies the same CMIP agent as this object.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, otherwise returns false.

Overrides *equals* in class `Object`

compareTo

Compares this `EMCmipAgentDn` with another object that implements `Comparable` interface. Order is determined by first lexicographically comparing the system name, and then the unique name if necessary.

```
public int compareTo(Comparable comparable)
```

Returns: 0 if the objects are identical. Less than 0 if this object is 'less than' the specified object. Greater than 0 if this object is 'greater than' the specified object.

Throws `ClassCastException` if the object parameter is not of class `EMCmipAgentDn`.

getSystemName

Returns the name of the MIS where the CMIP agent is stored.

```
public String getSystemName()
```

Returns the MIS name.

Overrides `getSystemName` in class `EMObjectDn`.

getUniqueName

Returns the name of the CMIP agent. This name is unique within the MIS.

```
public String getUniqueName()
```

Returns the unique name of the CMIP agent.

Overrides *getUniqueName* in class `EMObjectDn`.

hashCode

Returns a hashcode for this object.

```
public int hashCode()
```

Returns the hashcode.

4.15 EMCmipAgentMpaAddressInfo Class

public final class EMCmipAgentMpaAddressInfo

extends Object and implements Cloneable, Serializable.

The com.sun.em.api.topology.EMCmipAgentMpaAddressInfo class is a holder class for the value of EMCmipAgent's MPA_ADDRESS_INFO attribute consisting of MPA hostname and port number. Because this is simply a holder class, the data members are public.

See Also: MPA_ADDRESS_INFO, getMpaAddressInfo, setMpaAddressInfo

4.15.1 Variables

hostname

The hostname where the MPA is located.

```
public String hostname
```

port

The port number on the host used by the MPA.

```
public int port
```

4.15.2 Constructors

EMCmipAgentMpaAddressInfo

Creates an instance of EMCmipAgentMpaAddressInfo with the specified MPA hostname and port number.

```
public EMCmipAgentMpaAddressInfo(String hostname,int port)
```

hostname is the host where the MPA is located.

port is the port number on the host used by the MPA.

4.15.3 Methods

clone

Returns a clone of this EMCmipAgentMpaAddressInfo object.

```
public Object clone()
```

Returns a clone of this EMCmipAgentMpaAddressInfo.

Overrides *clone* in class Object.

equals

Compares this object against the specified object. The result is true if, and only if, the argument is not null and is a EMCmipAgentMpaAddressInfo object that has the same values for hostname and port.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, otherwise returns false.

Overrides *equals* in class Object.

hashCode

Returns a hashcode for this object.

```
public int hashCode()
```

Returns the hashcode.

Overrides `hashCode` in class `Object`.

toString

Returns a String representation of the `EMCmipAgentMpaAddressInfo`.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `Object`.

4.16 EMIndividualNodeListener Interface

public interface `EMIndividualNodeListener`

extends `EventListener`

The `com.sun.em.api.topology.EMIndividualNodeListener` is the listener interface for receiving events of a particular topology node.

See Also: `EMIndividualNodeEvent`, `addEMIndividualNodeListener`, `removeEMIndividualNodeListener`.

4.16.1 Methods

theNodeDeleted

Invoked when the topology node is deleted from the MIS.

```
public void theNodeDeleted(EMTopoNodeEvent event)
```

event is the event information.

theNodeChanged

Invoked when one or more attributes of the topology node were changed in the MIS.

```
public void theNodeChanged(EMTopoNodeEvent event)
```

event is the event information.

4.17 EMInvalidArgException Class

public class `EMInvalidArgException`

extends `EMTopoException`

The `com.sun.em.api.topology.EMInvalidArgException` is the exception returned when an argument (input parameter) is invalid.

4.17.1 Inheritance

```
java.lang.Object
   |
   +----java.lang.Throwable
           |
           +----java.lang.Exception
                   |
                   +----com.sun.em.api.topology.EMTopoException
                           |
                           +----com.sun.em.api.topology.EMInvalidArgException
```

4.17.2 Constructors

EMInvalidArgException

Constructs a new `EMInvalidArgException` with the specified detail message.

```
public EMInvalidArgException(String message)
```

message is the detail message.

EMInvalidArgException

Constructs a new `EMInvalidArgException` with no detail message.

```
public EMInvalidArgException()
```

4.18 `EMObject` Class

public abstract class `EMObject`

extends Object

The `com.sun.em.api.topology.EMObject` class is an abstract base class that specifies the interface supported by all the persistent object classes (POC): `EMTopoNode`, `EMTopoType`, `EMCmipAgent`, `EMRpcAgent`, and `EMSnmpAgent`. Each unit of persistent state is called an attribute and an object is made up of a set of these attributes.

To create a new object in the MIS, first the mandatory attributes required for creation must be set either by loading values from another object or setting the values explicitly using the POC's setter methods. Then, `createWithAllAttributes()` or `createWithSomeAttributes()` is called to create the object in the MIS. Note that `createWithAllAttributes()` only uses attributes that have been given a value. If the create method succeeds, then the `POC.DN` attribute will be set with the unique identifier of the new object. To destroy an object, first the `POC.DN` identifier must be set, and then the `destroy()` method may be called to delete the object from the MIS. This is a permanent, non-reversible operation, so some care should be taken when using this method.

In order to get the attribute values of a particular object, first the `POC.DN` identifier must be set, then either `loadAllAttributes()` or `loadSomeAttributes()` should be called. Once the attribute values are loaded, they stay cached within the POC and remain constant even if the values change in the MIS. In order to set the attribute values persistently in the MIS, first the `POC.DN` attribute must be set, then either `storeAllAttributes()` or `storeSomeAttributes()` may be called. Note that `storeAllAttributes()` only stores those attributes that have been given a value.

4.18.1 Constructors

EMObject

```
public EMObject()
```

4.18.2 Methods

clearAllAttributes

Clears the cached values for all object attributes, resetting them back to their default values (usually null or empty string or 0). This includes the `POC.DN` attribute that associates the POC with a particular object in the MIS. This method is useful when you want to reuse a POC instance to access a different object and do not want the previous values to remain in effect.

```
public abstract void clearAllAttributes()
```

clearSomeAttributes

Clears the cached values for the specified object attributes, resetting them back to their default values (usually null or empty string or 0). This includes the `POC.DN` attribute which associates the POC with a particular object in the MIS. This method is useful when you want to reuse a POC instance to access a different object and do not want the previous values to remain in effect.

```
public abstract void clearSomeAttributes(EMAttributeSet attributes)
```

attributes is the object attributes to have their cached values cleared.

createWithAllAttributes

Creates a new object in the MIS, storing all active attribute values in the object. Any attribute which was not given a value will take on the default value defined by the GDMO for the POC. Note that some attributes may be prohibited from being set at creation time; these attributes are ignored regardless of whether they have a cached value. In order for the create to succeed, the mandatory attributes required by the particular POC must be set. Please refer to the POC class documentation for the list of mandatory attributes required for creation.

```
public abstract void createWithAllAttributes()
                              throws EMTopoServiceException
```

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

createWithSomeAttributes

Creates a new object in the MIS, storing a subset of the active attribute values in the object. Only attributes specified in the parameter attributes will be stored in the new object, and then only if the attribute is active, that is, has been given a value. All other attributes will be given a default value defined by the GDMO for the POC. In order for the create to succeed, the mandatory attributes required by the particular POC must be set. Please refer to the POC class documentation for the list of mandatory attributes required for creation.

```
public abstract void createWithSomeAttributes
                                  (EMAttributeSet attributes)
    throws EMTopoServiceException
```

attributes is subset of the POC's attributes to store in the new object.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

Throws `EMAttributeNotCreatableException` if attributes contains any attributes which cannot be set at creation time.

destroy

Deletes the object identified by `POC.DN` from the MIS. This is a permanent, non-reversible operation, so some care should be taken when using this method.

```
public abstract void destroy()throws EMTopoServiceException,
    EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMUnknownObjectException` if the object `POC.DN` does not exist.

differences

Compares this POC against the specified POC, and returns the set of attributes for which the POCs have different values or null if the POCs are equal. If the argument *obj* is null or not an instance of this POC, then all active attributes of this POC are considered to be differences. Otherwise, an attribute has differing values if the attribute is active for one POC but not the other, or if the attribute is active for both POCs but the values of the attribute are not equal.

```
public abstract EMAttributeSet differences(EMObject obj)
```

obj is the POC to compare against.

Returns the set of attributes for which the POCs have different values or null if the POCs are equal.

differencesSubset

Compares this POC against the specified POC only for those attributes specified by the second parameter, and returns the set of attributes for which the POCs have different values or null if the POCs are equal. If the argument *obj* is null or not an instance of this POC, then all active attributes of this POC are considered to be

differences. Otherwise, an attribute has differing values if the attribute is active for one POC but not for the other, or if the attribute is active for both POCs but the values of the attribute are not equal.

```
public abstract EMAttributeSet
           differencesSubset(EMObject obj,EMAttributeSet attributes)
```

obj is the POC to compare against.

attributes is the set of attributes to compare.

Returns the set of attributes for which the POCs have different values or null if the POCs are equal.

equalsSubset

Compares this POC against the specified object only for those attributes specified by the second parameter, and returns the set of attributes for which the POCs have different values or null if the POCs are equal. If the argument *obj* is null or not an instance of POC, then the two objects are not equal. Otherwise, if the two POCs have the same set of active attributes out of the specified attributes, and the same value for each active attribute, then the POCs are equal.

```
public abstract boolean equalsSubset
                         (EMObject obj,EMAttributeSetattributes)
```

obj is the object to compare against.

attributes is the subset of attributes

Returns true if the objects are equal, otherwise returns false.

exists

Checks to see if the object identified by `POC.DN` exists.

```
public abstract boolean exists() throws EMTopoServiceException
```

Returns true if the object exists.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

getActiveAttributes

Returns the set of object attributes which have been given a value.

```
public abstract EMAttributeSet getActiveAttributes()
```

getDn

Gets the unique identifier from the cache.

```
public abstract EMObjectDn getDn()
```

Returns the unique identifier.

loadAllAttributes

Loads all attributes of the object identified by `POC.DN` from the MIS into the object's attribute cache. These attributes are now considered to be active and can be retrieved with the appropriate getter methods.

```
public abstract void loadAllAttributes() throws
    EMTopoServiceException,EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMUnknownObjectException` if the object POC.DN does not exist.

`EMAttributeDecodeException` if an error occured in converting the attribute values received in ASN.1 format into the local cache data format.

`EMAttributeNotSetException` if the POC.DN attribute is not set.

loadSomeAttributes

Loads the specified attributes of the object identified by POC.DN from the MIS into the object's attribute cache. These attributes are now considered to be active and can be retrieved with the appropriate getter methods.

```
public abstract void loadSomeAttributes
                              (EMAttributeSet attributes)
    throws EMTopoServiceException, EMUnknownObjectException
```

attributes is the object attributes to load from the MIS.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMUnknownObjectException` if the object POC.DN does not exist.

`EMAttributeDecodeException` if an error occured in converting the attribute values received in ASN.1 format into the local cache data format.

`EMAttributeNotSetException` if the POC.DN attribute is not set.

newInstance

Returns a new instance of the POC class without any attributes set.

```
public abstract EMObject newInstance()
```

Returns the new object.

setDn

Sets the unique identifier in the cache.

```
public abstract void setDn(EMObjectDn dn)
```

dn is the dn.

storeAllAttributes

Stores all attributes into the object identified by `POC.DN`. For attributes which have not been given a value by calling load or setter methods, a default value (usually null or empty string or 0) will be stored.

```
public abstract void storeAllAttributes()
    throws EMTopoServiceException, EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMUnknownObjectException` if the object `POC.DN` does not exist.

`EMAttributeEncodeException` if an error occured in converting the attribute values to ASN.1 for transmission to the MIS.

`EMAttributeNotSetException` if the `POC.DN` attribute is not set.

storeSomeAttributes

Stores the specified attributes into the object identified by `POC.DN`. For attributes that have not been given a value by calling load or setter methods, a default value (usually null or empty string or 0) will be stored. Note that some attributes are read-only or can only be set at creation time; these attributes are excluded regardless of whether they have a cached value.

```
public abstract void storeSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException, EMUnknownObjectException
```

attributes is the object attributes to store in the MIS.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMUnknownObjectException` if the object `POC.DN` does not exist.

`EMAttributeNotStoreableException` if attributes contains any attributes which cannot be set at creation time.

`EMAttributeEncodeException` if an error occured in converting the attribute values to ASN.1 for transmission to the MIS.

`EMAttributeNotSetException` if the `POC.DN` attribute is not set.

4.19 `EMObjectDn` Class

public abstract class `EMObjectDn`

extends Object

The `com.sun.em.api.topology.EMObjectDn` class is the abstract base class for unique identifiers of POC objects.

4.19.1 Constructors

`EMObjectDn`

```
public EMObjectDn()
```

4.19.2 Methods

`getSystemName`

Returns the name of the MIS where the POC's managed object is stored.

```
public abstract String getSystemName()
```

Returns the MIS name.

getUniqueName

Returns the name of the POC's managed object. This name is unique within the MIS.

```
public abstract String getUniqueName()
```

4.20 EMPlatformConfigEvent Class

public class EMPlatformConfigEvent

extends EventObject

The com.sun.em.api.topology.EMPlatformConfigEvent class contains information on MIS connection established and removed events.

4.20.1 Inheritance

```
java.lang.Object
   |
   +----java.util.EventObject
           |
           +----com.sun.em.api.topology.EMPlatformConfigEvent
```

4.20.2 Variables

MIS_ADDED

MIS connection established.

```
public static final int MIS_ADDED
```

MIS_REMOVED

MIS connection removed.

```
public static final int MIS_REMOVED
```

4.20.3 Constructors

EMPlatformConfigEvent

Creates an instance of `EMPlatformConfigEvent`.

```
public EMPlatformConfigEvent
                  (Object source, int eventType,String systemName)
```

eventType is the type of event.

systemName is the name of the MIS machine.

4.20.4 Methods

getEventType

Returns event type.

```
public int getEventType()
```

getSystemName

Returns MIS name.

```
public String getSystemName()
```

4.21 EMPlatformConfigListener Interface

public interface `EMPlatformConfigListener`

extends `EventListener`

The `com.sun.em.api.topology.EMPlatformConfigListener` interface is the listener interface for MIS connection established, removed events.

See Also: `EMPlatformConfigEvent`

4.21.1 Methods

misAdded

Invoked when MIS connection is established.

```
public void misAdded(EMPlatformConfigEvent event)
```

misRemoved

Invoked when MIS connection is removed.

```
public void misRemoved(EMPlatformConfigEvent event)
```

4.22 EMRpcAgent Class

public class EMRpcAgent

extends EMAgent implements Cloneable

An instance of the com.sun.em.api.topology.EMRpcAgent class represents the MIS object that contains configuration information for an RPC agent. The configuration information includes the read and write community strings, and supported infos.

Note – This class does not provide an interface to the agent's managed objects, but only to Solstice EM's configuration information for the agent.

4.22.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.topology.EMObject
           |
           +----com.sun.em.api.topology.EMAgent
                   |
                   +----com.sun.em.api.topology.EMRpcAgent
```

4.22.2 Constructors

EMRpcAgent

Constructs an instance of EMRpcAgent representing a particular RPC agent configuration object.

```
public EMRpcAgent(EMTopoPlatform platform, EMRpcAgentDn dn)
```

platform is the Topology API platform.

dn is the unique identifier of the RPC agent configuration object.

4.22.3 Methods

getRpcAgentFromManagedObject

Returns the unique identifier of the RPC agent configuration object whose management domain includes the specified managed object.

```
public static EMRpcAgentDn
    getRpcAgentFromManagedObject(EMTopoPlatform platform,
                                 String managedObjectDnSlashForm)
    throws EMTopoServiceException
```

platform is the Topology API platform.

managedObjectDnSlashForm is the fully distinguished name (fdn) of the managed object

Returns the RPC agent configuration object unique identifier; null if the managed object is not mananaged by a RPC agent.

Throws EMTopoServiceException if there is an internal error in the topology service, or a fault in the communication link to the topology service.

clearAllAttributes

Clears the cached values for all object attributes, resetting them back to their default values (usually null or empty string or 0). This includes the EMRpcAgent.DN attribute that associates the EMRpcAgent instance with a particular RPC agent object in the MIS. This method is useful when you want to reuse the EMRpcAgent instance to access a different RPC agent object and do not want the previous values to remain in effect.

```
public void clearAllAttributes()
```

Overrides clearAllAttributes in class EMObject.

clearSomeAttributes

Clears the cached values for the specified object attributes, resetting them back to their default values (usually null or empty string or 0). This includes the `EMRpcAgent.DN` attribute which associates the `EMRpcAgent` instance with a particular RPC agent object in the MIS. This method is useful when you want to reuse a `EMRpcAgent` instance to access a different proxy agent object and do not want the previous values to remain in effect.

```
public void clearSomeAttributes(EMAttributeSet attributes)
```

attributes is the object attributes to have their cached values cleared.

Overrides `clearSomeAttributes` in class `EMObject`.

clone

Returns a clone of this `EMRpcAgent` object.

```
public Object clone()
```

Returns a clone of this `EMRpcAgent`.

Overrides *clone* in class `Object`.

createWithAllAttributes

Creates a new RPC agent object in the MIS, storing all active attribute values in the object. Any attribute which was not given a value will take on a default value defined by the GDMO (generally null or empty string). In order for the create to succeed, the following attributes must be set:

`EMTopoAttribute.DN`

`EMRpcAgentAttribute.ADMINISTRATIVE_STATE`

`EMRpcAgentAttribute.INFOS`

Note – `EMAgentAttribute.OPERATIONAL_STATE` cannot be set at creation time, and therefore is ignored regardless of whether it has a cached value.

```
public void createWithAllAttributes()
    throws EMTopoServiceException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the any of the mandatory attributes are not set.

Overrides `createWithAllAttributes` in class `EMObject`.

`createWithSomeAttributes`

Creates a new object in the MIS, storing a subset of the active attribute values in the object. Only attributes specified in the parameter *attributes* will be stored in the new object, and then only if the attribute is active, that is,has been given a value. All other attributes will be give a default value defined by the GDMO (generally null or empty string). In order for the create to succeed, the following attributes must be set and must be members of the `EMAttributeSet` attributes:

`EMTopoAttribute.DN`

`EMRpcAgentAttribute.ADMINISTRATIVE_STATE`

`EMRpcAgentAttribute.INFOS`

`EMAgentAttribute.OPERATIONAL_STATE` cannot be set at creation time, and therefore is ignored regardless of whether it has a cached value. If `EMAgentAttribute.OPERATIONAL_STATE` is a member of the `EMAttributeSet` attributes, then an `EMAttributeNotCreatableException` will be thrown.

```
public void createWithSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException
```

attributes is subset of the `EMRpcAgent`'s attributes to store in the new object.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the any of the mandatory attributes are not set.

`EMAttributeNotCreatableException` if attributes contains any attributes which cannot be set at creation time.

Overrides `createWithSomeAttributes` in class `EMObject`.

destroy

Deletes the object identified by `EMRpcAgent.DN` from the MIS. This is a permanent, non-reversible operation, so some care should be taken when using this method.

```
public void destroy()throws EMTopoServiceException,
    EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMRpcAgent.DN` does not exist.

Overrides *destroy* in class `EMObject`.

differences

Compares this `EMRpcAgent` against the specified `EMRpcAgent`, and returns either the set of attributes for which the `EMRpcAgents` have different values, or null if the `EMRpcAgents` are equal. If the argument obj is null or not an instance of `EMRpcAgent`, then all active attributes of this EMRpcAgent are considered to be differences. Otherwise, an attribute has differing values if the attribute is active for one `EMRpcAgent` but not the other, or if the attribute is active for both `EMRpcAgents` but the values of the attribute are not equal.

```
public EMAttributeSet differences(EMObject obj)
```

obj isthe object to compare against

Returns the set of attributes for which the `EMRpcAgents` have different values or null if the `EMRpcAgents` are equal.

Overrides *differences* in class `EMObject`.

differencesSubset

Compares this `EMRpcAgent` against the specified `EMRpcAgent`, but only for those attributes specified by the second parameter, and returns the set of attributes for which the `EMRpcAgents` have different values or null if the `EMRpcAgents` are equal. If the argument *obj* is null or not an instance of `EMRpcAgent`, then all active attributes of this `EMRpcAgent` are considered to be differences. Otherwise, an attribute has differing values if the attribute is active for one `EMRpcAgent` but not the other, or if the attribute is active for both `EMRpcAgents` but the values of the attribute are not equal.

```
public EMAttributeSet
    differencesSubset(EMObject obj,EMAttributeSet attributes)
```

obj is the object to compare against.

attributes is the set of attributes to compare.

Returns the set of attributes for which the `EMRpcAgents` have different values or null if the `EMRpcAgents` are equal.

Overrides `differencesSubset` in class `EMObject`.

equals

Compares this `EMRpcAgent` against the specified object. If the argument obj is null or not an instance of `EMRpcAgent`, then the two objects are not equal. Otherwise, if the two `EMRpcAgents` have the same set of active attributes, and the same value for each active attribute, then the `EMRpcAgents` are equal.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, false otherwise.

Overrides `equals` in class `Object`.

equalsSubset

Compares this EMRpcAgent against the specified object, but only for those attributes specified by the second parameter, restricting the equality check to the the specified attributes. If the argument obj is null or not an instanceof EMRpcAgent, then the two objects are not equal. Otherwise, if the two EMRpcAgents have the same set of active attributes out of the specified attributes, and the same value for each active attribute, then the EMRpcAgents are equal.

```
public boolean equalsSubsett
                  (EMObject obj, EMAttributeSe attributes)
```

obj is the object to compare against.

attributes is the subset of attributes.

Returns true if the objects are equal, false otherwise.

Overrides equalsSubset in class EMObject.

exists

Checks to see if the object identified by EMRpcAgent.DN exists.

```
public boolean exists() throws EMTopoServiceException
```

Returns true if the object exists.

Throws:

EMTopoServiceException if there is an internal error in the topology service, or a fault in the communication link to the topology service.

EMAttributeNotSetException if the EMTopoAttribute.DN attribute is not set.

Overrides exists in class EMObject.

getActiveAttributes

Returns the set of object attributes which have been given a value.

```
public EMAttributeSet getActiveAttributes()
```

Returns the set of active attributes.

Overrides `getActiveAttributes` in class `EMObject`.

loadAllAttributes

Loads all attributes of the object identified by `EMRpcAgent.DN` from the MIS into the object's attribute cache. These attributes are now considered to be active and can be retrieved with the appropriate getter methods.

```
public void loadAllAttributes() throws EMTopoServiceException,
    EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMRpcAgent.DN` does not exist.

`EMAttributeDecodeException` if an error occured in converting the attribute values received in ASN.1 format into the local cache data format.

Overrides `loadAllAttributes` in class `EMObject`

loadSomeAttributes

Loads the specified attributes of the object identified by `EMRpcAgent.DN` from the MIS into the object's attribute cache. These attributes are now considered to be active and can be retrieved with the appropriate getter methods.

```
public void loadSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException, EMUnknownObjectException
```

attributes is the object attributes to load from the MIS.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMRpcAgent.DN` does not exist.

`EMAttributeDecodeException` if an error occured in converting the attribute values received in ASN.1 format into the local cache data format.

Overrides `loadSomeAttributes` in class `EMObject`.

newInstance

Returns a new instance of the `EMRpcAgent` class without any attributes set.

```
public EMObject newInstance()
```

Returns the new object.

Overrides `newInstance` in class `EMObject`.

storeAllAttributes

Stores all attributes into the object identified by `EMRpcAgent.DN`. For attributes that have not been given a value by calling load or setter methods, a default value (usually null or empty string or 0) will be stored. Note that the `EMAgentAttribute.OPERATIONAL_STATE` attribute is read-only and therefore cannot be stored in the MIS. This attribute is ignored even if it has cached values.

```
public void storeAllAttributes()
    throws EMTopoServiceException,EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMRpcAgent.DN` does not exist.

`EMAttributeEncodeException` if an error occured in converting the attribute values to ASN.1 for transmission to the MIS.

Overrides `storeAllAttributes` in class `EMObject`.

storeSomeAttributes

Stores the specified attributes into the object identified by `EMRpcAgent.DN`. For attributes that have not been given a value by calling load or setter methods, a default value (usually null or empty string or 0) will be stored. Note that the `EMAgentAttribute.OPERATIONAL_STATE` attribute is read-only and therefore cannot be stored in the MIS. If this attributes is a member of the `EMAttributeSet` attributes, then an `EMAttributeNotStoreableException` will be thrown.

```
public void storeSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException, EMUnknownObjectException
```

attributes is the object attributes to store in the MIS.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMRpcAgent.DN` does not exist.

`EMAttributeNotStoreableException` if attributes contains any attributes which cannot be set at creation time.

`EMAttributeEncodeException` if an error occured in converting the attribute values to ASN.1 for transmission to MIS.

Overrides `storeSomeAttributes` in class `EMObject`.

getAdministrativeState

Returns the EMAgentAttribute.ADMINISTRATIVE_STATE attribute.

```
public EMAgentAdministrativeState getAdministrativeState()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

Overrides getAdministrativeState in class EMAgent.

See Also: ADMINISTRATIVE_STATE

getDn

Returns the EMTopoAttribute.DN attribute.

```
public EMObjectDn getDn()
```

Returns the attribute value.

Throws: EMAttributeNotSetException if the attribute has not been set in the cache.

Overrides getDn in class EMObject

See Also: DN

getGetCommunityString

Returns the EMRpcAgentAttribute.GET_COMMUNITY_STRING attribute.

```
public String getGetCommunityString()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: GET_COMMUNITY_STRING

getOperationalState

Returns the `EMAgentAttribute.OPERATIONAL_STATE` attribute.

```
public EMAgentOperationalState getOperationalState()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

Overrides `getOperationalState` in class `EMAgent`.

See Also: `OPERATIONAL_STATE`

getInfos

Returns the `EMRpcAgentAttribute.INFOS` attribute.

```
public EMRpcAgentInfo[] getInfos()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `INFOS`

getSetCommunityString

Returns the `EMRpcAgentAttribute.SET_COMMUNITY_STRING` attribute.

```
public String getSetCommunityString()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `SET_COMMUNITY_STRING`

setAdministrativeState

Sets the EMAgentAttribute.ADMINISTRATIVE_STATE attribute.

```
public void setAdministrativeState
                  (EMAgentAdministrativeState administrativeState)
```

administrativeState is the attribute value.

Throws NullPointerException if *administrativeState* is null.

Overrides *setAdministrativeState* in class EMAgent.

See Also: ADMINISTRATIVE_STATE

setDn

Sets the EMTopoAttribute.DN attribute.

```
public void setDn(EMObjectDn dn)
```

dn is the attribute value.

Throws: NullPointerException if *dn* is null.

ClassCastException if *dn* is not an instance of EMRpcAgentDn.

Overrides setDn in class EMObject.

See Also: DN

setGetCommunityString

Sets the EMRpcAgentAttribute.GET_COMMUNITY_STRING attribute.

```
public void setGetCommunityString(String getCommunityString)
```

getCommunityString is the attribute value.

Throws NullPointerException if *getCommunityString* is null.

See Also: GET_COMMUNITY_STRING

setInfos

Sets the `EMRpcAgentAttribute.INFOS` attribute.

```
public void setInfos(EMRpcAgentInfo infos[])
```

infos is the attribute value.

Throws `NullPointerException` if *infos* is null.

See Also: `INFOS`

addInfo

Adds the specified info to the `EMRpcAgentAttribute.INFOS` attribute.

```
public void addInfo(EMRpcAgentInfo info)
```

info is the attribute value.

See Also: `INFOS`

removeInfo

Removes the specified supported MIB from the `EMRpcAgentAttribute.INFOS` attribute.

```
public void removeInfo(EMRpcAgentInfo info)
```

info is the attribute value.

See Also: `INFOS`

setSetCommunityString

Sets the EMRpcAgentAttribute.SET_COMMUNITY_STRING attribute.

```
public void setSetCommunityString(String setCommunityString)
```

setCommunityString is the attribute value.

Throws NullPointerException if *setCommunityString* is null.

See Also: SET_COMMUNITY_STRING

toString

Returns a String representation of the EMRpcAgent.

```
public String toString()
```

Returns the String representation.

Overrides toString in class Object.

4.23 EMRpcAgentAttribute Class

public class EMRpcAgentAttribute

extends EMTopoAttribute implements Serializable

The com.sun.em.api.topology.EMRpcAgentAttribute class represents a single attribute of the EMRpcAgent POC.

4.23.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMObjectAttribute
           |
           +----com.sun.em.api.topology.EMTopoAttribute
                   |
                   +----com.sun.em.api.topology.EMRpcAgentAttribute
```

4.23.2 Variables

GET_COMMUNITY_STRING_ID

```
public static final int GET_COMMUNITY_STRING_ID
```

GET_COMMUNITY_STRING

Community name for GET requests, that is, "public" or "private".

```
public static final EMRpcAgentAttribute GET_COMMUNITY_STRING
```

SET_COMMUNITY_STRING_ID

```
 public static final int SET_COMMUNITY_STRING_ID
```

SET_COMMUNITY_STRING

Community name for SET requests, that is, "public" or "private".

```
public static final EMRpcAgentAttribute SET_COMMUNITY_STRING
```

INFOS_ID

```
public static final int INFOS_ID
```

INFOS

List of RPC proxy hostname and rpc name pairs, that is, "rpchost", "RPC Proxy - ping".

```
public static final EMRpcAgentAttribute INFOS
```

4.23.3 Methods

toString

Returns a String representation of the EMRpcAgentAttribute.

```
public String toString()
```

Returns the String representation.

Overrides toString in class EMTopoAttribute.

4.24 EMRpcAgentAttributeSet Class

public class EMRpcAgentAttributeSet

extends EMAttributeSet implements Cloneable, Serializable

The com.sun.em.api.topology.EMRpcAgentAttributeSet class implements an abstract class which forms the basis for the attributes set classes of each EMRpcAgent. EMRpcAgentAttributeSet is used in the Topology API to communicate which attributes of an EMRpcAgent an API method should operate on.

4.24.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMAttributeSet
           |
           +----com.sun.em.api.topology.EMRpcAgentAttributeSet
```

4.24.2 Constructors

EMRpcAgentAttributeSet

Creates an `EMRpcAgentAttributeSet` which contains no attributes.

```
public EMRpcAgentAttributeSet()
```

clone

Returns a clone of this `EMRpcAgentAttributeSet` object.

```
public Object clone()
```

Overrides `clone` in class `EMAttributeSet`.

elements

Returns an enumeration of the attributes in this set.

```
public Enumeration elements()
```

Returns an enumeration of the attributes in this set.

Overrides `elements` in class `EMAttributeSet`.

4.25 EMRpcAgentDn Class

public final class EMRpcAgentDn

extends `EMObjectDn` and implements `Serializable`, `Comparable`.

An instance of the `com.sun.em.api.topology.EMRpcAgentDn` class uniquely identifies a RPC agent.

4.25.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.topology.EMObjectDn
           |
           +----com.sun.em.api.topology.EMRpcAgentDn
```

4.25.2 Constructors

EMRpcAgentDn

Creates an `EMRpcAgentDn` with the specified system name and agent name.

```
public EMRpcAgentDn(String systemName,String uniqueName)
```

systemName is the name of the MIS where the RPC agent is stored.

uniqueName is the unique name of the RPC agent within the MIS.

4.25.3 Methods

equals

Compares this object against the specified object. The result is true if, and only if, the argument is not null and is a `EMRpcAgentDn` object which identifies the same RPC agent as this object.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, false otherwise.

Overrides *equals* in class `Object`.

compareTo

Compares this `EMRpcAgentDn` with another object which implements `Comparable` interface. Order is determined by first lexicographically comparing the system name and then the unique name, if necessary.

```
public int compareTo(Comparable comparable)
```

Returns 0 if the objects are identical. Less than 0 if this object is 'less than' the specified object. Greater than 0 if this object is 'greater than' the specified object.

Throws `ClassCastException` if the *object* parameter is not of class `EMRpcAgentDn`.

getSystemName

Returns the name of the MIS where the RPC agent is stored.

```
public String getSystemName()
```

Returns the MIS name.

Overrides `getSystemName` in class `EMObjectDn`.

getUniqueName

Returns the name of the RPC agent. This name is unique within the MIS.

```
public String getUniqueName()
```

Returns the unique name of the RPC agent.

Overrides `getUniqueName` in class `EMObjectDn`.

hashCode

Returns a hashcode for this object.

```
public int hashCode()
```

Returns the hashcode.

Overrides `hashCode` in class `Object`.

toString

Returns a String representation of the `EMRpcAgentDn`.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class .

4.26 EMRpcAgentInfo Class

Public class EMRpcAgentInfo

Extends Object implements Cloneable, Serializable, Comparable.

The com.sun.em.api.topology.EMRpcAgentInfo class is a holder class for the EMRpcAgent class.

4.26.1 Variables

name

```
public String name
```

proxyHostname

```
public String proxyHostname
```

4.26.2 Constructors

EMRpcAgentInfo

```
public EMRpcAgentInfo(String name,String proxyHostname)
```

4.26.3 Methods

clone

```
public Object clone()
```

Returns a clone of this `EMRpcAgentInfo` object.

Overrides `clone` in class `Object`.

equals

Compares this object against the specified object. The result is true if, and only if, the argument is not null and is a location object that contains the same values for *name* and *proxyHostname*.

```
public boolean equals(Object obj)
```

obj is the object to compare with.

Returns true if the objects are the same; otherwise returns false.

Overrides `equals` in class `Object`.

compareTo

Compares this `EMRpcAgentInfo` with another object that implements `Comparable` interface. The order is determined first by comparing the name, and then comparing the *proxyHostname*.

```
public int compareTo(Comparable comparable)
```

Returns 0 if the objects are identical. Less than 0 if this object is less than the specified object. Greater than 0 if this object is greater than the specified object.

Throws: `ClassCastException` if the object parameter is not of class `EMRpcAgentInfo`.

toString

Returns a String representation of the `EMRpcAgentInfo`.

```
public String toString()
```

Returns the String representation.

Overrides *toString* in class `Object`.

4.27 EMSnmpAgent Class

public class `EMSnmpAgent`

extends `EMAgent` implements `Cloneable`

An instance of the `com.sun.em.api.topology.EMSnmpAgent` class represents the MIS object that contains configuration information for an SNMP agent. The configuration information includes the `GET` and `SET` community strings, the supported MIBs, and the transport address.

Note – This class does not provide an interface to the agent's managed objects, but only to Solstice EM's configuration information for the agent.

4.27.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.topology.EMObject
           |
           +----com.sun.em.api.topology.EMAgent
                   |
                   +----com.sun.em.api.topology.EMSnmpAgent
```

4.27.2 Constructors

EMSnmpAgent

Constructs an instance of EMSnmpAgent representing a particular SNMP agent configuration object.

```
public EMSnmpAgent(EMTopoPlatform platform, EMSnmpAgentDn dn)
```

platform is the Topology API platform.

dn is the unique identifier of the SNMP agent configuration object.

4.27.3 Methods

clearAllAttributes

Clears the cached values for all object attributes, resetting them back to their default values (usually null or empty string or 0). This includes the EMSnmpAgent.DN attribute that associates the EMSnmpAgent instance with a particular SNMP agent object in the MIS. This method is useful when you want to reuse the EMSnmpAgent instance to access a different SNMP agent object and do not want the previous values to remain in effect.

```
public void clearAllAttributes()
```

Overrides: clearAllAttributes in class EMObject.

clearSomeAttributes

Clears the cached values for the specified object attributes, resetting them back to their default values (usually null or empty string or 0). This includes the EMSnmpAgent.DN attribute that associates the EMSnmpAgent instance with a

particular SNMP agent object in the MIS. This method is useful when you want to reuse a `EMSnmpAgent` instance to access a different proxy agent object and do not want the previous values to remain in effect.

```
public void clearSomeAttributes (EMAttributeSet attributes)
```

attributes is the object attributes to have their cached values cleared.

Overrides `clearSomeAttributes` in class `EMObject`.

clone

Returns a clone of this `EMSnmpAgent` object.

```
public Object clone()
```

Overrides `clone` in class `Object`.

createWithAllAttributes

Creates a new SNMP agent object in the MIS, storing all active attribute values in the object. Any attribute which was not given a value will take on a default value defined by the GDMO (generally null or empty string). In order for the create to succeed, the following attributes must be set:

`EMTopoAttribute.DN`

`EMAgentAttribute.ADMINISTRATIVE_STATE`

`EMSnmpAgentAttribute.SYSTEM_TITLE`

`EMSnmpAgentAttribute.GET_COMMUNITY_STRING`

`EMSnmpAgentAttribute.SET_COMMUNITY_STRING`

`EMSnmpAgentAttribute.TRANSPORT_ADDRESS`

`EMSnmpAgentAttribute.SUPPORTED_MIBS`

`EMSnmpAgentAttribute.MANAGEMENT_PROTOCOL`

`EMSnmpAgentAttribute.ACCESS_CONTROL_ENFORCEMENT`

The following attributes cannot be set at creation time, and therefore are ignored regardless of whether they have a cached value:

EMAgentAttribute.OPERATIONAL_STATE

```
public void createWithAllAttributes()
    throws EMTopoServiceException
```

Throws:

EMTopoServiceException if there is an internal error in the topology service, or a fault in the communication link to the topology service.

EMAttributeNotSetException if any of the mandatory attributes are not set.

Overrides createWithAllAttributes in class EMObject.

createWithSomeAttributes

Creates a new object in the MIS, storing a subset of the active attribute values in the object. Only attributes specified in the parameter attributes will be stored in the new object, and then only if the attribute is active (i.e. has been given a value). All other attributes will be given a default value defined by the GDMO (generally null or empty string). In order for the create to succeed, the following attributes must be set and members of the EMAttributeSet attributes:

EMTopoAttribute.DN

EMAgentAttribute.ADMINISTRATIVE_STATE

EMSnmpAgentAttribute.SYSTEM_TITLE

EMSnmpAgentAttribute.GET_COMMUNITY_STRING

EMSnmpAgentAttribute.SET_COMMUNITY_STRING

EMSnmpAgentAttribute.TRANSPORT_ADDRESS

EMSnmpAgentAttribute.SUPPORTED_MIBS

EMSnmpAgentAttribute.MANAGEMENT_PROTOCOL

EMSnmpAgentAttribute.ACCESS_CONTROL_ENFORCEMENT

The following attributes cannot be set at creation time, and therefore are ignored regardless of whether they have a cached value or not:

EMAgentAttribute.OPERATIONAL_STATE

If any of these attributes are members of the `EMAttributeSet` attributes, then an `EMAttributeNotCreatableException` will be thrown.

```
public void createWithSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException
```

attributes is subset of the `EMSnmpAgent`'s attributes to store in the new object.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the any of the mandatory attributes are not set.

`EMAttributeNotCreatableException` if attributes contains any attributes which cannot be set at creation time.

Overrides `createWithSomeAttributes` in class `EMObject`.

destroy

Deletes the object identified by `EMSnmpAgent.DN` from the MIS. This is a permanent, non-reversible operation, so some care should be taken when using this method.

```
public void destroy() throws EMTopoServiceException,
    EMUnknownObjectExceptionpublic
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMSnmpAgent.DN` does not exist.

Overrides destroy in class `EMObject`.

differences

Compares this `EMSnmpAgent` against the specified `EMSnmpAgent`, and returns the set of attributes for which the `EMSnmpAgents` have different values or null if the `EMSnmpAgents` are equal. If the argument *obj* is null or not an instance of

EMSnmpAgent, then all active attributes of this EMSnmpAgent are considered to be differences. Otherwise, an attribute has differing values if the attribute is active for one EMSnmpAgent but not the other, or if the attribute is active for both EMSnmpAgents but the values of the attribute are not equal.

```
public EMAttributeSet differences(EMObject obj)
```

obj is the object to compare against.

Returns the set of attributes for which the EMSnmpAgents have different values or null if the EMSnmpAgents are equal.

Overrides differences in class EMObject.

differencesSubset

Compares this EMSnmpAgent against the specified EMSnmpAgent, and returns the set of attributes for which the EMSnmpAgents have different values or null if the EMSnmpAgents are equal. If the argument *obj* is null or not an instance of EMSnmpAgent, then all active attributes of this EMSnmpAgent are considered to be differences. Otherwise, an attribute has differing values if the attribute is active for one EMSnmpAgent but not the other, or if the attribute is active for both EMSnmpAgents but the values of the attribute are not equal.

```
public EMAttributeSet
       differencesSubset(EMObject obj,EMAttributeSet attributes)
```

obj is the object to compare against.

attributes is the set of attributes to compare.

Returns the set of attributes for which the EMSnmpAgents have different values or null if the EMSnmpAgents are equal.

Overrides differencesSubset in class EMObject.

equals

Compares this EMSnmpAgent against the specified object. If the argument *obj* is null or not an instance of EMSnmpAgent, then the two objects are not equal. Otherwise, if the two EMSnmpAgents have the same set of active attributes, and the same value for each active attribute, then the EMSnmpAgents are equal.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, false otherwise.

Overrides equals in class Object.

equalsSubset

Compares this EMSnmpAgent against the specified object, restricting the equality check to the specified attributes. If the argument *obj* is null or not an instance of EMSnmpAgent, then the two objects are not equal. Otherwise, if the two EMSnmpAgents have the same set of active attributes out of the specified attributes, and the same value for each active attribute, then the EMSnmpAgents are equal.

```
public Boolean equalsSubset(EMObject obj, EMAttributeSet attributes)
```

obj is the object to compare against.

attributes is the subset of attributes.

Returns true if the objects are equal, false otherwise.

Overrides equalsSubset in class EMObject.

exists

Checks to see if the object identified by EMSnmpAgent.DN exists.

```
public boolean exists() throws EMTopoServiceException
```

Returns true if the object exists.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

Overrides `exists` in class `EMObject`.

getSnmpAgentFromManagedObject

Returns the unique identifier of the SNMP agent configuration object whose management domain includes the specified managed object.

```
public static EMSnmpAgentDn t
    getSnmpAgentFromManagedObjec(EMTopoPlatform platform,
                                 String managedObjectDnSlashForm)
    throws EMTopoServiceException
```

platform is the Topology API platform.

`managedObjectDnSlashForm` is the fully distinguished name (FDN) of the managed object.

Returns the SNMP agent configuration object unique identifier; null if the managed object is not mananaged by an SNMP agent.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

getActiveAttributes

Returns the set of object attributes which have been given a value.

```
public EMAttributeSet getActiveAttributes()
```

Returns the set of active attributes.

Overrides `getActiveAttributes` in class `EMObject`.

loadAllAttributes

Loads all attributes of the object identified by the MIS into the object's attribute cache. These attributes are now considered to be active and can be retrieved with the appropriate getter methods.

```
public void loadAllAttributes()throws EMTopoServiceException,
    EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMSnmpAgent.DN` does not exist.

`EMAttributeDecodeException` if an error occured in converting the attribute values received in ASN.1 format into the local cache data format.

Overrides `loadAllAttributes` in class `EMObject`.

loadSomeAttributes

Loads the specified attributes of the object identified by `EMSnmpAgent.DN` from the MIS into the object's attribute cache. These attributes are now considered to be active and can be retrieved with the appropriate getter methods.

```
public void loadSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException, EMUnknownObjectException
```

`attributes` is the object attributes to load from the MIS.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMSnmpAgent.DN` does not exist.

`EMAttributeDecodeException` if an error occured in converting the attribute values received in ASN.1 format into the local cache data format.

Overrides `loadSomeAttributes` in class `EMObject.`

newInstance

Returns a new instance of the EMSnmpAgent class without any attributes set.

```
public EMObject newInstance()
```

Returns the new object overrides newInstance in class EMObject.

storeAllAttributes

Stores all attributes into the object identified by EMSnmpAgent.DN. For attributes that have not been given a value by calling load or setter methods, a default value (usually null or empty string or 0) will be stored. Note that the following EMSnmpAgent attributes are read-only and therefore cannot be stored in the MIS:

EMAgentAttribute.OPERATIONAL_STATE

EMSnmpAgentAttribute.SYSTEM_TITLE

These attributes are ignored even if they have cached values.

```
public void storeAllAttributes()
    throws EMTopoServiceException, EMUnknownObjectException
```

Throws:

EMTopoServiceException if there is an internal error in the topology service, or a fault in the communication link to the topology service.

EMAttributeNotSetException if the EMTopoAttribute.DN attribute is not set.

EMUnknownObjectException if the object EMSnmpAgent.DN does not exist.

EMAttributeEncodeException if an error occured in converting the attribute values to ASN.1 for transmission to the MIS.

Overrides: storeAllAttributes in class EMObject.

storeSomeAttributes

Stores the specified attributes into the object identified by `EMSnmpAgent.DN`. For attributes that have not been given a value by calling load or setter methods (a default value, usually null or empty string or 0) will be stored. Note that the following `EMSnmpAgent` attributes are read-only and therefore cannot be stored in the MIS:

`EMAgentAttribute.OPERATIONAL_STATE`

`EMSnmpAgentAttribute.SYSTEM_TITLE`

If any of these attributes are members of the `EMAttributeSet` attributes, then an `EMAttributeNotStoreableException` will be thrown.

```
public void storeSomeAttributes(EMAttributeSet attributes) throws
```

attributes is the object attributes to store in the MIS.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMSnmpAgent.DN` does not exist.

`EMAttributeNotStoreableException` if attribute contains any attributes which cannot be set at creation time.

`EMAttributeEncodeException` if an error occured in converting the attribute values to ASN.1 for transmission to the MIS.

Overrides `storeSomeAttributes` in class `EMObject`.

getAdministrativeState

Returns the `EMAgentAttribute.ADMINISTRATIVE_STATE` attribute.

```
public EMAgentAdministrativeState getAdministrativeState()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

Overrides `getAdministrativeState` in class `EMAgent`.

getDn

Returns the `EMTopoAttribute.DN` attribute.

```
public EMObjectDn getDn()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

Overrides `getDn` in class `EMObject`.

See Also: `DN`

getGetCommunityString

Returns the `EMSnmpAgentAttribute.GET_COMMUNITY_STRING` attribute.

```
public String getGetCommunityString()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `GET_COMMUNITY_STRING`

getOperationalState

Returns the `EMAgentAttribute.OPERATIONAL_STATE` attribute.

```
public EMAgentOperationalState getOperationalState()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

Overrides `getOperationalState` in class `EMAgent`.

See Also: `OPERATIONAL_STATE`

getSetCommunityString

Returns the `EMSnmpAgentAttribute.SET_COMMUNITY_STRING` attribute.

```
public String getSetCommunityString()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `SET_COMMUNITY_STRING`

getSupportedMIBs

Returns the `EMSnmpAgentAttribute.SUPPORTED_MIBS` attribute.

```
public String[] getSupportedMIBs()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `SUPPORTED_MIBS`

getSystemTitle

Returns the `EMSnmpAgentAttribute.SYSTEM_TITLE` attribute.

```
public String getSystemTitle()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `SYSTEM_TITLE`

getTransportAddress

Returns the EMSnmpAgentAttribute.TRANSPORT_ADDRESS attribute.

```
public String getTransportAddress()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: TRANSPORT_ADDRESS

getManagementProtocol

Returns the EMSnmpAgentAttribute.MANAGEMENT_PROTOCOL attribute.

```
public EMSnmpAgentManagementProtocol getManagementProtocol()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: MANAGEMENT_PROTOCOL

getAccessControlEnforcement

Returns the EMSnmpAgentAttribute.ACCESS_CONTROL_ENFORCEMENT attribute.

```
public EMSnmpAgentAccessControlEnforcement
                                  getAccessControlEnforcement()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: ACCESS_CONTROL_ENFORCEMENT

getAccessControlMechanism

Returns the EMSnmpAgentAttribute.ACCESS_CONTROL_MECHANISM attribute.

```
public EMSnmpAgentAccessControlMechanism
                                getAccessControlMechanism()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: ACCESS_CONTROL_MECHANISM

setAdministrativeState

Sets the EMAgentAttribute.ADMINISTRATIVE_STATE attribute.

```
public void setAdministrativeState
                  (EMAgentAdministrativeState administrativeState)
```

administrativeState is the attribute value.

Throws NullPointerException if *administrativeState* is null.

Overrides *setAdministrativeState* in class EMAgent.

See Also: ADMINISTRATIVE_STATE

setDn

Sets the EMTopoAttribute.DN attribute.

```
public void setDn(EMObjectDn dn)
```

dn is the attribute value.

Throws NullPointerException if *dn* is null.

Throws ClassCastException if *dn* is not an instance of EMSnmpAgentDn.

Overrides setDn in class EMObject.

See Also: DN

setGetCommunityString

Sets the `EMSnmpAgentAttribute.GET_COMMUNITY_STRING` attribute.

```
public void setGetCommunityString(String getCommunityString)
```

getCommunityString is the attribute value.

Throws `NullPointerException` if *getCommunityString* is null.

See Also: `GET_COMMUNITY_STRING`

setSetCommunityString

Sets the `EMSnmpAgentAttribute.SET_COMMUNITY_STRING` attribute.

```
public void setSetCommunityString(String setCommunityString)
```

setCommunityString is the attribute value.

Throws `NullPointerException` if *setCommunityString* is null.

See Also: `SET_COMMUNITY_STRING`

setSupportedMIBs

Sets the `EMSnmpAgentAttribute.SUPPORTED_MIBS` attribute.

```
public void setSupportedMIBs(String supportedMIBs[])
```

supportedMIBs is the attribute value.

Throws `NullPointerException` if *supportedMIBs* is null.

See Also: `SUPPORTED_MIBS`

addSupportedMIB

Adds the specified supported MIB to the `EMSnmpAgentAttribute.SUPPORTED_MIBS` attribute.

```
public void addSupportedMIB(String supportedMIB)
```

supportedMIB is the attribute value.

See Also: `SUPPORTED_MIBS`

removeSupportedMIB

Removes the specified `supportedMIB` from the `EMSnmpAgentAttribute.SUPPORTED_MIBS` attribute.

```
public void removeSupportedMIB(String supportedMIB)
```

supportedMIB is the attribute value.

See Also: `SUPPORTED_MIBS`

setSystemTitle

Sets the `EMSnmpAgentAttribute.SYSTEM_TITLE` attribute.

```
public void setSystemTitle(String systemTitle)
```

systemTitle is the attribute value.

Throws `NullPointerException` if *systemTitle* is null.

See Also: `SYSTEM_TITLE`

setTransportAddress

Sets the `EMSnmpAgentAttribute.TRANSPORT_ADDRESS` attribute.

```
public void setTransportAddress(String transportAddress)
```

transportAddress is the attribute value.

Throws `NullPointerException` if *transportAddress* is null.

See Also: `TRANSPORT_ADDRESS`

setManagementProtocol

Sets the `EMSnmpAgentAttribute.MANAGEMENT_PROTOCOL` attribute.

```
public void setManagementProtocol
               (EMSnmpAgentManagementProtocol managementProtocol)
```

managementProtocol is the attribute value.

Throws `NullPointerException` if *managementProtocol* is null.

See Also: `MANAGEMENT_PROTOCOL`

setAccessControlEnforcement

Sets the `EMSnmpAgentAttribute.ACCESS_CONTROL_ENFORCEMENT` attribute.

```
public void setAccessControlEnforcement
       (EMSnmpAgentAccessControlEnforcement accessControlEnforcement)
```

accessControlEnforcement is the attribute value.

Throws `NullPointerException` if *managementProtocol* is null.

See Also: `ACCESS_CONTROL_ENFORCEMENT`

setAccessControlMechanism

Sets the EMSnmpAgentAttribute.ACCESS_CONTROL_MECHANISM attribute.

```
public void setAccessControlMechanism
       (EMSnmpAgentAccessControlMechanism accessControlMechanism)
```

accessControlMechanism is the attribute value.

Throws NullPointerException if *managementProtocol* is null.

See Also: ACCESS_CONTROL_MECHANISM

toString

Returns a String representation of the EMSnmpAgent.

```
public String toString()
```

Returns the String representation.

Overrides toString in class Object.

4.28 EMSnmpAgentAccessControlEnforcement Class

public class EMSnmpAgentAccessControlEnforcement

extends Object implements Serializable

The com.sun.em.api.topology.EMSnmpAgentAccessControlEnforcement class represents the Internet management protocol used by a proxy agent to manage devices.

4.28.1 Variables

AGENT

AGENT means that access control is applied at the Internet agent.

```
public static final EMSnmpAgentAccessControlEnforcement AGENT
```

PROXY

PROXY means that access control is applied at the ISO/Internet proxy.

```
public static final EMSnmpAgentAccessControlEnforcement PROXY
```

BOTH

BOTH means that access control is applied at both the Internet agent and the ISO/Internet proxy.

```
public static final EMSnmpAgentAccessControlEnforcement BOTH
```

MIN_ACCESS_CONTROL_ENFORCEMENT

MIN_ACCESS_CONTROL_ENFORCEMENT means the same as AGENT.

```
public static final EMSnmpAgentAccessControlEnforcement
                          MIN_ACCESS_CONTROL_ENFORCEMENT
```

MAX_ACCESS_CONTROL_ENFORCEMENT

MAX_ACCESS_CONTROL_ENFORCEMENT means the same as BOTH.

```
public static final EMSnmpAgentAccessControlEnforcement
                              MAX_ACCESS_CONTROL_ENFORCEMENT
```

4.28.2 Methods

equals

Compares this `EMSnmpAgentAccessControlEnforcement` against the specified object. The result is true if, and only if, the argument is not null and is a `EMSnmpAgentAccessControlEnforcement` object with the same value as this `EMSnmpAgentAccessControlEnforcement`.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, otherwise false.

Overrides `equals` in class `Object`.

toString

Returns a String representation of the `EMSnmpAgentAccessControlEnforcement`.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `Object`.

4.29 EMSnmpAgentAccessControlMechanism Class

public class `EMSnmpAgentAccessControlMechanism`

extends `Object` implements `Serializable`

The `com.sun.em.api.topology.EMSnmpAgentAccessControlMechanism` class represents the Internet management protocol used by a proxy agent to manage devices.

4.29.1 Variables

NO_ACCESS_CONTROL

NO_ACCESS_CONTROL means no access control is used.

```
public static final EMSnmpAgentAccessControlMechanism
                                          NO_ACCESS_CONTROL
```

INTERNET

INTERNET means that Internet access control as specified in SNMPv2SEC is used.

```
public static final EMSnmpAgentAccessControlMechanism INTERNET
```

ISO

ISO means that ISO/CCITT access control as specified in ISO10164-9 is used.

```
public static final EMSnmpAgentAccessControlMechanism ISO
```

MIN_ACCESS_CONTROL_MECHANISM

MIN_ACCESS_CONTROL_MECHANISM means the same as NO_ACCESS_CONTROL.

```
public static final EMSnmpAgentAccessControlMechanism
                                 MIN_ACCESS_CONTROL_MECHANISM
```

MAX_ACCESS_CONTROL_MECHANISM

MAX_ACCESS_CONTROL_MECHANISM means the same as ISO.

```
public static final EMSnmpAgentAccessControlMechanism
                                 MAX_ACCESS_CONTROL_MECHANISM
```

4.29.2 Methods

equals

Compares this `EMSnmpAgentAccessControlMechanism` against the specified object. The result is true if, and only if, the argument is not null and is a `EMSnmpAgentAccessControlMechanism` object with the same value as this `EMSnmpAgentAccessControlMechanism`.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, otherwise returns false.

Overrides `equals` in class `Object`.

toString

Returns a String representation of the `EMSnmpAgentAccessControlMechanism`.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `Object`.

4.30 EMSnmpAgentAttribute Class

public class EMSnmpAgentAttribute

extends EMTopoAttribute implements Serializable

The com.sun.em.api.topology.EMSnmpAgentAttribute class represents a single attribute of the EMSnmpAgent POC.

4.30.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMObjectAttribute
           |
           +----com.sun.em.api.topology.EMTopoAttribute
                   |
                   +----com.sun.em.api.topology.EMSnmpAgentAttribute
```

4.30.2 Variables

SYSTEM_TITLE_ID

```
public static final int SYSTEM_TITLE_ID
```

SYSTEM_TITLE

OID of system title, that is, "1.2.3.4".

```
public static final EMSnmpAgentAttribute SYSTEM_TITLE
```

GET_COMMUNITY_STRING_ID

```
public static final int GET_COMMUNITY_STRING_ID
```

GET_COMMUNITY_STRING

Community name for GET requests, that is, "public" or "private".

```
public static final EMSnmpAgentAttribute GET_COMMUNITY_STRING
```

SET_COMMUNITY_STRING_ID

```
public static final int SET_COMMUNITY_STRING_ID
```

SET_COMMUNITY_STRING

Community name for SET requests, that is, "public" or "private".

```
public static final EMSnmpAgentAttribute SET_COMMUNITY_STRING
```

TRANSPORT_ADDRESS_ID

```
public static final int TRANSPORT_ADDRESS_ID
```

TRANSPORT_ADDRESS

IP address of the system associated with the Internet agent, specified as a string, such as "12.34.56.78". An optional port number may be appended, such as "12.34.56.78:1234".

```
public static final EMSnmpAgentAttribute TRANSPORT_ADDRESS
```

SUPPORTED_MIBS_ID

```
public static final int SUPPORTED_MIBS_ID
```

SUPPORTED_MIBS

The names of the MIBs the SNMP agent supports.

```
public static final EMSnmpAgentAttribute SUPPORTED_MIBS
```

MANAGEMENT_PROTOCOL_ID

```
public static final int MANAGEMENT_PROTOCOL_ID
```

MANAGEMENT_PROTOCOL

The version of SNMP protocol the SNMP agent supports.

```
public static final EMSnmpAgentAttribute MANAGEMENT_PROTOCOL
```

ACCESS_CONTROL_ENFORCEMENT_ID

```
public static final int ACCESS_CONTROL_ENFORCEMENT_ID
```

ACCESS_CONTROL_ENFORCEMENT

The indication of where access control is applied.

```
public static final EMSnmpAgentAttribute
                                    ACCESS_CONTROL_ENFORCEMENT
```

ACCESS_CONTROL_MECHANISM_ID

```
public static final int ACCESS_CONTROL_MECHANISM_ID
```

ACCESS_CONTROL_MECHANISM

The indication of what type of access control is used.

```
public static final EMSnmpAgentAttribute
                                    ACCESS_CONTROL_MECHANISM
```

4.30.3 Methods

toString

Returns a String representation of the `EMSnmpAgentAttribute`.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `EMTopoAttribute`.

4.31 EMSnmpAgentAttributeSet Class

public class EMSnmpAgentAttributeSet

extends EMAttributeSet implements Cloneable, Serializable

The com.sun.em.api.topology.EMSnmpAgentAttributeSet class implements an abstract class which forms the basis for the attributes set classes of each EMSnmpAgent. EMSnmpAgentAttributeSet is used to communicate the attributes of an EMSnmpAgent that an API method should operate on.

4.31.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMAttributeSet
           |
           +----com.sun.em.api.topology.EMSnmpAgentAttributeSet
```

4.31.2 Constructors

EMSnmpAgentAttributeSet

```
public EMSnmpAgentAttributeSet()
```

Creates an EMSnmpAgentAttributeSet which contains no attributes.

4.31.3 Methods

clone

Returns a clone of this `EMSnmpAgentAttributeSet` object.

```
public Object clone()
```

Returns a clone of this `EMSnmpAgentAttributeSet`.

Overrides `clone` in class `EMAttributeSet`.

elements

Returns an enumeration of the attributes in this set.

```
public Enumeration elements()
```

Returns an enumeration of the attributes in this set.

Overrides `elements` in class `EMAttributeSet`.

4.32 `EMSnmpAgentDn` Class

public final class `EMSnmpAgentDn`

extends `EMObjectDn` implements `Serializable`, `Comparable`

An instance of the `com.sun.em.api.topology.EMSnmpAgentDn` class uniquely identifies a SNMP agent.

4.32.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.topology.EMObjectDn
           |
           +----com.sun.em.api.topology.EMSnmpAgentDn
```

4.32.2 Constructors

EMSnmpAgentDn

Creates an EMSnmpAgentDn with the specified system name and agent name.

```
public EMSnmpAgentDn(String systemName, String uniqueName)
```

systemName is the name of the MIS where the SNMP agent is stored.

uniqueName is the unique name of the SNMP agent within the MIS.

4.32.3 Methods

equals

Compares this object against the specified object. The result is true if, and only if, the argument is not null and is a EMSnmpAgentDn object that identifies the same SNMP agent as this object.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, otherwise returns false.

Overrides equals in class Object.

compareTo

Compares this `EMSnmpAgentDn` with another object that implements Comparable interface. Order is determined by first lexicographically comparing the system name, and then the unique name if necessary.

```
public int compareTo(Comparable comparable)
```

Returns 0 if the objects are identical. Less than 0 if this object is 'less than' the specified object. Greater than 0 if this object is 'greater than' the specified object.

Throws `ClassCastException` if the object parameter is not of class `EMSnmpAgentDn`.

getSystemName

Returns the name of the MIS where the SNMP agent is stored.

```
public String getSystemName()
```

Returns the MIS name.

Overrides *getSystemName* in class `EMObjectDn`.

getUniqueName

Returns the name of the SNMP agent. This name is unique within the MIS.

```
public String getUniqueName()
```

Returns the unique name of the SNMP agent.

Overrides *getUniqueName* in class `EMObjectDn`.

hashCode

Returns a hashcode for this object.

```
public int hashCode()
```

Returns the hashcode.

Overrides `hashCode` in class `Object`.

toString

Returns a String representation of the `EMSnmpAgentDn`.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `Object`.

4.33 EMSnmpAgentManagementProtocol Class

public class `EMSnmpAgentManagementProtocol`

extends `Object` implements `Serializable`

The `com.sun.em.api.topology.EMSnmpAgentManagementProtocol` class represents the Internet management protocol used by a proxy agent to manage devices.

4.33.1 Variables

SNMP_V1

The SNMP_V1 protocol means that the agent is using SNMP version 1 as management protocol.

```
public static final EMSnmpAgentManagementProtocol SNMP_V1
```

SNMP_V2

The SNMP_V2 protocol means that the agent is using SNMP version 2 as management protocol. Note that SNMP_V2 is not supported yet.

```
public static final EMSnmpAgentManagementProtocol SNMP_V2
```

4.33.2 Methods

equals

Compares this EMSnmpAgentManagementProtocol against the specified object. The result is true if, and only if, the argument is not null and is an EMSnmpAgentManagementProtocol object with the same value as this EMSnmpAgentManagementProtocol.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, otherwise returns false.

Overrides equals in class Object.

toString

Returns a String representation of the EMSnmpAgentManagementProtocol.

```
public String toString()
```

Returns the String representation.

Overrides toString in class Object.

4.34 EMTopoNode Class

public class EMTopoNode

extends EMObject implements Cloneable, Serializable

The com.sun.em.api.topology.EMTopoNode class represents a topology node, which is the unit of management in Solstice EM. Using the standard POC methods, you can create, delete, and compare topology nodes. Using the EMTopoNode's access methods you can get and set the name, topology pathname, logical and geographical location, topology type, and associated managed objects and their corresponding CMIP, RPC, and/or SNMP agent objects among other attributes. The EMTopoNode class also provides a callback mechanism to notify clients when a topology node has been created, deleted, or had one or more attributes changed.

4.34.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.topology.EMObject
           |
           +----com.sun.em.api.topology.EMTopoNode
```

4.34.2 Constructors

EMTopoNode

Creates an instance of `EMTopoNode` without initializing it to a particular topology node. The version of the constructor is typically used when creating topology nodes, although you can use the `setDn()` method to set the DN attribute after construction time.

```
public EMTopoNode(EMTopoPlatform platform)
```

platform is the Topology API platform.

EMTopoNode

Creates an instance of `EMTopoNode` to represent a particular topology node.

```
public EMTopoNode(EMTopoPlatform platform, EMTopoNodeDn dn)
```

platform is the Topology API platform.

dn is the unique identifier of the topology node.

4.34.3 Methods

addEMTopoNodeListener

Adds the listener to the list of objects to be called when a topology node is created, deleted, or changed.

```
public static void addEMTopoNodeListener(
               EMTopoPlatform platform, EMTopoNodeListener listener)
    throws EMTopoServiceException
```

platform is the Topology API platform.

listener is listener of topology node events.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

See Also: `EMTopoNodeListener`

findNodesByManagedObject

Returns all nodes whose `MANAGED_OBJECTS` attributes include the specified managed object. The nodes from both the local and remote MISs, that is, the list returned by `EMTopoPlatform.getSystemNames()`, are included in the search set.

```
public static EMTopoNodeDn[] t
    findNodesByManagedObjec (EMTopoPlatform platform,String
                                managedObjectDnSlashForm)
    throws EMTopoServiceException, EMInvalidArgException
```

platform is the Topology API platform.

`managedObjectDnSlashForm` is the managed object fully distinguished name (FDN) in slash form, that is, "/systemId=\"ultra\"/networkId=23/bstId=43".

Returns nodes that have the specified managed object.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

Throws `EMInvalidArgException` if `managedObjectDnSlashForm` is an invalid FDN.

findNodesByManagedObject

Returns all nodes whose `MANAGED_OBJECTS` attributes includes the specified managed object. Only the nodes contained in the specified MIS systems are included in the search set.

```
public static EMTopoNodeDn[]
    findNodesByManagedObject(EMTopoPlatform platform, String
                    managedObjectDnSlashForm, String systemNames[])
    throws EMInvalidArgException, EMTopoServiceException
```

platform is the Topology API platform.

`managedObjectDnSlashForm` is the managed object *dn* in slash form, that is, "/systemId=\"ultra\"/networkId=23/bstId=43"

systemNames is the MIS systems to restrict the query to.

Returns nodes that have the specified managed object.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMInvalidArgException` if one of the specified system names is not a valid system name of an MIS that is visible through the current `Platform` connection.

findNodesByName

Returns all nodes whose `NAME` attribute equals the specified name. The nodes from both the local and remote MISs, that is,the list returned by `EMTopoPlatform.getSystemNames()`, are included in the search set.

```
public static EMTopoNodeDn[]
    findNodesByName(EMTopoPlatform platform,String name)
    throws EMTopoServiceException
```

platform is the Topology API platform.

name is the name to match.

Returns nodes with the given administrative name.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

findNodesByName

Returns all nodes whose NAME attribute equals the specified name. Only the nodes contained in the specified MIS systems are included in the search set.

```
public static EMTopoNodeDn[]
    findNodesByNamem(EMTopoPlatfor platform, String name,
                                          String systemNames[])
    throws EMTopoServiceException, EMInvalidArgException
```

platform is the Topology API platform.

name is the name to match.

systemNames is the MIS systems to restrict the query to.

Returns nodes with the given administrative name.

Throws:

EMTopoServiceException if there is an internal error in the topology service, or a fault in the communication link to the topology service.

EMInvalidArgException if one of the specified system names is not a valid system name of an MIS that is visible through the current Platform connection.

findNodesByType

Returns all nodes whose TYPE_NAME attribute equals the specified type name. The nodes from both the local and remote MISs, that is, the list returned by EMTopoPlatform.getSystemNames(), are included in the search set.

```
public static EMTopoNodeDn[]
        findNodesByType(EMTopoPlatform platform, String typeName)
    throws EMTopoServiceException
```

platform is the Topology API platform.

typeName is the type name to match.

Returns nodes of the specified type.

Throws EMTopoServiceException if there is an internal error in the topology service, or a fault in the communication link to the topology service.

findNodesByType

Returns all nodes whose TYPE_NAME attribute equals the specified type name. Only the nodes contained in the specified MIS systems are included in the search set.

```
public static EMTopoNodeDn[]
    findNodesByType(EMTopoPlatform platform, String typeName,
                                             String systemNames[])
    throws EMTopoServiceException, EMInvalidArgException
```

platform is the Topology API platform.

typeName is the type name to match.

systemNames is the MIS systems to restrict the query to.

Returns nodes of the specified type.

Throws:

EMTopoServiceException if there is an internal error in the topology service, or a fault in the communication link to the topology service.

EMInvalidArgException if one of the specified system names is not a valid system name of an MIS that is visible through the current Platform connection.

findMOsByNodes

Returns the names of all the managed objects corresponding to the specified topology nodes in the specified MIS system. In other words, the returned managed object names are contained in the specified topology nodes' EMTopoNodeAttribute.MANAGED_OBJECTS attribute.

```
public static MOName[]
    findMOsByNodes(EMTopoPlatform platform,String systemName,
                                   int nodeIDs[], boolean needDNs)
    throws EMTopoServiceExceptionloadNodesInView
```

platform is the Topology API platform.

systemName is the MIS system to restrict the query to.

nodeIDs is the group of topology nodes this query is performed on; topology node ID is the integer part of topology node DN.

needDNs indicates whether the return `MONames` should contain managed object names (FDNs) in addition to the internal managed object IDs. See `MOName` class for details.

Returns the managed object names correponding to the specified topology nodes in the specified MIS system.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

loadNodesInView

Returns all the nodes in the specified view. The returned nodes would only have the specified attributes filled in.

```
public static EMTopoNode[]
    loadNodesInView(EMTopoPlatform platform, EMTopoNodeDn viewDn,
                                EMTopoNodeAttributeSet attributes)
    throws EMTopoServiceException
```

platform is the Topology API platform.

viewDn is the parent view.

attributes is the set specifying the attributes that will be returned.

Returns the nodes of the specified view with the specified attributes.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

findRootNodes

Returns all nodes named "Root". The nodes from both the local and remote MISs, (the list returned by `EMTopoPlatform.getSystemNames()`) are included in the search set.

```
public static EMTopoNodeDn[]
    findRootNodes(EMTopoPlatform platform)
    throws EMTopoServiceException
```

platform is the Topology API platform.

Returns the root nodes.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

findRootNodes

Returns all nodes named "Root". Only the nodes contained in the specified MIS systems are included in the search set.

```
public static EMTopoNodeDn[]
    findRootNodes (EMTopoPlatform platform, String systemNames[])
    throws EMTopoServiceException, EMInvalidArgException
```

platform is the Topology API platform.

systemNames is the MIS systems to restrict the query to.

Returns the root nodes.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMInvalidArgException` if one of the specified system names is not a valid system name of an MIS that is visible through the current `Platform` connection.

getTopologyPathnames

Constructs all possible topology pathnames for the specified topology node name contained in the specified view. At a minimum, there will be one pathname for each parent view of viewDn. However, the actual number of pathnames may be higher since each parent can also have more than one parent continuing until the root of the topology graph "Root" is reached.

```
public static String[]
    getTopologyPathnames(EMTopoPlatform platform,
                         EMTopoNodeDn viewDn,String name)
    throws EMTopoServiceException, EMInvalidArgException
```

platform is the Topology API platform.

viewDn is the unique identifier of a topology node which is a view.

name is a topology node NAME. Note that the name does not have to the name of a real topology node contained within viewDn.

Returns the topology pathnames of the view with the topology node name appended.

Throws:

EMTopoServiceException if there is an internal error in the topology service, or a fault in the communication link to the topology service.

EMInvalidArgException if viewDn is not a view.

getTopologyPathnamesOfView

Constructs all possible topology pathnames for the specified topology view. At a minimum, there will be one pathname for each parent view of viewDn . However, the actual number of pathnames may be higher since each parent can also have more than one parent continuing until the root of the topology graph "Root" is reached.

```
public static String[]
    getTopologyPathnamesOfView(EMTopoPlatform platform,
                               EMTopoNodeDn viewDn)
    throws EMTopoServiceException, EMInvalidArgException
```

platform is the Topology API platform.

viewDn is the unique identifier of a topology node which is a view.

Returns the topology pathnames of the view.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMInvalidArgException` if *viewDn* is not a view.

isView

Returns true if the node's `TYPE_NAME` is a view type, that is, if `EMTopoType.isView(TYPE_NAME)` equals true. This method is more efficient than the client loading the TYPE_NAME and calling `EMTopoType.isView(TYPE_NAME)`.

```
public static boolean isView
                      (EMTopoPlatform platform,EMTopoNodeDn dn)
    throws EMTopoServiceException
```

platform is the Topology API platform.

dn is the unique identifier of the topology node.

Returns true if the node's *TYPE_NAME* is a view type.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

loadNodes

Loads the specified attributes for each of the specified topology nodes and returns the result. When loading a large block of topology nodes, using `loadNodes()` is faster than creating an `EMTopoNode` and calling `EMTopoNode.loadSomeAttributes()` for each one.

```
public static EMTopoNode[]
    loadNodes(EMTopoPlatform platform,EMTopoNodeAttributeSet
                              attributes, EMTopoNodeDn nodeDns[])

    throws EMTopoServiceException, EMUnknownObjectException
```

platform is the Topology API platform.

attributes is the attributes to load for each topology node.

nodeDns is the topology nodes to load.

Returns the cached attributes for the specified topology nodes.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMUnknownObjectException` if the object `EMTopoAttribute.DN` does not exist.

`EMAttributeDecodeException` if an error occurred in converting the attribute values received in ASN.1 format into the local cache data format.

loadNodesInBatches

Asynchronously loads specified attributes for each of the specified topology nodes, calling the specified listener's `batchReceived()` method after each batch of topology nodes have been loaded.

```
public static void loadNodesInBatches
    (EMTopoPlatform platform,EMAttributeSet attributes,EMTopoNodeDn
    nodeDns[], int batchSize,EMTopoNodeBatchLoaderListener listener)
```

platform is the Topology API platform.

attributes is the attributes to load for each topology node.

nodeDns is the topology nodes to load.

batchSize is the number of nodes to load in each batch.

listener is the listener to notify after each batch is loaded.

See Also: `EMTopoNodeBatchLoaderListener`

removeEMTopoNodeListener

Removes the specified listener from the list of objects to be called when a topology node is created, deleted, or changed.

```
public static void removeEMTopoNodeListener
               (EMTopoPlatform platform,EMTopoNodeListener listener)
throws EMTopoServiceException
```

platform is the Topology API platform.

listener is the listener of topology events.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

See Also: `EMTopoNodeListener`

clearAllAttributes

Clears the cached values for all object attributes, resetting them back to their default values (usually null or empty string or 0). This includes the `EMTopoAttribute.DN` attribute which associates the `EMTopoNode` instance with a particular topology node object in the MIS. This method is useful when you want to reuse the `EMTopoNode` instance to access a different topology node object and do not want the previous values to remain in effect.

```
public void clearAllAttributes()
```

Overrides `clearAllAttributes` in class `EMObject`.

clearSomeAttributes

Clears the cached values for the specified object attributes, resetting them back to their default values (usually null or empty string or 0). This includes the `EMTopoAttribute.DN` attribute which associates the *EMTopoNode* instance with a particular topology node object in the MIS.

This method is useful when you want to reuse a `EMTopoNode` instance to access a different proxy agent object and do not want the previous values to remain in effect.

```
public void clearSomeAttributes(EMAttributeSet attributes)
```

attributes is the object attributes to have their cached values cleared.

Overrides `clearSomeAttributes` in class `EMObject`.

clone

Returns a clone of this `EMTopoNode` object.

```
public Object clone()
```

Returns a clone of this `EMTopoNode`.

Overrides `clone` in class `Object`.

createWithAllAttributes

Creates a new topology node object in the MIS, storing all active attribute values in the object. Any attribute which was not given a value will take on a default value defined by the GDMO (generally null or empty string). In order for the create to succeed, the following attributes must be set:

`EMTopoNodeAttribute.NAME`

`EMTopoNodeAttribute.TYPE_NAME`

`EMTopoNodeAttribute.PARENTS`

The following attributes cannot be set at creation time, and therefore are ignored regardless of whether they have a cached value:

`EMTopoAttribute.DN`

`EMTopoNodeAttribute.TOPOLOGY_PATHNAMES`

`EMTopoNodeAttribute.CHILDREN`

`EMTopoNodeAttribute.VIEW_CHILDREN`

`EMTopoNodeAttribute.PROPAGATED_SEVERITY`

`EMTopoNodeAttribute.MONITOR_VISIBLE_CHILDREN`

`EMTopoNodeAttribute.MONITOR_HIDDEN_CHILDREN`

`EMTopoNodeAttribute.MONITOR_MAX_VISIBLE_CHILDREN`

```
public void createWithAllAttributes()
    throws EMTopoServiceException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if any of the mandatory attributes are not set: `EMTopoNodeAttribute.NAME`, `EMTopoNodeAttribute.TYPE_NAME`, and `EMTopoNodeAttribute.PARENTS`.

Overrides `createWithAllAttributes` in class `EMObject`.

`createWithSomeAttributes`

Creates a new object in the MIS, storing a subset of the active attribute values in the object. Only attributes specified in the parameter attributes will be stored in the new object, and then only if the attribute is active, that is, has been given a value. All other attributes will be give a default value defined by the GDMO, generally null or empty string. In order for the create to succeed, the following attributes must be set and members of the `EMAttributeSet` attributes:

`EMTopoNodeAttribute.NAME`

`EMTopoNodeAttribute.TYPE_NAME`

`EMTopoNodeAttribute.PARENTS`

The following attributes cannot be set at creation time, and therefore are ignored regardless of whether they have a cached value:

`EMTopoAttribute.DN`

`EMTopoNodeAttribute.TOPOLOGY_PATHNAMES`

`EMTopoNodeAttribute.CHILDREN`

`EMTopoNodeAttribute.VIEW_CHILDREN`

`EMTopoNodeAttribute.PROPAGATED_SEVERITY`

`EMTopoNodeAttribute.MONITOR_VISIBLE_CHILDREN`

`EMTopoNodeAttribute.MONITOR_HIDDEN_CHILDREN`

`EMTopoNodeAttribute.MONITOR_MAX_VISIBLE_CHILDREN`

If any of these attributes are members of the `EMAttributeSet` attributes, then an `EMAttributeNotCreatableException` will be thrown.

```
public void createWithSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException
```

attributes is subset of the `EMTopoNode`'s attributes to store in the new object.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if any of the mandatory attributes are not set: `EMTopoNodeAttribute.NAME`, `EMTopoNodeAttribute.TYPE_NAME`, and `EMTopoNodeAttribute.PARENTS`.

`EMAttributeNotCreatableException` if attributes contains any attributes which cannot be set at creation time.

Overrides `createWithSomeAttributes` in class `EMObject`.

destroy

Deletes the object identified by `EMTopoAttribute.DN` from the MIS. This is a permanent, non-reversible operation, so some care should be taken when using this method.

```
public void destroy()throws EMTopoServiceException,
    EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMTopoAttribute.DN` does not exist.

Overrides `destroy` in class `EMObject`

differences

Compares this `EMTopoNode` against the specified `EMTopoNode`, and returns the set of attributes for which the `EMTopoNode`s have different values or null if the `EMTopoNode`s are equal. If the argument *obj* is null or not an instance of `EMTopoNode`, then all active attributes of this `EMTopoNode` are considered to be differences. Otherwise, an attribute has differing values if the attribute is active for one `EMTopoNode` but not the other, or if the attribute is active for both `EMTopoNode`s but the values of the attribute are not equal.

```
public EMAttributeSet differences(EMObject obj)
```

obj is the object to compare against.

Returns the set of attributes for which the `EMTopoNode`s have different values or null if the `EMTopoNode`s are equal.

Overrides `differences` in class `EMObject`.

differencesSubset

Compares this EMTopoNode against the specified EMTopoNode, and returns the set of attributes for which the EMTopoNodes have different values or null if the EMTopoNodes are equal. If the argument *obj* is null or not an instance of EMTopoNode, then all active attributes of this EMTopoNode are considered to be differences. Otherwise, an attribute has differing values if the attribute is active for one EMTopoNode but not the other, or if the attribute is active for both EMTopoNodes but the values of the attribute are not equal.

```
public EMAttributeSet
    differencesSubset(EMObject obj,EMAttributeSet attributes)
```

obj is the object to compare against.

attributes is the set of attributes to compare.

Returns the set of attributes for which the EMTopoNodes have different values or null if the EMTopoNodes are equal.

Overrides differencesSubset in class EMObject.

equals

Compares this EMTopoNode against the specified object. If the argument *obj* is null or not an instance of EMTopoNode, then the two objects are not equal. Otherwise, if the two EMTopoNodes have the same set of active attributes and the same value for each active attribute, then the EMTopoNodes are equal.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, otherwise returns false.

Overrides equals in class Object.

equalsSubset

Compares this EMTopoNode against the specified object, restricting the equality check to the specified attributes. If the argument *obj* is null or not an instance of EMTopoNode, then the two objects are not equal. Otherwise, if the two EMTopoNodes have the same set of active attributes out of the specified attributes, and the same value for each active attribute, then the EMTopoNodes are equal.

```
public boolean equalsSubset(EMObject obj,
                            EMAttributeSet attributes)
```

obj is the object to compare against.

attributes is the subset of attributes.

Returns true if the objects are equal, otherwise returns false.

Overrides *equalsSubset* in class EMObject.

exists

Checks to see if the object identified by EMTopoAttribute.DN exists.

```
public boolean exists() throws EMTopoServiceException
```

Returns true if the object exists.

Throws:

EMTopoServiceException if there is an internal error in the topology service, or a fault in the communication link to the topology service.

EMAttributeNotSetException if the EMTopoAttribute.DN attribute is not set.

Overrides exists in class EMObject.

getActiveAttributes

Returns the set of object attributes which have been given a value.

```
public EMAttributeSet getActiveAttributes()
```

Returns the set of active attributes.

Overrides `getActiveAttributes` in class `EMObject`.

loadAllAttributes

Loads all attributes of the object identified by `EMTopoAttribute.DN` from the MIS into the object's attribute cache. These attributes are now considered to be active and can be retrieved with the appropriate getter methods.

```
public void loadAllAttributes()throws EMTopoServiceException,
    EMUnknownObjectException
```

Throws:

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMUnknownObjectException` if the object `EMTopoAttribute.DN` does not exist.

`EMAttributeDecodeException` if an error occured in converting the attribute values received in ASN.1 format into the local cache data format.

Overrides `loadAllAttributes` in class `EMObject`.

loadSomeAttributes

Loads the specified attributes of the object identified by `EMTopoAttribute.DN` from the MIS into the object's attribute cache. These attributes are now considered to be active and can be retrieved with the appropriate getter methods.

```
public void loadSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException, EMUnknownObjectException
```

attributes is the object attributes to load from the MIS.

Throws:

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMUnknownObjectException` if the object `EMTopoAttribute.DN` does not exist.

`EMAttributeDecodeException` if an error occured in converting the attribute values received in ASN.1 format into the local cache data format.

Overrides `loadSomeAttributes` in class `EMObject`.

newInstance

Returns a new instance of the `EMTopoNode` class without any attributes set.

```
public EMObject newInstance()
```

Returns the new object.

Overrides `newInstance` in class `EMObject`.

storeAllAttributes

Stores all attributes into the object identified by `EMTopoAttribute.DN`. For attributes that have not been given a value by calling load or setter methods, a default value (usually null or empty string or 0) will be stored. Note that the following `EMTopoNode` attributes are read-only and therefore cannot be stored in the MIS:

`EMTopoNodeAttribute.TOPOLOGY_PATHNAMES`

`EMTopoNodeAttribute.CHILDREN`

`EMTopoNodeAttribute.VIEW_CHILDREN`

`EMTopoNodeAttribute.PROPAGATED_SEVERITY`

`EMTopoNodeAttribute.MONITOR_HIDDEN_CHILDREN`

`EMTopoNodeAttribute.MONITOR_MAX_VISIBLE_CHILDREN`

These attributes are ignored even if they have cached values.

```
public void storeAllAttributes()throws EMTopoServiceException,
    EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMUnknownObjectException` if the object `EMTopoAttribute.DN` does not exist.

`EMAttributeEncodeException` if an error occured in converting the attribute values to ASN.1 for transmission to MIS.

`EMAttributeNotSetException` if the EMTopoAttribute.DN attribute is not set.

Overrides `storeAllAttributes` in class `EMObject`

storeSomeAttributes

Stores the specified attributes into the object identified by `EMTopoAttribute.DN`. For attributes that have not been given a value by calling load or setter methods, a default value (usually null or empty string or 0) will be stored. Note that the following `EMTopoNode` attributes are read-only and therefore cannot be stored in the MIS:

`EMTopoNodeAttribute.TOPOLOGY_PATHNAMES`

`EMTopoNodeAttribute.CHILDREN`

`EMTopoNodeAttribute.VIEW_CHILDREN`

`EMTopoNodeAttribute.PROPAGATED_SEVERITY`

`EMTopoNodeAttribute.MONITOR_HIDDEN_CHILDREN`

`EMTopoNodeAttribute.MONITOR_MAX_VISIBLE_CHILDREN`

If any of these attributes are members of the *EMAttributeSet* attributes, then an `EMAttributeNotStoreableException` will be thrown.

```
public void storeSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException, EMUnknownObjectException
```

attributes is the object attributes to store in the MIS.

Throws:

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMAttributeNotStoreableException` if attributes contains any attributes which cannot be set at creation time.

`EMUnknownObjectException` if the object `EMTopoAttribute.DN` does not exist.

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeEncodeException` if an error occured in converting the attribute values to ASN.1 for transmission to MIS.

Overrides `storeSomeAttributes` in class `EMObject`.

getChildren

Returns the `EMTopoNodeAttribute.CHILDREN` attribute.

```
public EMTopoNodeDn[] getChildren()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `CHILDREN`

getCmipAgents

Returns the `EMTopoNodeAttribute.CMIP_AGENTS` attribute.

```
public EMCmipAgentDn[] getCmipAgents()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `CMIP_AGENTS`

getDisplayStatuses

Returns the `EMTopoNodeAttribute.DISPLAY_STATUSES` attribute.

```
public EMTopoNodeDisplayStatus[] getDisplayStatuses()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `DISPLAY_STATUSES`

getDn

Returns the `EMTopoNodeAttribute.DN` attribute.

```
public EMObjectDn getDn()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

Overrides `getDn` in class `EMObject`.

See Also: `DN`

getGeographicalLocation

Returns the `EMTopoNodeAttribute.GEOGRAPHICAL_LOCATION` attribute.

```
public EMTopoNodeGeoLocation getGeographicalLocation()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `GEOGRAPHICAL_LOCATION`

getLayerName

Returns the `EMTopoNodeAttribute.LAYER_NAME` attribute.

```
public String getLayerName()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `LAYER_NAME`

getLinks

Returns the `EMTopoNodeAttribute.LINKS` attribute.

```
public EMTopoNodeDn[] getLinks()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `LINKS`

getLogicalLocation

Returns the logical location of the topology node in the specified parent view.

```
public EMTopoNodeLocation
                      getLogicalLocation(EMTopoNodeDn parentDn)
    throws EMInvalidArgException
```

Returns the logical lcoation.

Throws:

`EMAttributeNotSetException` if the attribute has not been set in the cache.

`EMInvalidArgException` if there is no location for the specified parent view in the `EMTopoNodeAttribute.LOGICAL_LOCATIONS` attribute.

See Also: `LOGICAL_LOCATIONS`

getLogicalLocations

Returns the `EMTopoNodeAttribute.LOGICAL_LOCATIONS` attribute.

```
public EMTopoNodeLocationInParent[] getLogicalLocations()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `LOGICAL_LOCATIONS`

getManagedObjects

Returns the `EMTopoNodeAttribute.MANAGED_OBJECTS` attribute.

```
public String[] getManagedObjects()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `MANAGED_OBJECTS`

getMonitorHiddenChildren

Returns the `EMTopoNodeAttribute.MONITOR_HIDDEN_CHILDREN` attribute.

```
public EMTopoNodeDn[] getMonitorHiddenChildren()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `MONITOR_HIDDEN_CHILDREN`

getMonitorMaxVisibleChildren

Returns the EMTopoNodeAttribute.MONITOR_MAX_VISIBLE_CHILDREN attribute.

```
public Integer getMonitorMaxVisibleChildren()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: MONITOR_MAX_VISIBLE_CHILDREN

getMonitorRotation

Returns the EMTopoNodeAttribute.MONITOR_ROTATION attribute.

```
public Integer getMonitorRotation()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: MONITOR_ROTATION

getMonitorVisibleChildren

Returns the EMTopoNodeAttribute.MONITOR_VISIBLE_CHILDREN attribute.

```
public EMTopoNodeDn[] getMonitorVisibleChildren()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: MONITOR_VISIBLE_CHILDREN

getArrayVisibleChildren

Returns the `EMTopoNodeAttribute.ARRAY_VISIBLE_CHILDREN` attribute.

```
public EMTopoNodeDn[] getArrayVisibleChildren()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `ARRAY_VISIBLE_CHILDREN`

getArrayHiddenChildren

Returns the `EMTopoNodeAttribute.ARRAY_HIDDEN_CHILDREN` attribute.

```
public EMTopoNodeDn[] getArrayHiddenChildren()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `ARRAY_HIDDEN_CHILDREN`

getArrayCellSize

Returns the `EMTopoNodeAttribute.ARRAY_CELL_SIZE` attribute.

```
public EMTopoNodeArrayCellSize getArrayCellSize()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `ARRAY_CELL_SIZE`

getArrayOrientation

Returns the EMTopoNodeAttribute.ARRAY_ORIENTATION attribute.

```
public EMTopoNodeArrayOrientation getArrayOrientation()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: ARRAY_ORIENTATION

getArrayNumColumns

Returns the EMTopoNodeAttribute.ARRAY_NUM_COLUMNS attribute.

```
public Integer getArrayNumColumns()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: ARRAY_NUM_COLUMNS

getBusLogicalLocations

Returns the EMTopoNodeAttribute.BUS_LOGICAL_LOCATIONS attribute.

```
public EMTopoNodeLocation[] getBusLogicalLocations()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: BUS_LOGICAL_LOCATIONS

getName

Returns the `EMTopoNodeAttribute.NAME` attribute.

```
public String getName()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `NAME`

getParents

Returns the `EMTopoNodeAttribute.PARENTS` attribute.

```
public EMTopoNodeDn[] getParents()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `PARENTS`

getPropagatePeers

Returns the `EMTopoNodeAttribute.PROPAGATE_PEERS` attribute.

```
public EMTopoNodeDn[] getPropagatePeers()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `PROPAGATE_PEERS`

getPropagatedSeverity

Returns the `EMTopoNodeAttribute.PROPAGATED_SEVERITY` attribute.

```
public EMSeverity getPropagatedSeverity()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `PROPAGATED_SEVERITY`

getRpcAgents

Returns the `EMTopoNodeAttribute.RPC_AGENTS` attribute.

```
public EMRpcAgentDn[] getRpcAgents()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `RPC_AGENTS`

getSeverity

Returns the `EMTopoNodeAttribute.SEVERITY` attribute.

```
public EMSeverity getSeverity()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `SEVERITY`

getSnmpAgents

Returns the `EMTopoNodeAttribute.SNMP_AGENTS` attribute.

```
public EMSnmpAgentDn[] getSnmpAgents()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `SNMP_AGENTS`

getState

Returns the `EMTopoNodeAttribute.STATE` attribute.

```
public Integer getState()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `STATE`

getTopologyPathnames

Returns the `EMTopoNodeAttribute.TOPOLOGY_PATHNAMES` attribute.

```
public String[] getTopologyPathnames()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `TOPOLOGY_PATHNAMES`

getTypeName

Returns the EMTopoNodeAttribute.TYPE_NAME attribute.

```
public String getTypeName()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: TYPE_NAME

getUserData

Returns the EMTopoNodeAttribute.USER_DATA attribute.

```
public EMTopoNodeUserDatum[] getUserData()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: USER_DATA

getViewBackgroundImageFilename

Returns the EMTopoNodeAttribute.VIEW_BACKGROUND_IMAGE_FILENAME attribute.

```
public String getViewBackgroundImageFilename()
```

Returns the attribute value.

Throws EMAttributeNotSetException if the attribute has not been set in the cache.

See Also: VIEW_BACKGROUND_IMAGE_FILENAME

getViewChildren

Returns the `EMTopoNodeAttribute.VIEW_CHILDREN` attribute.

```
public EMTopoNodeDn[] getViewChildren()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `VIEW_CHILDREN`

getViewDefaultGeoArea

Returns the `EMTopoNodeAttribute.VIEW_DEFAULT_GEO_AREA` attribute.

```
public EMTopoNodeViewDefaultGeoArea getViewDefaultGeoArea()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `VIEW_DEFAULT_GEO_AREA`

getViewMapConfigFilename

Returns the `EMTopoNodeAttribute.VIEW_MAP_CONFIG_FILENAME` attribute.

```
public String getViewMapConfigFilename()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `VIEW_MAP_CONFIG_FILENAME`

isSeverityPropagated

Returns the `EMTopoNodeAttribute.IS_SEVERITY_PROPAGATED` attribute.

```
public Boolean isSeverityPropagated()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `IS_SEVERITY_PROPAGATED`

setDisplayStatuses

Sets the `EMTopoNodeAttribute.DISPLAY_STATUSES` attribute.

```
public void setDisplayStatuses
                      (EMTopoNodeDisplayStatus displayStatuses[])
```

displayStatuses is the attribute value.

Throws `NullPointerException` if *displayStatuses* is null.

See Also: `DISPLAY_STATUSES`

setDn

Sets the `EMTopoAttribute.DN` attribute.

```
public void setDn(EMObjectDn dn)
```

dn is the attribute value.

Throws:

NullPointerException if *dn* is null.

`ClassCastException` if *dn* is not an instance of `EMTopoNodeDn`.

Overrides `setDn` in class `EMObject.`

See Also: `DN`

setGeographicalLocation

Sets the `EMTopoNodeAttribute.GEOGRAPHICAL_LOCATION` attribute.

```
public void setGeographicalLocation
                              (EMTopoNodeGeoLocation geoLocation)
```

geoLocation is the attribute value.

Throws `NullPointerException` if *geoLocation* is null.

See Also: `GEOGRAPHICAL_LOCATION`

setIsSeverityPropagated

Sets the `EMTopoNodeAttribute.IS_SEVERITY_PROPAGATED` attribute.

```
public void setIsSeverityPropagated(Boolean isSeverityPropagated)
```

isSeverityPropagated is the attribute value.

Throws `NullPointerException` if *isSeverityPropagated* is null.

See Also: `IS_SEVERITY_PROPAGATED`

setLayerName

Sets the `EMTopoNodeAttribute.LAYER_NAME` attribute.

```
public void setLayerName(String layerName)
```

layerName is the attribute value.

Throws `NullPointerException` if *layerName* is null.

See Also: `LAYER_NAME`

setLinks

Sets the `EMTopoNodeAttribute.LINKS` attribute.

```
public void setLinks(EMTopoNodeDn links[])
```

links is the attribute value.

Throws `NullPointerException` if *links* is null.

See Also: `LINKS`

setLogicalLocations

Sets the `EMTopoNodeAttribute.LOGICAL_LOCATIONS` attribute.

```
public void setLogicalLocations
                          (EMTopoNodeLocationInParent locations[])
```

locations is the attribute value.

Throws `NullPointerException` if *locations* is null.

See Also: `LOGICAL_LOCATIONS`

setManagedObjects

Sets the `EMTopoNodeAttribute.MANAGED_OBJECTS` attribute.

```
public void setManagedObjects(String managedObjects[])
```

managedObjects is the attribute value.

Throws `NullPointerException` if *managedObjects* is null.

See Also: `MANAGED_OBJECTS`

setMonitorRotation

Sets the `EMTopoNodeAttribute.MONITOR_ROTATION` attribute.

```
public void setMonitorRotation(Integer monitorRotation)
```

monitorRotation is the attribute value.

Throws `NullPointerException` if *monitorRotation* is null.

See Also:`MONITOR_ROTATION`

setMonitorVisibleChildren

Sets the `EMTopoNodeAttribute.MONITOR_VISIBLE_CHILDREN` attribute.

```
public void setMonitorVisibleChildren
                              (EMTopoNodeDn monitorVisibleChildren[])
    throws EMInvalidArgException
```

monitorVisibleChildren is the attribute value.

Throws:

`NullPointerException` if *monitorVisibleChildren* is null.

`EMInvalidArgException` if maximum number of visible children is exceeded.

See Also: `MONITOR_VISIBLE_CHILDREN`

setArrayVisibleChildren

Sets the `EMTopoNodeAttribute.ARRAY_VISIBLE_CHILDREN` attribute.

```
public void setArrayVisibleChildren
                              (EMTopoNodeDn arrayVisibleChildren[])
```

arrayVisibleChildren is the attribute value.

Throws `NullPointerException` if *arrayVisibleChildren* is null.

See Also: `ARRAY_VISIBLE_CHILDREN`

addArrayVisibleChild

Adds the specified `arrayVisibleChild` to the `EMTopoNodeAttribute.ARRAY_VISIBLE_CHILDREN` attribute.

```
public void addArrayVisibleChild(EMTopoNodeDn arrayVisibleChild)
```

arrayVisibleChild is the attribute value.

See Also: `ARRAY_VISIBLE_CHILDREN`

removeArrayVisibleChild

Removes the specified `arrayVisibleChild` from the `EMTopoNodeAttribute.ARRAY_VISIBLE_CHILDREN` attribute.

```
public void removeArrayVisibleChild
                              (EMTopoNodeDn arrayVisibleChild)
```

arrayVisibleChild is the attribute value.

See Also: `ARRAY_VISIBLE_CHILDREN`

setArrayCellSize

Sets the `EMTopoNodeAttribute.ARRAY_CELL_SIZE` attribute.

```
public void setArrayCellSize(EMTopoNodeArrayCellSize arrayCellSize)
```

arrayCellSize is the attribute value.

Throws `NullPointerException` if *arrayCellSize* is null.

See Also: `ARRAY_CELL_SIZE`

setArrayOrientation

Sets the `EMTopoNodeAttribute.ARRAY_ORIENTATION` attribute.

```
public void setArrayOrientation
                      (EMTopoNodeArrayOrientation arrayOrientation)
```

arrayOrientation is the attribute value.

Throws `NullPointerException` if *arrayOrientation* is null.

See Also: `ARRAY_ORIENTATION`

setArrayNumColumns

Sets the `EMTopoNodeAttribute.ARRAY_NUM_COLUMNS` attribute.

```
public void setArrayNumColumns(Integer arrayNumColumns)
```

arrayNumColumns is the attribute value.

Throws `NullPointerException` if *arrayNumColumns* is null.

See Also:`ARRAY_NUM_COLUMNS`

setBusLogicalLocations

Sets the `EMTopoNodeAttribute.BUS_LOGICAL_LOCATIONS` attribute.

```
public void setBusLogicalLocations
                          (EMTopoNodeLocation busLogicalLocations[])
```

busLogicalLocations is the attribute value.

Throws `NullPointerException` if *busLogicalLocations* is null.

See Also: `BUS_LOGICAL_LOCATIONS`

setName

Sets the `EMTopoNodeAttribute.NAME` attribute.

```
public void setName(String name)
```

name is the attribute value.

Throws `NullPointerException` if *name* is null.

See Also: `NAME`

setParents

Sets the `EMTopoNodeAttribute.PARENTS` attribute.

```
public void setParents(EMTopoNodeDn parents[])
```

parents is the attribute value.

Throws `NullPointerException` if *parents* is null.

See Also: `PARENTS`

setPropagatePeers

Sets the `EMTopoNodeAttribute.PROPAGATE_PEERS` attribute.

```
public void setPropagatePeers(EMTopoNodeDn propagatePeers[])
```

propagatePeers is the attribute value.

Throws `NullPointerException` if *propagatePeers* is null.

See Also: `PROPAGATE_PEERS`

setState

Sets the `EMTopoNodeAttribute.STATE` attribute.

```
public void setState(Integer state)
```

state is the attribute value.

Throws `NullPointerException` if *state* is null.

See Also: `STATE`

setTypeName

Sets the `EMTopoNodeAttribute.TYPE_NAME` attribute.

```
public void setTypeName(String typeName)
```

typeName is the attribute value.

Throws `NullPointerException` if *typeName* is null.

See Also: `TYPE_NAME`

setUserData

Sets the `EMTopoNodeAttribute.USER_DATA` attribute.

```
public void setUserData(EMTopoNodeUserDatum userData[])
```

userData is the attribute value.

Throws `NullPointerException` if *userData* is null.

See Also: `USER_DATA`

setViewBackgroundImageFilename

Sets the `EMTopoNodeAttribute.VIEW_BACKGROUND_IMAGE_FILENAME` attribute.

```
public void setViewBackgroundImageFilename(String filename)
```

filename is the attribute value.

Throws `NullPointerException` if *filename* is null.

See Also: `VIEW_BACKGROUND_IMAGE_FILENAME`

setViewMapConfigFilename

Sets the `EMTopoNodeAttribute.VIEW_MAP_CONFIG_FILENAME` attribute.

```
public void setViewMapConfigFilename(String filename)
```

filename is the attribute value.

Throws `NullPointerException` if *filename* is null.

See Also: `VIEW_MAP_CONFIG_FILENAME`

setViewDefaultGeoArea

Sets the `EMTopoNodeAttribute.VIEW_DEFAULT_GEO_AREA` attribute.

```
public void setViewDefaultGeoArea
                      (EMTopoNodeViewDefaultGeoArea defaultGeoArea)
```

defaultGeoArea is the attribute value.

Throws `NullPointerException` if *defaultGeoArea* is null.

See Also: `VIEW_DEFAULT_GEO_AREA`

addEMIndividualNodeListener

Adds the listener to the list of objects to be called when this topology node is changed or deleted.

```
public void addEMIndividualNodeListener
                              (EMIndividualNodeListener listener)
    throws EMTopoServiceException
```

listener is listener of topology node events.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

See Also: `EMIndividualNodeListener`

removeEMIndividualNodeListener

Removes the specified listener from the list of objects to be called when this topology node is changed or deleted.

```
public void removeEMIndividualNodeListener
                              (EMIndividualNodeListener listener)
    throws EMTopoServiceException
```

listener is listener of topology events.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

See Also: `EMIndividualNodeListener`

toString

Returns a String representation of the `EMTopoNode`.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `Object`.

4.35 EMTopoNodeArrayCellSize Class

public class EMTopoNodeArrayCellSize

extends Object implements Cloneable, Serializable, Comparable

The com.sun.em.api.topology.EMTopoNodeArrayCellSize class is a holder class for the width (in characters) and height (in pixels) of an array cell. Because this is simply a holder class, the data members are public.

4.35.1 Variables

width

```
public int width
```

height

```
public int height
```

4.35.2 Constructors

EMTopoNodeArrayCellSize

Creates an instance of EMTopoNodeArrayCellSize with the specified width in characters and height in pixels.

```
public EMTopoNodeArrayCellSize(int width, int height)
```

width is the width in characters. Value 0 means the cell width will be set to the minimum width necessary for the widest cell label to be visible.

height is the height in pixels.

4.35.3 Methods

clone

Return a clone of this `EMTopoNodeArrayCellSize` object.

```
public Object clone()
```

Returns a clone of this `EMTopoNodeArrayCellSize`.

Overrides `clone` in class `Object`.

equals

Compares this object against the specified object. The result is true if, and only if, the argument is not null and is a `EMTopoNodeArrayCellSize` object that contains the same values for width and height.

```
public boolean equals(Object obj)
```

obj is the object to compare with.

Returns true if the objects are the same; otherwise returns false.

Overrides `equals` in class `Object`.

compareTo

Compares this `EMTopoNodeArrayCellSize` with another object that implements `Comparable` interface. Order is determined by first comparing the width, then height if the widths are equal.

```
public int compareTo(Comparable comparable)
```

Returns 0 if the objects are identical, less than 0 if this object is less than the specified object, and greater than 0 if this object is greater than the specified object.

Throws `ClassCastException` if the object parameter is not of class `EMTopoNodeArrayCellSize`.

toString

Returns a String representation of the EMTopoNodeArrayCellSize.

```
public String toString()
```

Returns the String representation.

Overrides toString in class Object

4.36 EMTopoNodeArrayOrientation Class

public class EMTopoNodeArrayOrientation

extends Object implements Serializable

The com.sun.em.api.topology.EMTopoNodeArrayOrientation class represents the orientation of an array, either horizontal or vertical. This indicates whether the topology nodes grouped by the array should be laid down row by row or column by column.

4.36.1 Variables

HORIZONTAL

The HORIZONTAL orientation means that the topology nodes grouped by the array will be laid down row by row.

```
public static final EMTopoNodeArrayOrientation HORIZONTAL
```

VERTICAL

The VERTICAL orientation means that the topology nodes grouped by the array will be laid down column by column.

```
public static final EMTopoNodeArrayOrientation VERTICAL
```

4.36.2 Methods

equals

Compares this `EMTopoNodeArrayOrientation` against the specified object. The result is true if, and only if, the argument is not null and is a `EMTopoNodeArrayOrientation` object with the same value as this `EMTopoNodeArrayOrientation`.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, false otherwise.

Overrides `equals` in class `Object`.

toString

Returns a String representation of the `EMTopoNodeArrayOrientation`.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `Object`.

4.37 EMTopoNodeAttribute Class

public class `EMTopoNodeAttribute`

extends `EMTopoAttribute` implements `Serializable`

The `com.sun.em.api.topology.EMTopoNodeAttribute` class represents a single attribute of the `EMTopoNode` persistent object class (POC).

4.37.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMObjectAttribute
           |
           +----com.sun.em.api.topology.EMTopoAttribute
                   |
                   +----com.sun.em.api.topology.EMTopoNodeAttribute
```

4.37.2 Variables

ARRAY_VISIBLE_CHILDREN

Subset of the `CHILDREN` list of nodes that should be included in the array node's cells when displayed in the Viewer. The `ARRAY_VISIBLE_CHILDREN` nodes are placed in order into the array cells according to `ARRAY_ORIENTATION`. Note that this attribute is valid only if the node is an array, that is, if `EMTopoType.isArray(platform,TYPE_NAME)` returns true.

```
public static final EMTopoNodeAttribute ARRAY_VISIBLE_CHILDREN
```

ARRAY_HIDDEN_CHILDREN

List of remaining nodes after `ARRAY_VISIBLE_CHILDREN` is subtracted from `CHILDREN`. These nodes do not appear in array cells, even if there are empty cells. Note that this attribute is valid only if the node is an array, that is, if `EMTopoType.isArray(platform,TYPE_NAME)` returns true.

```
public static final EMTopoNodeAttribute ARRAY_HIDDEN_CHILDREN
```

ARRAY_ORIENTATION

The orientation of the array, either horizontal or vertical. It indicates whether the topology nodes grouped by the array should be laid down row by row or column by column. Note that this attribute is valid only if the node is an array, that is, if `EMTopoType.isArray(platform,TYPE_NAME)` returns true.

```
public static final EMTopoNodeAttribute ARRAY_ORIENTATION
```

ARRAY_NUM_COLUMNS

The number of columns or the number of rows for the horizontal or vertical orientation, respectively. Note that this attribute is valid only if the node is an array, that is, if `EMTopoType.isArray(platform,TYPE_NAME)` returns true.

```
public static final EMTopoNodeAttribute ARRAY_NUM_COLUMNS
```

ARRAY_CELL_SIZE

The width (in characters) and height (in pixels) of the array. Note that this attribute is valid only if the node is an array, that is,if `EMTopoType.isArray(platform,TYPE_NAME)` returns true.

```
public static final EMTopoNodeAttribute ARRAY_CELL_SIZE
```

BUS_LOGICAL_LOCATIONS

List of (x,y) points which define the bus's shape. The points are constrained so that all line segments are alternately horizontal or vertical. Note that this attribute is valid only if the node is bus, that is, if `EMTopoType.isBus(platform,TYPE_NAME)` returns true.

```
public static final EMTopoNodeAttribute BUS_LOGICAL_LOCATIONS
```

CHILDREN

List of topology nodes that are contained by this node.

```
public static final EMTopoNodeAttribute CHILDREN
```

CMIP_AGENTS

List of CMIP agents which have managed objects listed as part of MANAGED_OBJECTS attribute.

```
public static final EMTopoNodeAttribute CMIP_AGENTS
```

DISPLAY_STATUSES

A list of user-defined string labels and integer value pairs, that for example, "CPUUsage", 45. This attribute can be used by developers to store a limited amount of integer data with each topology node.

```
public static final EMTopoNodeAttribute DISPLAY_STATUSES
```

GEOGRAPHICAL_LOCATION

The latitude and longitude in degrees floating-point of the location of this node, for example, 45.6780 degrees latitude, -152.234 degrees longitude.

```
public static final EMTopoNodeAttribute GEOGRAPHICAL_LOCATION
```

IS_SEVERITY_PROPAGATED

If true, then the node's severity will be propagated to each of its parents and each of its PROPAGATE_PEERS where it will factor in the calculation of their PROPAGATED_SEVERITY.

```
public static final EMTopoNodeAttribute IS_SEVERITY_PROPAGATED
```

LAYER_NAME

The name of the logical layer that the node belongs to.

```
public static final EMTopoNodeAttribute LAYER_NAME
```

LINKS

For links (topology nodes with `TYPE_NAME` equal to `EMTopoTypeDn.LINK`), `LINKS` is a list of topology nodes that the link is connected to. For non-link topology nodes, `LINKS` is a list of links the topology node is connected to.

```
public static final EMTopoNodeAttribute LINKS
```

LOGICAL_LOCATIONS

A list of parent node-coordinate pairs which locate a node within each parent node's view in a logical coordinate space. Note that a node can have a different logical coordinate for each parent that contains it, whereas the node can only have one geographical coordinate.

```
public static final EMTopoNodeAttribute LOGICAL_LOCATIONS
```

MANAGED_OBJECTS

List of managed object DNs in ASCII slash format associated with this node. Note that a managed object may be associated with multiple nodes. To find out the topology nodes that a managed object *dn* is associated with, call `EMTopoPlatform.findNodesByMananagedObject()`.

```
public static final EMTopoNodeAttribute MANAGED_OBJECTS
```

MONITOR_HIDDEN_CHILDREN

List of remaining nodes after MONITOR_VISIBLE_CHILDREN is subtracted from CHILDREN. These nodes do not appear in a monitor section, even if there are empty sections. Note that this attribute is valid only if the node is a monitor, that is, if EMTopoType.isMonitor(platform,TYPE_NAME) returns true.

```
public static final EMTopoNodeAttribute MONITOR_HIDDEN_CHILDREN
```

MONITOR_MAX_VISIBLE_CHILDREN

Maximum number of sections supported by the particular type of monitor. Note that this attribute is valid only if the node is a monitor, that is, if EMTopoType.isMonitor(platform,TYPE_NAME) returns true.

```
public static final EMTopoNodeAttribute
                                MONITOR_MAX_VISIBLE_CHILDREN
```

MONITOR_ROTATION

Number of degrees to rotate the monitor node when displaying in the Viewer's canvas. The starting point for the rotation is at 12 o'clock. Note that this attribute is valid only if the node is a monitor, that is,if EMTopoType.isMonitor(platform,TYPE_NAME) returns true.

```
public static final EMTopoNodeAttribute MONITOR_ROTATION
```

MONITOR_VISIBLE_CHILDREN

Subset of the CHILDREN list of nodes that should be included in the monitor node's sections when displayed in the Viewer. The MONITOR_VISIBLE_CHILDREN nodes are placed, in order, into the monitor sections, starting at 12 o'clock and moving clockwise until all monitor sections are filled or there are no more nodes in MONITOR_VISIBLE_CHILDREN. Note that this attribute is valid only if the node is a monitor, that is, if EMTopoType.isMonitor(platform,TYPE_NAME) returns true.

```
public static final EMTopoNodeAttribute MONITOR_VISIBLE_CHILDREN
```

NAME

The administrative name of this node. Note that the `NAME` is not unique; multiple topology nodes may have the same name. To find out the topology nodes that a managed object *dn* is associated with, call `EMTopoPlatform.findNodesByName()`. Note that the name "Root" is reserved for the topology node that is the root of the topology graph.

```
public static final EMTopoNodeAttribute NAME
```

PARENTS

List of topology nodes that contain this node in the topology directed acyclic graph (DAG).

```
public static final EMTopoNodeAttribute PARENTS
```

PROPAGATE_PEERS

List of topology nodes to propagate this topology node's severity to if `IS_SEVERITY_PROPAGATED` is true.

```
public static final EMTopoNodeAttribute PROPAGATE_PEERS
```

PROPAGATED_SEVERITY

The highest severity amongst the topology node's severity, the severities of all the node's children whose `IS_SEVERITY_PROPAGATED` attribute is true, and the severities of all nodes whose PROPAGATE_PEERS attribute include this node and `IS_SEVERITY_PROPAGATED` attribute is true. Note that normally an application never sets the `PROPAGATED_SEVERITY` attribute; the Alarm Service automatically updates the `PROPAGATED_SEVERITY` attribute based on the above criteria.

```
public static final EMTopoNodeAttribute PROPAGATED_SEVERITY
```

RPC_AGENTS

List of RPC agents which have managed objects listed as part of the MANAGED_OBJECTS attribute.

```
public static final EMTopoNodeAttribute RPC_AGENTS
```

SNMP_AGENTS

List of SNMP agents that have managed objects listed as part of the MANAGED_OBJECTS attribute.

```
public static final EMTopoNodeAttribute SNMP_AGENTS
```

STATE

Can be used to store an integer value.

```
public static final EMTopoNodeAttribute STATE
```

SEVERITY

The highest severity alarm posted against any of the topology node's MANAGED_OBJECTS. Note that normally an application never sets the SEVERITY attribute; the Alarm Service automatically updates the SEVERITY attribute to reflect the current outstanding alarms posted against the topology node.

```
public static final EMTopoNodeAttribute SEVERITY
```

TOPOLOGY_PATHNAMES

List of all topology pathnames for the node. At a minimum, there will be one pathname for each parent. However, the actual number of pathnames may be higher since each parent can also have more than one parent, for example,

"/Root/Internet/129.146.74.0/host-45"

"/Root/hosts/host-45"

```
public static final EMTopoNodeAttribute TOPOLOGY_PATHNAMES
```

TYPE_NAME

Type name of this node.

```
public static final EMTopoNodeAttribute TYPE_NAME
```

USER_DATA

A list of user-defined attribute name-value pairs. A single attribute name-value pair should appear for each attribute listed in `EMTopoType.USER_DATA_ATTRIBUTE_NAMES` attribute of this node's `EMTopoType` identified by the `TYPE_NAME` attribute.

```
public static final EMTopoNodeAttribute USER_DATA
```

VIEW_BACKGROUND_IMAGE_FILENAME

Absolute pathname of the raster file image to be displayed when the Viewer is displaying this view node in logical view mode. Note that this attribute is valid only if the node is a view, that is if `EMTopoType.isView(platform,TYPE_NAME)` returns true.

```
public static final EMTopoNodeAttribute
                                  VIEW_BACKGROUND_IMAGE_FILENAME
```

VIEW_CHILDREN

Subset of CHILDREN whose TYPE_NAME is a view; that is, EMTopoType.isView(TYPE_NAME) returns true.

```
public static final EMTopoNodeAttribute VIEW_CHILDREN
```

VIEW_DEFAULT_GEO_AREA

Default geographical area (specified as a center and view width in km) to be displayed when the VIEW_MAP_CONFIG_FILENAME is first displayed. Note that this attribute is valid only if the node is a view, that is, if EMTopoType.isView(platform,TYPE_NAME) returns true.

```
public static final EMTopoNodeAttribute VIEW_DEFAULT_GEO_AREA
```

VIEW_MAP_CONFIG_FILENAME

Absolute pathname of geographical map configuration (GMC) file to be displayed when the Viewer is displaying this view node in geographical view mode. Note that this attribute is valid only if the node is a view, that is, if EMTopoType.isView(platform,TYPE_NAME) returns true.

```
public static final EMTopoNodeAttribute VIEW_MAP_CONFIG_FILENAME
```

4.37.3 Methods

toString

Returns a String representation of the EMTopoNodeAttribute.

```
public String toString()
```

Returns the String representation.

Overrides toString in class EMTopoAttribute

4.38 EMTopoNodeAttributeSet Class

public class EMTopoNodeAttributeSet

extends EMAttributeSet

implements Cloneable, Serializable

The com.sun.em.api.topology.EMTopoNodeAttributeSet class implements is an abstract class which forms the basis for the attributes set classes of each EMTopoNode. EMTopoNodeAttributeSet is used in the Topology API to communicate which attributes of an EMTopoNode an API method should operate on.

4.38.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMAttributeSet
           |
           +----com.sun.em.api.topology.EMTopoNodeAttributeSet
```

4.38.2 Constructors

EMTopoNodeAttributeSet

Creates an EMTopoNodeAttributeSet that contains no attributes.

```
public EMTopoNodeAttributeSet()
```

4.38.3 Methods

clone

Returns a clone of this EMTopoNodeAttributeSet object.

```
public Object clone()
```

Overrides clone in class EMAttributeSet.

elements

Returns an enumeration of the attributes in this set.

```
public Enumeration elements()
```

Overrides elements in class EMAttributeSet.

4.39 EMTopoNodeBatchLoaderEvent Class

public class EMTopoNodeBatchLoaderEvent

extends EventObject implements Cloneable, Serializable

The com.sun.em.api.topology.EMTopoNodeBatchLoaderEvent class contains the event information for batch loading of topology nodes. When EMTopoNodeBatchLoaderListener.batchReceived() method is called, an instance of EMTopoNodeBatchLoaderEvent is passed as an argument. The event contains the EMTopoNodes that have been loaded in this batch.

See Also: EMTopoNodeBatchLoaderListener, loadNodesInBatches

4.39.1 Inheritance

```
java.lang.Object
   |
   +----java.util.EventObject
           |
           +----com.sun.em.api.topology.EMTopoNodeBatchLoaderEvent
```

4.39.2 Constructors

EMTopoNodeBatchLoaderEvent

Creates an instance of `EMTopoNodeBatchLoaderEvent`.

```
public EMTopoNodeBatchLoaderEvent
                              (Object source, EMTopoNodenodes[])
```

source is the source of the event.

nodes is the nodes loaded in this batch.

4.39.3 Methods

getNodes

Returns the topology nodes from this batch. To find out how many topology nodes, call the array *size()* method.

```
public EMTopoNode[] getNodes()
```

Returns the nodes from this batch.

4.40 EMTopoNodeBatchLoaderListener Interface

public interface `EMTopoNodeBatchLoaderListener`

extends `EventListener`

The `com.sun.em.api.topology.EMTopoNodeBatchLoaderListener` listener is the interface for asynchronously loading arbitrary numbers of topology nodes in fixed-size batches.

See Also: `EMTopoNodeBatchLoaderEvent`, `loadNodesInBatches`

4.40.1 Methods

batchReceived

Invoked once *batchSize* number of topology nodes have been loaded.

```
public void batchReceived
                              (EMTopoNodeBatchLoaderEvent event)
```

event is the event information.

4.41 EMTopoNodeDisplayStatus Class

public class `EMTopoNodeDisplayStatus`

extends `Object` implements `Cloneable`, `Serializable`, `Comparable`

The `com.sun.em.api.topology.EMTopoNodeDisplayStatus` class is a holder class for the string label and integer value pairs which make up `EMTopoNode`'s `DISPLAY_STATUSES` attribute. Because this is simply a holder class, the data members are public.

See Also: `DISPLAY_STATUSES`, `getDisplayStatuses`, `setDisplayStatuses`

4.41.1 Variables

`label`

The label, that is, "CPU Usage".

```
public String label
```

`value`

The integer value associated with the label.

```
public int value
```

4.41.2 Constructors

`EMTopoNodeDisplayStatus`

Creates an instance of `EMTopoNodeDisplayStatus` with the specified label and value pair.

```
public EMTopoNodeDisplayStatus(String label, int value)
```

label is the label.

value is the value.

4.41.3 Methods

clone

Returns a clone of this `EMTopoNodeDisplayStatus` object.

```
public Object clone()
```

Overrides `clone` in class `Object`.

equals

Compares this object against the specified object. The result is true if, and only if, the argument is not null and is a `DisplayStatus` object that contains the same values for label and value

```
public boolean equals(Object obj)
```

obj is the object to compare with.

Returns true if the objects are the same; otherwise returns false.

Overrides `equals` in class `Object`.

compareTo

Compares this `EMTopoNodeDisplayStatus` with another object which implements `Comparable` interface. Order is determined by first lexicographically comparing the label and then the value if the labels are equal.

```
public int compareTo(Comparable comparable)
```

Returns 0 if the objects are identical, less than 0 if this object is less than the specified object, greater than 0 if this object is greater than the specified object.

Throws `ClassCastException` if the object parameter is not of class `EMTopoNodeDisplayStatus`.

toString

Returns a String representation of the `EMTopoNodeDisplayStatus`.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `Object`.

4.42 EMTopoNodeDn Class

public final class `EMTopoNodeDn`

extends `EMObjectDn`

implements `Serializable`, `Comparable`

An instance of the `com.sun.em.api.topology.EMTopoNodeDn` class uniquely identifies a topology node.

4.42.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.topology.EMObjectDn
           |
           +----com.sun.em.api.topology.EMTopoNodeDn
```

4.42.2 Constructors

EMTopoNodeDn

Creates an EMTopoNodeDn with the specified system name and unique ID.

```
public EMTopoNodeDn(String systemName, int uniqueId)
```

systemName is the name of the MIS where the topology node is stored.

uniqueId is the unique identifier of the topology node within the MIS.

4.42.3 Methods

equals

Compares this object against the specified object. The result is true if, and only if, the argument is not null and is a EMTopoNodeDn object which identifies the same topology node as this object.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, false otherwise.

Overrides *equals* in class Object.

compareTo

Compares this EMTopoNodeDn with another object that implements Comparable interface. Order is determined by comparing the system name lexicographically, and then comparing the unique identifier.

```
public int compareTo(Comparable comparable)
```

Returns 0 if the objects are identical, less than 0 if this object is 'less than' the specified object, greater than 0 if this object is 'greater than' the specified object.

Throws `ClassCastException` if the object parameter is not of class `EMTopoNodeDn`.

getSystemName

Returns the name of the MIS where the topology node is stored.

```
public String getSystemName()
```

Returns the MIS name.

Overrides `getSystemName` in class `EMObjectDn`.

getUniqueName

Returns the integer identifier of the topology node as a string. This identifier is unique within the MIS.

```
public String getUniqueName()
```

Returns the unique identifier.

Overrides `getUniqueName` in class `EMObjectDn.`

getUniqueId

```
public int getUniqueId()
```

Returns the integer identifier of the topology node. This identifier is unique within the MIS.

Returns the unique identifier.

hashCode

Returns a hashcode for this object.

```
public int hashCode()
```

Returns the hashcode.

Overrides `hashCode` in class `Object`.

toString

Returns a String representation of the `EMTopoNodeDn`.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `Object`.

4.43 EMTopoNodeEvent Class

public class `EMTopoNodeEvent`

extends `EventObject`

implements `Cloneable`

The `com.sun.em.api.topology.EMTopoNodeEvent` class contains information on topology node creation, deletion, and change events. When `EMTopoNodeListener` `nodeCreated()`, `nodeDeleted()`, or `nodeChanged()` methods are called, they are passed an instance of `EMTopoNodeEvent` containing the event information. The event information includes the type of event (creation, deletion, change), the node *dn* the event occured on, and for change events, an instance of `EMTopoNode` containing the new values of the modified attributes.

See Also: `EMTopoNodeListener`, `addEMTopoNodeListener`, `removeEMTopoNodeListener`

4.43.1 Inheritance

```
java.lang.Object
   |
   +----java.util.EventObject
           |
           +----com.sun.em.api.topology.EMTopoNodeEvent
```

4.43.2 Variables

OBJECT_CREATED

A topology node was created in the MIS.

```
public static final int OBJECT_CREATED
```

OBJECT_DELETED

A topology node was deleted in the MIS.

```
public static final int OBJECT_DELETED
```

OBJECT_CHANGED

One or more attributes of a topology node were changed in the MIS.

```
public static final int OBJECT_CHANGED
```

4.43.3 Constructors

EMTopoNodeEvent

Creates an instance of EMTopoNodeEvent.

```
public EMTopoNodeEvent
                   (Object source, int eventType,EMTopoNodeDn nodeDn)
```

source is the source of the event.

eventType is the type of event.

nodeDn is the unique identifier of the topology node the event occured on.

EMTopoNodeEvent

Creates an instance of EMTopoNodeEvent. This version of the constructor should only be used for OBJECT_CHANGED events.

```
public EMTopoNodeEvent
(Object source, int eventType, EMTopoNodeDnnodeDn, EMTopoNode changes)
```

source is the object which is the source of the event.

eventType is the type of topology node event.

nodeDn is the unique identifier of the topology node the event occured on.

changes is the new values of the modified attributes.

4.43.4 Methods

getEventType

Returns the type of topology node event.

```
public int getEventType()
```

Returns type of topology node event.

getNodeDn

Returns the unique identifier of the topology node the event occurred on.

```
public EMTopoNodeDn getNodeDn()
```

Returns unique identifier of topology node.

getChanges

For `OBJECT_CHANGED` events, returns the new values of the modified attributes; otherwise, returns null. To find out which attributes were changed, call `EMTopoNode.getActiveAttributes()`. The normal `EMTopoNode` getter methods may then be used to get the new values of the active attributes.

```
public EMTopoNode getChanges()
```

Returns the new values of the modified attributes.

isView

This method should only be used for OBJECT_CHANGED events. It returns true if the topology node represents a view; otherwise it returns false. This method is meant to save a callback to the server side when the information about the node being a view or not is needed.

```
public boolean isView()
```

Returns the boolean value indicating if topology node represents a view.

getViewChildren

This method should only be used for OBJECT_CHANGED events and when the changed attributes include EMTopoNodeAttribute.CHILDREN. It returns a list of the view children node Dns of the topology node that the event is about. Note that its view children node Dns are only a subset of its children node Dns. This method is meant to save a call back to the server side when view children is needed.

```
public EMTopoNodeDn[] getViewChildren()
```

Returns the array of view children node Dns of the topology node

toString

Returns a String representation of the EMTopoNodeEvent.

```
public String toString()
```

Returns the String representation.

Overrides toString in class EventObject.

4.44 EMTopoNodeGeoLocation Class

public class EMTopoNodeGeoLocation

extends Object implements Cloneable, Serializable

The com.sun.em.api.topology.EMTopoNodeGeoLocation class is a holder class for the longitude, latitude value of EMTopoNode's GEOGRAPHICAL_LOCATION attribute. Because this is simply a holder class, the data members are public.

See Also: GEOGRAPHICAL_LOCATION, getGeographicalLocation, setGeographicalLocation

4.44.1 Variables

longitude

The longitude (x) coordinate. Valid range is -180.0 degrees to 180.0 degrees.

```
public double longitude
```

latitude

```
public double latitude
```

4.44.2 Constructors

EMTopoNodeGeoLocation

Creates an instance of `EMTopoNodeGeoLocation` with the specified longitude and latitude.

```
public EMTopoNodeGeoLocation(double longitude, double latitude)
```

longitude is the longitude.

latitude is the latitude.

4.44.3 Methods

clone

Returns a clone of this `EMTopoNodeGeoLocation` object.

```
public Object clone()
```

Returns a clone of this `EMTopoNodeGeoLocation`.

Overrides `clone` in class `Object`.

equals

Compares this object against the specified object. The result is true if, and only if, the argument is not null and is a `EMTopoNodeGeoLocation` object that contains the same values for longitude and latitude .

```
public boolean equals(Object obj)
```

obj is the object to compare with.

Returns true if the objects are the same; otherwise returns false.

Overrides `equals` in class `Object`.

toString

Returns a String representation of the `EMTopoNodeGeoLocation` object.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `Object`.

4.45 EMTopoNodeListener Interface

public interface EMTopoNodeListener

extends EventListener

The `com.sun.em.api.topology.EMTopoNodeListener` listener is the interface for receiving topology node events.

See Also: `EMTopoNodeEvent`, `addEMTopoNodeListener`, `removeEMTopoNodeListener`

4.45.1 Methods

nodeCreated

Invoked when a topology node is created in the MIS.

```
public void nodeCreated(EMTopoNodeEvent event)
```

event is the event information.

nodeDeleted

Invoked when a topology node is deleted from the MIS.

```
public abstract void nodeDeleted(EMTopoNodeEvent event)
```

event is the event information.

nodeChanged

Invoked when one or more attributes of a topology node were changed in the MIS.

```
public abstract void nodeChanged(EMTopoNodeEvent event)
```

event is the event information.

4.46 EMTopoNodeLocation Class

public class `EMTopoNodeLocation`

extends `Object` implements `Cloneable`, `Serializable`, `Comparable`

The `com.sun.em.api.topology.EMTopoNodeLocation` class is a holder class for the x, y, and z coordinates which mark a logical location. Because this is simply a holder class, the data members are public.

4.46.1 Variables

x

```
public int x
```

y

```
public int y
```

z

```
public int z
```

4.46.2 Constructors

EMTopoNodeLocation

Creates an instance of `EMTopoNodeLocation` with the specified x, y, and z coordinates.

```
public EMTopoNodeLocation(int x, int y, int z)
```

x is the x coordinate.

y is the y coordinate.

z is the z coordinate.

4.46.3 Methods

clone

Return a clone of this `EMTopoNodeLocation` object.

```
public Object clone()
```

Returnsa clone of this `EMTopoNodeLocation`.

Overrides `clone` in class `Object`.

equals

Compares this object against the specified object. The result is true if, and only if, the argument is not null and is a `EMTopoNodeLocation` object that contains the same values for x , y , and z .

```
public boolean equals(Object obj)
```

obj is the object to compare with.

Returns true if the objects are the same; otherwise returns false.

Overrides `equals` in class `Object`.

compareTo

Compares this `EMTopoNodeLocation` with another object which implements `Comparable` interface. Order is determined by first comparing the x coordinate, then y coordinate if the x coordinates are equal, and finally the z coordinate if both the x and y coordinates are equal.

```
public int compareTo(Comparable comparable)
```

Returns 0 if the objects are identical, less than 0 if this object is less than the specified object, greater than 0 if this object is greater than the specified object.

Throws `ClassCastException` if the object parameter is not of class `EMTopoNodeLocation`.

toString

Returns a String representation of the `EMTopoNodeLocation` object.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `Object`.

4.47 EMTopoNodeLocationInParent Class

public class `EMTopoNodeLocationInParent`

extends `Object` implements `Cloneable`, `Serializable`, `Comparable`

The `com.sun.em.api.topology.EMTopoNodeLocatioInParent` class is a holder class for the node *dn* and logical location pairs which make up the EMTopoNode's `LOGICIAL_LOCATIONS` attribute. Because this is simply a holder class, the data members are public.

See Also: `LOGICAL_LOCATIONS`, `getLogicalLocation`, `getLogicalLocations`, `setLogicalLocations`

4.47.1 Variables

parent

The relevant topology node's logical view.

```
public EMTopoNodeDn parent
```

location

The logical location.

```
public EMTopoNodeLocation location
```

4.47.2 Constructors

EMTopoNodeLocationInParent

Creates an instance of `EMTopoNodeLocationInParent` with the specified view node and logical location.

```
public EMTopoNodeLocationInParent
                    (EMTopoNodeDn parent, EMTopoNodeLocation location)
```

view is the view node.

location is the logical location.

4.47.3 Methods

clone

Returns a clone of this `EMTopoNodeLocationInParent` object.

```
public Object clone()
```

Overrides `clone` in class `Object`.

equals

Compares this object against the specified object. The result is true if, and only if, the argument is not null and is a `LocationInParent` object that contains the same values for parent and location

```
public boolean equals(Object obj)
```

obj is the object to compare with.

Returns true if the objects are the same; otherwise returns false.

Overrides *equals* in class `Object`.

compareTo

Compares this `EMTopoNodeLocationInParent` with another object that implements `Comparable` interface. Order is determined by first comparing the parent *dn*, then the location if the parent *dns* are equal.

```
public int compareTo(Comparable comparable)
```

Returns 0 if the objects are identical, less than 0 if this object is less than the specified object, greater than 0 if this object is greater than the specified object.

Throws `ClassCastException` if the object parameter is not of class `EMTopoNodeLocationInParent`.

toString

Returns a String representation of the `EMTopoNodeLocationInParent`.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `Object`.

4.48 EMTopoNodeUserDatum Class

public class EMTopoNodeUserDatum

extends Object implements Cloneable, Serializable, Comparable

The com.sun.em.api.topology.EMTopoNodeUserDatum class is a holder class for the attribute name and value pairs which make up EMTopoNode's USER_DATA attribute. Because this is simply a holder class, the data members are public.

See Also: USER_DATA, getUserData, setUserData

4.48.1 Constructors

EMTopoNodeUserDatum

Creates an instance of EMTopoNodeUserDatum with the specified attribute name and value.

```
public EMTopoNodeUserDatum
                    (String attributeName, AbstractData value)
```

attributeName is the name of the attribute.

value is the attribute value.

4.48.2 Methods

getAttributeName

Returns the name of the user data attribute. It is the GDMO attribute name. In order to minimize possibility of namespace clashes, it should be a fully specified name with documentName:attributeName, for example, "Rec. X.721 | ISO/IEC 10165-2 : 1992":perceivedSeverity.

```
public String getAttributeName()
```

Returns the `attributeName`.

getAttributeValue

Returns the value of the user data attribute in the form of an `AbstractData` object.

```
public AbstractData getAttributeValue()
```

Returns the attributeValue.

clone

Returns a clone of this `EMTopoNodeUserDatum` object.

```
public Object clone()
```

Overrides `clone` in class `Object`.

equals

Compares this object against the specified object. The result is true if, and only if, the argument is not null and is a `UserDatum` object that contains the same values for *attributeName* and value.

```
public boolean equals(Object obj)
```

obj is the object to compare with.

Returns true if the objects are the same; otherwise false.

Overrides `equals` in class `Object`.

compareTo

Compares this EMTopoNodeUserDatum with another object that implements Comparable interface. Order is determined by first lexicographically comparing the *attributeName*, and then the attribute value if the attributeNames are equal.

```
public int compareTo(Comparable comparable)
```

Returns 0 if the objects are identical, less than 0 if this object is less than the specified object, greater than 0 if this object is greater than the specified object.

Throws ClassCastException if the object parameter is not of class EMTopoNodeUserDatum.

toString

Returns a String representation of the EMTopoNodeUserDatum object.

```
public String toString()
```

Returns the String representation.

Overrides toString in class Object.

4.49 EMTopoNodeViewDefaultGeoArea Class

public class EMTopoNodeViewDefaultGeoArea

extends Object implements Cloneable, Serializable.

The com.sun.em.api.topology.EMTopoNodeViewDefaultGeoArea class is a holder class for the center and view width (in km) values of EMTopoNode's VIEW_DEFAULT_GEO_AREA attribute. Because this is simply a holder class, the data members are public.

See Also: VIEW_DEFAULT_GEO_AREA, getViewDefaultGeoArea, setViewDefaultGeoArea

4.49.1 Variables

center

The center of the default view for geographical mode.

```
public EMTopoNodeGeoLocation center
```

widthInKm

The number of kilometers that should be displayed horizontally from edge-to-edge in the Viewer's Canvas. This variable controls how far zoomed in/out the default view will appear in geographical mode.

```
public double widthInKm
```

4.49.2 Constructors

EMTopoNodeViewDefaultGeoArea

Creates an instance of `EMTopoNodeViewDefaultGeoArea` with the specified center and view width (in km).

```
public EMTopoNodeViewDefaultGeoArea
                  (EMTopoNodeGeoLocation center,double widthInKm)
```

center is the center of the default geographical view.

widthInKm is the number of kilometers that should be displayed horizontally from edge-to-edge in the Viewer's Canvas.

4.49.3 Methods

clone

Returns a clone of this `EMTopoNodeViewDefaultGeoArea` object.

```
public Object clone()
```

equals

Compares this object against the specified object. The result is true if, and only if, the argument is not null and is a `EMTopoNodeViewDefaultGeoArea` object that contains the same values for *center* and *widthInKm*.

```
public boolean equals(Object obj)
```

obj is the object to compare with.

Returns true if the objects are the same, otherwise returns false.

Overrides `equals` in class `Object`.

toString

Returns a String representation of the `EMTopoNodeViewDefaultGeoArea` object.

```
public String toString()
```

Overrides `toString` in class `Object`.

4.50 EMTopoPlatform Class

com.sun.em.api.topology.public class `EMTopoPlatform`

extends `Object`

The `com.sun.em.api.topology.EMTopoPlatform` class represents the Topology API as a whole. Constructors of Persistent Object Classes (POC) and static methods require the `EMTopoPlatform` as the first argument. More than one `EMTopoPlatform` can be instantiated and used simultaneously just like Java PMI's Platform. This allows a program to connect directly to two different MISs and access their topology information without setting up MIS to MIS Communication (MMC) between the two.

The `EMTopoPlatform` provides methods to query which MISs are visible through the connection to the local MIS, and register listeners to be called when MMC is added or removed between the local MIS and another MIS.

4.50.1 Constructors

EMTopoPlatform

Constructs an instance of the `EMTopoPlatform` class.

```
public EMTopoPlatform(Platform platform)
    throws EMTopoServiceException
```

platform is PMI Platform class.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

4.50.2 Methods

addEMPlatformConfigListener

Adds the specified listener to the list of objects to be called when MMC (MIS to MIS Communication) is added or removed between the local MIS and another MIS.

```
public void addEMPlatformConfigListener(
                              EMPlatformConfigListener listener)
    throws EMTopoServiceException
```

listener is listener of MMC events.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

See Also: `EMPlatformConfigListener`

getLocalSystemName

Returns the name of the MIS that the Topology API is directly connected to.

```
public String getLocalSystemName() throws EMTopoServiceException
```

Returns the local MIS name.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

getSystemNames

Returns all MIS names that are visible through the connection to the local MIS, including the local MIS name. Note that for an MIS to be visible as a remote MIS through the connection to the local MIS, MMC (MIS to MIS Communication) must be setup between the local MIS and each such MIS. This can be accomplished at the user level with the MIS Manager (`em_mismgr`) application.

```
public String[] getSystemNames() throws EMTopoServiceException
```

Returns names of all MISs visible through the connection to the local MIS.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

4.51 `removeEMPlatformConfigListener` Class

Removes the specified listener from the list of objects to be called when MIS to MIS Communication (MMC) is added or removed between the local MIS and another MIS.

```
public void removeEMPlatformConfigListener
                                (EMPlatformConfigListener listener)
    throws EMTopoServiceException
```

listener is listener of MMC events.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

See Also: `EMPlatformConfigListener`

4.52 `EMTopoType` Class

public class `EMTopoType`

extends `EMObject` implements `Cloneable`

An instance of the `com.sun.em.api.topology.EMTopoType` class represents a topology type. Every topology node is classified as a particular topology type. The topology types form a hierarchy with the seven base types Array, Bus, Container, Device, Link, Monitor, and Sun with other subtypes derived from them. Beyond the standard persistent object class (POC) methods which allow you to create, delete and compare topology types, the `EMTopoType` class provides the following additional services:

- static methods `isArray()`, `isBus()`, `isContainer()`, `isDevice()`, `isMonitor()`, `isLink()` to categorize types by their base types.
- static method `isView()` to determine whether topology nodes of the given type are allowed to contain other topology nodes of at least one type.

4.52.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.topology.EMObject
           |
           +----com.sun.em.api.topology.EMTopoType
```

4.52.2 Constructors

EMTopoType

Constructs an instance of `EMTopoType` representing a particular topology type.

```
public EMTopoType(EMTopoPlatform platform, EMTopoTypeDn dn)
```

platform is the Topology API platform.

dn is the unique identifier of the topology type.

4.52.3 Methods

findAllTypes

Returns all topology types located on the local MIS. Note that there is no method to get all the topology types on a remote MIS. A basic assumption is that in a multiple MIS configuration, each MIS will be configured with the exact same set of topology types.

```
public static EMTopoTypeDn[] findAllTypes(EMTopoPlatform platform)
    throws EMTopoServiceException
```

platform is the Topology API platform.

Returns all topology types located on the local MIS.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

findRootTypes

Returns the root topology types located on the local MIS. Some of the root topology types are `EMTopoTypeDn.CONTAINER`, `EMTopoTypeDn.DEVICE`, `EMTopoTypeDn.MONITOR`, `EMTopoTypeDn.LINK`. Additional root types can be added, thus the need for this method. Note that there is no method to get the root topology types on a remote MIS. A basic assumption is that in a multiple MIS configuration, each MIS will be configured with the exact same set of topology types.

```
public static EMTopoTypeDn[]
                          findRootTypes(EMTopoPlatform platform)
    throws EMTopoServiceException
```

platform is the Topology API platform.

Returns the root topology types located on the local MIS.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

isContainer

Returns true if the specified type's `ALL_BASE_TYPES` attribute contains `EMTopoTypeDn.CONTAINER`; otherwise, returns false.

```
public static boolean isContainer
                       (EMTopoPlatform platform, String typeName)
    throws EMTopoServiceException
```

platform is the Topology API platform.

Returns true if the type is a descendent of `EMTopoTypeDn.CONTAINER`.

Throws `EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`NullPointerException` if *typeName* is null.

isDevice

Returns true if the specified type's `ALL_BASE_TYPES` attribute contains *EMTopoTypeDn.DEVICE*; otherwise, returns false.

```
public static boolean isDevice
                      (EMTopoPlatform platform, String typeName)
    throws EMTopoServiceException
```

platform is the Topology API platform.

Returns true if the type is a descendent of `EMTopoTypeDn.DEVICE`.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`NullPointerException` if *typeName* is null.

isLink

Returns true if the specified type's `ALL_BASE_TYPES` attribute contains `EMTopoTypeDn.LINK`; otherwise, returns false.

```
public static boolean isLink(EMTopoPlatform platform,
                             String typeName)
    throws EMTopoServiceException
```

platform is the Topology API platform.

Returns true if the type is a descendent of `EMTopoTypeDn.LINK`.

Throws :

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`NullPointerException` if *typeName* is null.

isMonitor

Returns true if the specified type's ALL_BASE_TYPES attribute contains *EMTopoTypeDn.MONITOR*; otherwise, returns false.

```
public static boolean isMonitor
                      (EMTopoPlatform platform, String typeName)
    throws EMTopoServiceException
```

platform is the Topology API platform.

Returns true if the type is a descendent of EMTopoTypeDn.MONITOR.

Throws:

EMTopoServiceException if there is an internal error in the topology service, or a fault in the communication link to the topology service.

NullPointerException if *typeName* is null.

isArray

Returns true if the specified type's ALL_BASE_TYPES attribute contains EMTopoTypeDn.ARRAY; otherwise, returns false.

```
public static boolean isArray
                      (EMTopoPlatform platform, String typeName)
    throws EMTopoServiceException
```

platform is the Topology API platform.

Returns true if the type is a descendent of EMTopoTypeDn.ARRAY.

Throws:

EMTopoServiceException if there is an internal error in the topology service, or a fault in the communication link to the topology service.

NullPointerException if *typeName* is null.

isBus

Returns true if the specified type's `ALL_BASE_TYPES` attribute contains `EMTopoTypeDn.BUS`; otherwise, returns false.

```
public static boolean isBus
                      (EMTopoPlatform platform, String typeName)
    throws EMTopoServiceException
```

platform is the Topology API platform.

Returns true if the type is a descendent of `EMTopoTypeDn.BUS`.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`NullPointerException` if *typeName* is null.

isView

Returns true if the specified type's `LEGAL_CHILDREN` attribute contains at least one type; otherwise, returns false.

```
public static boolean isView
                      (EMTopoPlatform platform, String typeName)
    throws EMTopoServiceException
```

platform is the Topology API platform.

Returns true if the topology nodes of this type can contain at least one type of topology nodes.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`NullPointerException` if *typeName* is null.

clearAllAttributes

Clears the cached values for all object attributes, resetting them back to their default values (usually null or empty string or 0). This includes the `EMTopoType.DN` attribute which associates the `EMTopoType` instance with a particular topology type object in the MIS. This method is useful when you want to reuse the `EMTopoType` instance to access a different topology type object and do not want the previous values to remain in effect.

```
public void clearAllAttributes()
```

Overrides `clearAllAttributes` in class `EMObject`.

clearSomeAttributes

Clears the cached values for the specified object attributes, resetting them back to their default values (usually null or empty string or 0). This includes the `EMTopoType.DN` attribute which associates the `EMTopoType` instance with a particular topology type object in the MIS. This method is useful when you want to reuse a `EMTopoType` instance to access a different proxy agent object and do not want the previous values to remain in effect.

```
public void clearSomeAttributes(EMAttributeSet attributes)
```

attributes is the object attributes whose cached values will be cleared.

Overrides `clearSomeAttributes` in class `EMObject`.

clone

Return a clone of this `EMTopoType` object.

```
public Object clone()
```

Overrides `clone` in class `Object`.

createWithAllAttributes

Creates a new topology type object in the MIS, storing all active attribute values in the object. Any attribute which was not given a value will take on a default value defined by the GDMO, generally null or empty string. In order for the create to succeed, the following attributes must be set:

`EMTopoAttribute.DN`

`EMTopoTypeAttribute.BASE_TYPE`

`EMTopoTypeAttribute.LAYER_NAME`

The following attributes cannot be set at creation time, and therefore are ignored regardless of whether they have a cached value:

`EMTopoTypeAttribute.ALL_BASE_TYPES`

`EMTopoTypeAttribute.SUB_TYPES`

```
public void createWithAllAttributes()
    throws EMTopoServiceException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if any of the mandatory attributes are not set: `EMTopoAttribute.DN`, `EMTopoTypeAttribute.BASE_TYPE`, and `EMTopoTypeAttribute.LAYER_NAME`.

Overrides `createWithAllAttributes` in class `EMObject`.

createWithSomeAttributes

Creates a new object in the MIS, storing a subset of the active attribute values in the object. Only attributes specified in the parameter attributes will be stored in the new object, and only if the attribute is active, that is it has been given a value. All other attributes will be given a default value defined by the GDMO, generally null or empty string. In order for the create to succeed, the following attributes must be set and must be members of the `EMAttributeSet` attributes:

`EMTopoAttribute.DN`

`EMTopoTypeAttribute.BASE_TYPE`

`EMTopoTypeAttribute.LAYER_NAME`

The following attributes cannot be set at creation time, and therefore are ignored regardless of whether they have a cached value:

`EMTopoTypeAttribute.ALL_BASE_TYPES`

`EMTopoTypeAttribute.SUB_TYPES`

If any of these attributes are members of the `EMAttributeSet` attributes, then an `EMAttributeNotCreatableException` will be thrown.

```
public void createWithSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException
```

attributes is subset of the EMTopoType's attributes to store in the new object.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the any of the mandatory attributes are not set.

`EMAttributeNotCreatableException` if attributes contains any attributes which cannot be set at creation time.

Overrides `createWithSomeAttributes` in class `EMObject`.

destroy

Deletes the object identified by `EMTopoType.DN` from the MIS. This is a permanent, non-reversible operation, so some care should be taken when using this method.

```
public void destroy() throws EMTopoServiceException,
    EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMTopoType.DN` does not exist.

Overrides `destroy` in class `EMObject`.

differences

Compares this EMTopoType against the specified EMTopoType, and returns the set of attributes for which the EMTopoTypes have different values or null if the EMTopoTypes are equal. If the argument **obj** is null or not an instance of EMTopoType, then all active attributes of this EMTopoType are considered to be differences. Otherwise, an attribute has differing values if the attribute is active for one EMTopoType but not the other, or if the attribute is active for both EMTopoTypes but the values of the attribute are not equal.

```
public EMAttributeSet differences(EMObject obj)
```

obj is the object to compare against.

Returns the set of attributes for which the EMTopoTypes have different values or null if the EMTopoTypes are equal.

Overrides differences in class EMObject.

differencesSubset

Compares this EMTopoType against the specified EMTopoType, and returns the set of attributes for which the EMTopoTypes have different values or null if the EMTopoTypes are equal. If the argument *obj* is null or not an instance of EMTopoType, then all active attributes of this EMTopoType are considered to be differences. Otherwise, an attribute has differing values if the attribute is active for one EMTopoType but not the other, or if the attribute is active for both EMTopoTypes but the values of the attribute are not equal.

```
public EMAttributeSet
           differencesSubset(EMObject obj,EMAttributeSet attributes)
```

obj is the object to compare against.

attributes is the set of attributes to compare.

Returns the set of attributes for which the EMTopoTypes have different values or null if the EMTopoTypes are equal.

Overrides differencesSubset in class EMObject.

equals

Compares this EMTopoType against the specified object. If the argument *obj* is null or not an instance of EMTopoType, then the two objects are not equal. Otherwise, if the two EMTopoTypes have the same set of active attributes, and the same value for each active attribute, then the EMTopoTypes are equal.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, otherwise returns false.

Overrides equals in class Object.

equalsSubset

Compares this EMTopoType against the specified object, restricting the equality check to the specified attributes. If the argument *obj* is null or not an instance of EMTopoType, then the two objects are not equal. Otherwise, if the two EMTopoTypes have the same set of active attributes out of the specified attributes, and the same value for each active attribute, then the EMTopoTypes are equal.

```
public boolean equalsSubset(EMObject obj, EMAttributeSet attributes)
```

obj is the object to compare against.

attributes is the subset of attributes.

Returns true if the objects are equal, otherwise returns false.

Overrides equalsSubset in class EMObject.

exists

Checks to see if the object identified by EMTopoType.DN exists.

```
public boolean exists() throws EMTopoServiceException
```

Returns true if the object exists.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

Overrides `exists` in class `EMObject`.

getActiveAttributes

Returns the set of object attributes which have been given a value.

```
public EMAttributeSet getActiveAttributes()
```

Returns the set of active attributes.

Overrides `getActiveAttributes` in class `EMObject`.

loadAllAttributes

Loads all attributes of the object identified by `EMTopoType.DN` from the MIS into the object's attribute cache. These attributes are now considered to be active and can be retrieved with the appropriate getter methods.

```
public void loadAllAttributes() throws EMTopoServiceException,
    EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMTopoType.DN` does not exist.

`EMAttributeDecodeException` if an error occured in converting the attribute values received in ASN.1 format into the local cache data format.

Overrides `loadAllAttributes` in class `EMObject`.

loadSomeAttributes

Loads the specified attributes of the object identified by `EMTopoType.DN` from the MIS into the object's attribute cache. These attributes are now considered to be active and can be retrieved with the appropriate getter methods.

```
public void loadSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException, EMUnknownObjectException
```

attributes is the object attributes to load from the MIS.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMTopoType.DN` does not exist.

`EMAttributeDecodeException` if an error occurred in converting the attribute values received in ASN.1 format into the local cache data format.

Overrides: `loadSomeAttributes` in class `EMObject`.

newInstance

Returns a new instance of the `EMTopoType` class without any attributes set.

```
public EMObject newInstance()
```

Returns the new object.

Overrides `newInstance` in class `EMObject`.

storeAllAttributes

Stores all attributes into the object identified by `EMTopoType.DN`. For attributes that have not been given a value by calling load or setter methods, a default value, usually null or empty string or 0, will be stored. Note that the following `EMTopoType` attributes are read-only and therefore cannot be stored in the MIS:

`EMTopoTypeAttribute.BASE_TYPE`

`EMTopoTypeAttribute.ALL_BASE_TYPES`

`EMTopoTypeAttribute.SUB_TYPES`

These attributes are ignored even if they have cached values.

```
public void storeAllAttributes() throws EMTopoServiceException,
    EMUnknownObjectException
```

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMUnknownObjectException` if the object `EMTopoType.DN` does not exist.

`EMAttributeEncodeException` if an error occurred in converting the attribute values to ASN.1 for transmission to MIS.

Overrides `storeAllAttributes` in class `EMObject`.

storeSomeAttributes

Stores the specified attributes into the object identified by `EMTopoType.DN`. For attributes that have not been given a value by calling load or setter methods, a default value (usually null or empty string or 0) will be stored. Note that the following `EMTopoType` attributes are read-only and therefore cannot be stored in the MIS:

`EMTopoTypeAttribute.BASE_TYPE`

`EMTopoTypeAttribute.ALL_BASE_TYPES`

`EMTopoTypeAttribute.SUB_TYPES`

If any of these attributes are members of the `EMAttributeSet` attributes, then an `EMAttributeNotStoreableException` will be thrown.

```
public void storeSomeAttributes(EMAttributeSet attributes)
    throws EMTopoServiceException, EMUnknownObjectException
```

attributes is the object attributes to store in the MIS.

Throws:

`EMTopoServiceException` if there is an internal error in the topology service, or a fault in the communication link to the topology service.

`EMAttributeNotSetException` if the `EMTopoAttribute.DN` attribute is not set.

`EMAttributeNotStoreableException` if attributes contains any attributes which cannot be set at creation time.

`EMUnknownObjectException` if the object `EMTopoAttribute.DN` does not exist.

`EMAttributeEncodeException` if an error occured in converting the attribute values to ASN.1 for transmission to MIS.

Overrides `storeSomeAttributes` in class `EMObject`.

addLegalChild

Adds child to the `EMTopoTypeAttribute.LEGAL_CHILDREN` attribute.

```
public void addLegalChild(String child)
```

child is the name of the type to add.

Throws:

`EMAttributeNotSetException` if the attribute has not been set in the cache.

`NullPointerException` if *child* is null.

See Also: `LEGAL_CHILDREN`

getAllBaseTypes

Returns the `EMTopoTypeAttribute.ALL_BASE_TYPES` attribute.

```
public String[] getAllBaseTypes()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `ALL_BASE_TYPES`

getBaseType

Returns the `EMTopoTypeAttribute.BASE_TYPE` attribute.

```
public String getBaseType()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `BASE_TYPE`

getDn

Returns the `EMTopoTypeAttribute.DN` attribute.

```
public EMObjectDn getDn()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

Overrides `getDn` in class `EMObject`.

See Also: `DN`

getLayerName

Returns the `EMTopoTypeAttribute.LAYER_NAME` attribute.

```
public String getLayerName()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `LAYER_NAME`

getLegalChildren

Returns the `EMTopoTypeAttribute.LEGAL_CHILDREN` attribute.

```
public String[] getLegalChildren()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `LEGAL_CHILDREN`

getSubTypes

Returns the `EMTopoTypeAttribute.SUB_TYPES` attribute.

```
public String[] getSubTypes()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `SUB_TYPES`

getUserDataAttributeNames

Returns the `EMTopoTypeAttribute.USER_DATA_ATTRIBUTE_NAMES` attribute.

```
public String[] getUserDataAttributeNames()
```

Returns the attribute value.

Throws `EMAttributeNotSetException` if the attribute has not been set in the cache.

See Also: `USER_DATA_ATTRIBUTE_NAMES`

setBaseType

Sets the `EMTopoTypeAttribute.BASE_TYPE` attribute.

```
public void setBaseType(String baseType)
```

baseType is the attribute value.

Throws `NullPointerException` if *baseType* is null.

See Also: `BASE_TYPE`

setDn

Sets the `EMTopoAttribute.DN` attribute.

```
public void setDn(EMObjectDn dn)
```

dn is the attribute value.

Throws:

`NullPointerException` if *dn* is null.

`ClassCastException` if *dn* is not an instance of `EMTopoTypeDn`.

Overrides `setDn` in class `EMObject`.

See Also: `DN`

setLayerName

Sets the `EMTopoTypeAttribute.LAYER_NAME` attribute.

```
public void setLayerName(String layerName)
```

layerName is the attribute value.

Throws `NullPointerException` if *layerName* is null.

See Also: `LAYER_NAME`

setUserDataAttributeNames

Sets the EMTopoTypeAttribute.USER_DATA_ATTRIBUTE_NAMES attribute.

```
public void setUserDataAttributeNames
                                (String userDataAttributeNames[])
```

userDataAttributeNames is the attribute value.

Throws NullPointerException if *userDataAttributeNames* is null.

See Also: USER_DATA_ATTRIBUTE_NAMES

addUserDataAttributeName

Adds the specified userDataAttributeName to the EMTopoTypeAttribute.USER_DATA_ATTRIBUTE_NAMES attribute.

```
public void addUserDataAttributeName(String userDataAttributeName)
```

userDataAttributeName is the attribute value.

See Also: USER_DATA_ATTRIBUTE_NAMES

removeUserDataAttributeName

Removes the specified userDataAttributeName from the EMTopoTypeAttribute.USER_DATA_ATTRIBUTE_NAMES attribute.

```
public void removeUserDataAttributeName
                                    (String userDataAttributeName)
```

userDataAttributeName is the attribute value.

See Also: USER_DATA_ATTRIBUTE_NAMES

toString

Returns a String representation of the EMTopoType.

```
public String toString()
```

Returns the String representation.

Overrides toString in class Object.

4.53 EMTopoTypeAttribute Class

public class EMTopoTypeAttribute

extends EMTopoAttribute implements Serializable

The com.sun.em.api.topology.EMTopoTypeAttribute class represents a single attribute of the EMTopoType persistent object class (POC).

4.53.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMObjectAttribute
           |
           +----com.sun.em.api.topology.EMTopoAttribute
                   |
                   +----com.sun.em.api.topology.EMTopoTypeAttribute
```

4.53.2 Variables

BASE_TYPE

The name of the base type of this type. For example, for a type Device-Router-XyzRouter, the BASE_TYPE would be Router.

```
public static final EMTopoTypeAttribute BASE_TYPE
```

ALL_BASE_TYPES

A list of all the base types of this type. For example, for a type Device-Router-XyzRouter, the list would contain Device, Router.

```
public static final EMTopoTypeAttribute ALL_BASE_TYPES
```

SUB_TYPES

The names of all topology types whose BASE_TYPE is this type. For example, for the type hierarchy:

A-B-C, A-B-D, A-B-C-E

The SUB_TYPES of B is C and D but not E.

```
public static final EMTopoTypeAttribute SUB_TYPES
```

LEGAL_CHILDREN

List of topology types of topology nodes that topology nodes of this type can contain within the topology node graph. If this list is not empty, then `EMTopoType.isView()` will evaluate to true for this type.

```
public static final EMTopoTypeAttribute LEGAL_CHILDREN
```

LAYER_NAME

Name of the layer that includes topology nodes of this type.

```
public static final EMTopoTypeAttribute LAYER_NAME
```

USER_DATA_ATTRIBUTE_NAMES

A list of GDMO attribute names that define the contents of the `EMTopoNode.USER_DATA` attribute for topology nodes of this type.

```
public static final EMTopoTypeAttribute
                                USER_DATA_ATTRIBUTE_NAMES
```

4.53.3 Methods

toString

Returns a String representation of the `EMTopoTypeAttribute`.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `EMTopoAttribute`.

4.54 EMTopoTypeAttributeSet Class

public class `EMTopoTypeAttributeSet`

extends `EMAttributeSet` implements `Cloneable`, `Serializable`

The `com.sun.em.api.topology.EMTopoTypeAttributeSet` class implements an abstract class that forms the basis for the attributes set classes of each `EMTopoType`. `EMTopoTypeAttributeSet` is used in the Topology API to communicate which attributes of a `EMTopoType` an API method should operate on.

4.54.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.common.EMAttributeSet
           |
           +----com.sun.em.api.topology.EMTopoTypeAttributeSet
```

4.54.2 Constructors

EMTopoTypeAttributeSet

Creates an `EMTopoTypeAttributeSet` which contains no attributes.

```
public EMTopoTypeAttributeSet()
```

4.54.3 Methods

clone

Returns a clone of this `EMTopoTypeAttributeSet` object.

```
public Object clone()
```

Overrides `clone` in class `EMAttributeSet` .

elements

Returns an enumeration of the attributes in this set.

```
public Enumeration elements()
```

Overrides *elements* in class `EMAttributeSet`.

4.55 EMTopoTypeDn Class

public final class EMTopoTypeDn

extends EMObjectDn implements Serializable, Comparable

An instance of the com.sun.em.api.topology.EMTopoTypeDn class uniquely identifies a topology type.

4.55.1 Inheritance

```
java.lang.Object
   |
   +----com.sun.em.api.topology.EMObjectDn
           |
           +----com.sun.em.api.topology.EMTopoTypeDn
```

4.55.2 Variables

CONTAINER

Container type. In the out-of-the-box type hierarchy, the Container type is the base type for most of the topology types whose topology nodes can be containers or views of other nodes. However, any topology type can be made to be a container by modifying the EMTopoType.LEGAL_CHILREN attribute.

```
public static final String CONTAINER
```

DEVICE

Device type. In the out-of-the-box type hierarchy, the Device type is the base type for most of the topology types that represent managed devices. In general, these types have an empty attribute, so nodes of these types cannot contain other nodes.

```
public static final String DEVICE
```

LINK

Link type. Topology nodes of type *Link* or of a subtype of *Link* will be displayed as lines drawn between the two topology nodes specified in the node's `LINKS` attribute.

```
public static final String LINK
```

MONITOR

Monitor type. Topology nodes whose type is a subtype of Monitor act as containers of other nodes. They are drawn with special graphics in the Viewer. Instead of a static icon, Monitors are drawn as circles or hexagons whose middles are divided into 1, 3, 6, or 36 sections. Each section represents one topology node that is contained within the Monitor node. The `EMTopoNode` attributes `MONITOR_ROTATION` and `MONITOR_VISIBLE_CHILDREN` control the appearance of the Monitor in the Viewer. Note that no topology nodes should be created with the base type Monitor; instead, use one of the subtypes `OMNISECTOR`, `HEXAGON`, `HEXAGON120`, or `CIRCLE`.

```
public static final String MONITOR
```

OMNISECTOR

OmniSector type, a subclass of *Monitor*. An OmniSector Monitor has a hexagon shape with the whole interior representing a single child topology node.

```
public static final String OMNISECTOR
```

HEXAGON

Hexagon type, a subclass of *Monitor*. The Hexagon Monitor has a hexagon shape with the interior divided into six equal sections with each section representing a child topology node.

```
public static final String HEXAGON
```

HEXAGON120

Hexagon120 type, a subclass of *Monitor*. The Hexagon120 Monitor has a hexagon shape with the interior divided into three equal sections with each section representing a child topology node.

```
public static final String HEXAGON120
```

CIRCLE

Circle type, a subclass of *Monitor*. The Circle Monitor has a circular shape with the interior divided dynamically into as many equal sections *necessary*, up to a maximum of 36, to display each node listed in the node's `MONITOR_VISIBLE_CHILDREN` attribute.

```
public static final String CIRCLE
```

4.55.3 Constructors

EMTopoTypeDn

Creates an `EMTopoTypeDn` with the specified system name and type name.

```
public EMTopoTypeDn(String systemName, String uniqueName)
```

systemName is the name of the MIS where the topology type is stored.

uniqueName is the unique name of the topology type within the MIS.

4.55.4 Methods

equals

Compares this object against the specified object. The result is true, if and only if, the argument is not null and is an `EMTopoTypeDn` object that identifies the same topology type as this object.

```
public boolean equals(Object obj)
```

obj is the object to compare against.

Returns true if the objects are equal, otherwise returns false.

Overrides `equals` in class `Object`.

compareTo

Compares this `EMTopoTypeDn` with another object that implements `Comparable` interface. Order is determined by first lexicographically comparing the system name, and then the unique name if necessary.

```
public int compareTo(Comparable comparable)
```

Returns 0 if the objects are identical, less than 0 if this object is 'less than' the specified object, greater than 0 if this object is 'greater than' the specified object.

Throws `ClassCastException` if the object parameter is not of class `EMTopoTypeDn`.

getSystemName

Returns the name of the MIS where the topology type is stored.

```
public String getSystemName()
```

Returns the MIS name.

Overrides `getSystemName` in class `EMObjectDn`.

getUniqueName

Returns the name of the topology type. This name is unique within the MIS.

```
public String getUniqueName()
```

Returns the unique name of the topology type.

Overrides `getUniqueName` in class `EMObjectDn`.

hashCode

Returns a hashcode for this object.

```
public int hashCode()
```

Returns the hashcode.

Overrides `hashCode` in class `Object`.

toString

Returns a String representation of the `EMTopoTypeDn`.

```
public String toString()
```

Returns the String representation.

Overrides `toString` in class `Object`.

Index

C

E

F

G

J

L

M

O

P

Q

R

S

T