HP Operations Manager i

for the Windows operating system

Software Version: 8.10

Topology Synchronization API Documentation

Document Release Date: 15 January 2010
Software Release Date: June 2009

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

Trademark Notices

Microsoft and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Acknowledgements

This product includes ANTLR.

This product includes software developed by Andy Clark.

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes software developed by Daisuke Okajima and Kohsuke Kawaguchi (http://relaxngcc.sf.net/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software developed by the Indiana University Extreme! Lab (http://www.extreme.indiana.edu/).

This product includes software developed by the JDOM Project (http://www.jdom.org/).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http:// www.openssl.org/).

This product includes software developed by the OpenSymphony Group (http://www.opensymphony.com/).

This product includes code licensed from RSA Data Security.

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online web site at:

www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Introduction

This document is for content developers to use together with the *HP Operations Manager i Extensibility Guide* to customize and extend the topology synchronization functionality of HP Operations Manager i (HP OMi).

This document contains the interfaces needed for developing scripts to perform additional processing and customizing during the topology synchronization process.

Prerequisites

The prerequisites for using this document are as follows:

- You should be familiar with the relevant HP Software products and components, including the associated documentation:
  - HP Operations Manager i (HP OMi). The *HP Operations Manager i Extensibility Guide* is an essential reference for this document.
  - HP Operations Manager (HPOM) for Windows and UNIX.
  - HP Business Availability Center (HP BAC).
  - Universal Configuration Management Database (UCMDB). Detailed administrative and operational knowledge is assumed.
- Knowledge of Groovy scripting and syntax is required for creating Groovy scripts. HP OMi supports Groovy for its scripting capabilities, and uses Groovy scripts in the topology synchronization process.

Package
com.hp.opr.ts.interfaces.data.ci

Representations of intermediate configuration items for internal use in the synchronization process. This package contains the interfaces only. Implementations can be found in the 'impl' sub package.

com.hp.opr.ts.interfaces.data.ci
Class CiMessageCategory

```
java.lang.Object
   |
   +-java.lang.Enum
      |
      +-com.hp.opr.ts.interfaces.data.ci.CiMessageCategory
```

All Implemented Interfaces:
java.io.Serializable, java.lang.Comparable

---

public final class **CiMessageCategory**
extends java.lang.Enum

Categories for an ICiMessage.

Field Summary

public static final	DEFAULT The default category.
public static final	NODE_SYNC Category for messages that occurred during synchronization of the node.
public static final	SERVICE_SYNC Category for messages that occurred during synchronization of the service.

Method Summary

static CiMessageCategory	valueOf(java.lang.String name)
static CiMessageCategory[]	values()

Methods inherited from class java.lang.Enum

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

Methods inherited from class java.lang.Object

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

Methods inherited from interface java.lang.Comparable

compareTo

Fields

(continued on next page)

(continued from last page)

DEFAULT

```
public static final com.hp.opr.ts.interfaces.data.ci.CiMessageCategory DEFAULT
```

The default category.

SERVICE_SYNC

```
public static final com.hp.opr.ts.interfaces.data.ci.CiMessageCategory SERVICE_SYNC
```

Category for messages that occurred during synchronization of the service.

NODE_SYNC

```
public static final com.hp.opr.ts.interfaces.data.ci.CiMessageCategory NODE_SYNC
```

Category for messages that occurred during synchronization of the node.

Methods

values

```
public static CiMessageCategory[] values()
```

valueOf

```
public static CiMessageCategory valueOf(java.lang.String name)
```

com.hp.opr.ts.interfaces.data.ci
Class CiMessageSeverity

```
java.lang.Object
   |
   +-java.lang.Enum
      |
      +-com.hp.opr.ts.interfaces.data.ci.CiMessageSeverity
```

All Implemented Interfaces:
java.io.Serializable, java.lang.Comparable

---

public final class **CiMessageSeverity**
extends java.lang.Enum

Severities of an `ICiMessage`.

Field Summary

public static final	`INFO` Severity for purely informational messages.
public static final	`SEVERE` Severity for severe messages.
public static final	`WARNING` Severity for warning messages.

Method Summary

int	`getOrder()` Get the order number of a severity.
static CiMessageSeverity	`valueOf(java.lang.String name)`
static CiMessageSeverity[]	`values()`

Methods inherited from class `java.lang.Enum`

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

Methods inherited from class `java.lang.Object`

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

Methods inherited from interface `java.lang.Comparable`

`compareTo`

(continued from last page)

Fields

INFO

```
public static final com.hp.opr.ts.interfaces.data.ci.CiMessageSeverity INFO
```

Severity for purely informational messages.

WARNING

```
public static final com.hp.opr.ts.interfaces.data.ci.CiMessageSeverity WARNING
```

Severity for warning messages. The customer should be aware of this message, however there will be no severe problems in case the message is ignored.

SEVERE

```
public static final com.hp.opr.ts.interfaces.data.ci.CiMessageSeverity SEVERE
```

Severity for severe messages. The customer must take this message serious. In case the customer ignores this message it cannot be ensured that a application is working properly.

Methods

values

```
public static CiMessageSeverity[] values()
```

valueOf

```
public static CiMessageSeverity valueOf(java.lang.String name)
```

getOrder

```
public int getOrder()
```

Get the order number of a severity. A higher order number reflects a bigger impact.

Returns:
order number

com.hp.opr.ts.interfaces.data.ci
Interface ICi

All Subinterfaces:
INode

---

public interface **ICi**
extends

The interface to a CI data structure representing the normalized Topology Sync configuration item model.

There are specific properties like ID, the type or the service type definition and arbitrary key/value pairs that can be added as attributes.

Every CI can have containment relations, that is a CI may have 0 to n child CIs and 0 or 1 parent CI. In parallel, every CI may have dependencies to any other CI. These dependencies can have an arbitrary type. See `ICiRelation` for the underlying model of such a dependency.

Method Summary

void	`addCmdbFromRelation(java.lang.String fromOmId, java.lang.String fromCMDBId, java.lang.String relationType)` Adds the CMDB relation from another CI with the specified OM ID.
void	`addCmdbToRelation(java.lang.String toOmId, java.lang.String toCMDBId, java.lang.String relationType)` Adds the CMDB relation to another CI with the specified OM ID.
void	`addMessage(CiMessageSeverity severity, CiMessageCategory category, java.lang.String message)` Add a new message.
void	`addMessage(CiMessageSeverity severity, java.lang.String message)` Add a new message.
void	`addMessage(ICiMessage message)` Add the message.
void	`addOmIdAlias(java.lang.String omId)` Sets an OMId as alias.
ICiRelation	`addOmRelation(ICiRelation relation)` Add a relation from the current CI to the CI contained in the 'to' property.
ICiRelation	`addOmRelation(ICi toCi, java.lang.String relationType)` Add a relation to the specified CI from the current CI using the specified relation type.
void	`dispose()` Dispose this CI.
boolean	`equals(java.lang.Object obj)` Two CIs are considered equal, if the IDs of both CIs are the same.
java.util.Set	`getCmdbAttributeNames()` Get a list of all attributes for the CMDB associated with this CI.

java.util.Map	getCmdbAttributes() Get a map containing all the attributes for the CMDB contained in this CI.
java.lang.Object	getCmdbAttributeValue(java.lang.String name) Get the value of the given attribute for the CMDB of the CI.
java.util.Collection	getCmdbFromRelations() Gets the CMDB relations from other CIs.
java.lang.String	getCmdbId() Gets the uCMDB id.
java.util.Collection	getCmdbToRelations() Gets the CMDB relations to other CIs.
java.lang.String	getCmdbTypeId() Get the ID of this CI's type.
java.util.Collection	getContexts() Gets the collection of assigned contexts.
java.lang.String	getHostedOn() Gets the node name where the CI is hosted on
java.lang.String	getHostedOnOmId() Gets the node OMId where the CI is hosted on
java.util.List	getMessages() Gets all messages attached to this CI in the same order as the messages have been added.
java.util.List	getMessages(CiMessageCategory category) Gets all messages of the given category attached to this CI in the same order as the messages have been added.
java.util.List	getMessages(CiMessageSeverity severity) Gets all messages that are attached to this CI and that are of equal or higher severity level of the severity specified as parameter.
java.util.List	getMessages(CiMessageSeverity severity, CiMessageCategory category) Gets all messages that are attached to this CI, that are assigned to the given category and that are of equal or higher severity level of the severity specified in the first parameter.
INavigator	getNavigator() Gets an INavigator instance that uses this CI as context CI.
java.util.Set	getOmAttributeNames() Get a list of all attributes from OM associated with this CI.
java.util.Map	getOmAttributes() Get a map containing all the attributes from OM contained in this CI.
java.lang.Object	getOmAttributeValue(java.lang.String name) Get the value of the given attribute from OM of the CI.
java.lang.String	getOmCaption() Returns the caption of the CI, which is the value of the attribute display_label.
java.util.Set	getOmChildCis() Get a set of all child CIs, to which the current CI has a 'containment' relationship.

java.util.Set	getOmChildRelations() Retrieve a set of CI relations that points to CIs that are contained by the current CI.
java.util.Set	getOmDependencyRelations() Gets the dependencies to other CIs.
java.util.Collection	getOmDependentRelations() Gets all relations from other CIs that have a dependency to this CI.
java.lang.String	getOmId() Get the ID of the CI in OM.
java.util.Collection	getOmIdAliases() Gets the collection of all OmId aliases assigned to this CI.
ICi	getOmParentCi() Get the parent item of this CI.
ICiRelation	getOmParentRelation() Get the CI relation from the parent CI to the current CI.
ICi	getOmRootCi() Get the root CI of the hierarchy in which the current CI is contained.
java.lang.String	getOmTypeId() Get the service type definition ID of this CI.
java.lang.String	getRootContainerOmId() Get the root container OM Id of this CI.
boolean	hasContext(java.lang.String context) Checks if the given context is assigned to this CI.
boolean	hasContexts() Checks for assigned contexts.
int	hashCode() An implementation of an ICi must override the hashCode() to match the implementation of an ICis equals(Object) method.
boolean	hasMessages() Check if there are any messages attached to this CI.
boolean	hasMessages(CiMessageSeverity severity) Check if there are any messages attached to this CI with a severity that is equal to or higher than the given severity.
boolean	isChildOf(ICi parent) Checks if the current CI is a child of the given CI.
boolean	isExternalStub() Checks if the CI is an external stub.
boolean	isNode() Checks if this CI is a node.
boolean	isParentOf(ICi child) Checks if the current CI a parent of the given CI.

boolean	isRootCi() Checks if this CI is the root CI.
boolean	isService() Checks if this CI will be imported as service.
java.lang.String	printCi() Print the short description of the CI.
boolean	removeRelation(ICiRelation relation) Remove the given relation.
boolean	removeRelation(java.lang.String toCiId) Remove the relation to the CI with the given ID.
void	setCaption(java.lang.String caption) Set the caption of the CI.
void	setCmdbAttribute(java.lang.String attribute, java.lang.Object value) Sets the CMDB attribute and its value to the CMDB attributes.
void	setCmdbId(java.lang.String cmdbId) Sets the uCMDB id.
void	setCmdbTypeId(java.lang.String type) Set the ID of the CMDB CI type.
void	setContext(java.lang.String context) Assigns this CI to the given context.
void	setHostedOnOmId(java.lang.String hostedOnOmId)
void	setIsNode(boolean isNode) Specify whether this CI is a Node.
void	setIsService(boolean isService) Specify whether this CI is a Service.
void	setOmAttribute(java.lang.String attribute, java.lang.Object value) Sets the attribute and its value to the OM attributes.
void	setOmId(java.lang.String id) Set the ID of the OM Service.
void	setOmTypeId(java.lang.String serviceTypeDefinitionID) Set the service type definition of this CI.
void	setRootContainerOmId(java.lang.String rootContainerOmId) Sets the root container OM id.

Methods

setOmAttribute

```
public void setOmAttribute(java.lang.String attribute,
        java.lang.Object value)
```

(continued from last page)

Sets the attribute and its value to the OM attributes.

Parameters:

`attribute` - name of the attribute
`value` - value of the attribute

---

setCmdbAttribute

```
public void setCmdbAttribute(java.lang.String attribute,
        java.lang.Object value)
```

Sets the CMDB attribute and its value to the CMDB attributes.

Parameters:

`attribute` - name of the attribute
`value` - the value of the attribute

---

addMessage

```
public void addMessage(CiMessageSeverity severity,
        CiMessageCategory category,
        java.lang.String message)
```

Add a new message.

The CI message describes an event that has occurred during processing of this CI.

Parameters:

`severity` - the severity
`message` - the message
`category` - the category

---

addMessage

```
public void addMessage(CiMessageSeverity severity,
        java.lang.String message)
```

Add a new message.

The CI message describes an event that has occurred during processing of this CI.

Parameters:

`severity` - the severity
`message` - the message

---

addMessage

```
public void addMessage(ICiMessage message)
```

Add the message.

The CI message describes an event that has occurred during processing of this CI.

Parameters:

`message` - the message

---

addOmRelation

```
public ICiRelation addOmRelation(ICi toCi,
        java.lang.String relationType)
```

(continued from last page)

Add a relation to the specified CI from the current CI using the specified relation type. According to the relation type, the relation will become a containment relationship (parent/child) or a dependency.

If it starts with "container_" then it will be added to the containments.

Parameters:

`toCi` - the CI

`relationType` - the relation type

Returns:

the `ICiRelation` instance that has been created during this operation.

---

addOmRelation

`public ICiRelation addOmRelation(ICiRelation relation)`

Add a relation from the current CI to the CI contained in the 'to' property. According to the type, the relation will become a containment relationship (parent/child) or a dependency.

If it starts with "container_" then it will be added to the containments.

Note that the 'from' property of the relation must be the CI on which you call this method.

Parameters:

`relation` - the relation to the other CI.

Returns:

the added CI Relation

---

dispose

`public void dispose()`

Dispose this CI. All references to other objects will be set to null, so that this CI can safely be disposed by the Garbage Collector.

After this method has been called make sure that the CI will no longer be used.

---

equals

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

Two CIs are considered equal, if the IDs of both CIs are the same.

Parameters:

`obj` - the object that represents another CI.

Returns:

true, if the ID of both CIs are equal.

See Also:

`Object.equals(java.lang.Object)`

---

getOmAttributeNames

`public java.util.Set getOmAttributeNames()`

Get a list of all attributes from OM associated with this CI.

Returns:

(continued from last page)

Set of all attributes associated with this Ci.

getOmAttributes

`public java.util.Map getOmAttributes()`

Get a map containing all the attributes from OM contained in this CI. The key of the map is the name of the attribute.

Returns:
Map of attributes.

getOmAttributeValue

`public java.lang.Object getOmAttributeValue(java.lang.String name)`

Get the value of the given attribute from OM of the CI.

Parameters:
`name` - Name of the attribute.

Returns:
Value of the attribute.

getCmdbAttributeNames

`public java.util.Set getCmdbAttributeNames()`

Get a list of all attributes for the CMDB associated with this CI.

Returns:
Set of all attributes associated with this Ci.

getCmdbAttributes

`public java.util.Map getCmdbAttributes()`

Get a map containing all the attributes for the CMDB contained in this CI. The key of the map is the name of the attribute.

Returns:
Map of attributes.

getCmdbAttributeValue

`public java.lang.Object getCmdbAttributeValue(java.lang.String name)`

Get the value of the given attribute for the CMDB of the CI.

Parameters:
`name` - Name of the attribute.

Returns:
Value of the attribute.

getOmCaption

`public java.lang.String getOmCaption()`

Returns the caption of the CI, which is the value of the attribute `display_label`.

(continued from last page)

Returns:
Caption.

See Also:
`setCaption(String)`

getOmChildCis

`public java.util.Set getOmChildCis()`

Get a set of all child CIs, to which the current CI has a 'containment' relationship. This method returns the CIs that are referenced by the containment relationships.

The returned set is independent of the real relations. Adding or removing CIs from this set will have no effect on the current child relations of this CI. This method is just a convenience method for easier traversion of the CI's topology hierarchy.

Returns:
Set of child CIs.

getOmChildRelations

`public java.util.Set getOmChildRelations()`

Retrieve a set of CI relations that points to CIs that are contained by the current CI. A CI is contained by this CI if the relation type was identified as a containment relation when it was added using the method `addOmRelation(ICi, String)`.

Do not add to or remove relations directly from the returned list.

To add a new relation use the method `addOmRelation(ICiRelation)`

Using the `remove(ICiRelation)` method on the returned list will remove the relation from this CI to the child CI, but it will not delete the reverse parent relation from the child CI to this CI. Instead, use either the method `removeRelation(ICiRelation)` or `removeRelation(String)`.

Returns:
List of `ICiRelation` instances.

See Also:
`addOmRelation(ICiRelation)`
`addOmRelation(ICi, String)`
`removeRelation(ICiRelation)`
`removeRelation(String)`

getOmDependencyRelations

`public java.util.Set getOmDependencyRelations()`

Gets the dependencies to other CIs.

Do not add or remove relations from the returned collection. Always use the methods

- `addOmRelation(ICiRelation)` to add a new relation.
- `removeRelation(ICiRelation)` to remove a dependency to ensure the integrity of the data model.

Returns:
the dependencies to other CIs.

(continued from last page)

See Also:

`addOmRelation(ICiRelation)`
`addOmRelation(ICi, String)`
`removeRelation(ICiRelation)`
`removeRelation(String)`

getOmId

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

Get the ID of the CI in OM.

Returns:

ID of the CI.

See Also:

`setOmId(String)`

getMessages

```
public java.util.List getMessages()
```

Gets all messages attached to this CI in the same order as the messages have been added.

The CI message describes an event that has occurred during processing of this CI.

Returns:

the messages

getMessages

```
public java.util.List getMessages(CiMessageCategory category)
```

Gets all messages of the given category attached to this CI in the same order as the messages have been added.

The CI message describes an event that has occurred during processing of this CI.

Parameters:

`category` - the category

Returns:

the messages

getMessages

```
public java.util.List getMessages(CiMessageSeverity severity)
```

Gets all messages that are attached to this CI and that are of equal or higher severity level of the severity specified as parameter. The order of the messages is the order in which the messages were added to this CI.

The CI message describes an event that has occurred during processing of this CI.

Parameters:

`severity` - the severity

Returns:

the messages

(continued from last page)

getMessages

```
public java.util.List getMessages(CiMessageSeverity severity,
        CiMessageCategory category)
```

Gets all messages that are attached to this CI, that are assigned to the given category and that are of equal or higher severity level of the severity specified in the first parameter. The order of the messages is the order in which the messages were added to this CI.

The CI message describes an event that has occurred during processing of this CI.

Parameters:

`severity` - the severity. If set to `null` no filtering will be done by severity.
`category` - the category. If set to `null` no filtering will be done by category.

Returns:

the messages

---

getNavigator

```
public INavigator getNavigator()
```

Gets an `INavigator` instance that uses this CI as context CI.

Returns:

the navigator

---

getOmParentCi

```
public ICi getOmParentCi()
```

Get the parent item of this CI. This means that the returned CI has a 'contains' association to the current CI.

Returns `null` if there is no parent CI and the current CI is an orphan. In that case the current CI will be located below the configured root service in the HP OM Service Navigator.

Returns:

Parent CI. May return `null`.

See Also:

`ICiRelation.getType()`

---

getOmParentRelation

```
public ICiRelation getOmParentRelation()
```

Get the CI relation from the parent CI to the current CI. A parent CI is the CI that has a 1:n containment relationship to the current CI.

The function returns `null` when the current CI is the topmost CI in the hierarchical topology, i.e. the function `isRootCi()` returns `true`.

Returns:

CI relation from the parent to this CI. `null`, if `isRootCi()` returns `true`.

---

getOmRootCi

```
public ICi getOmRootCi()
```

Get the root CI of the hierarchy in which the current CI is contained.

(continued from last page)

Returns:
the root CI.

getOmTypeId

`public java.lang.String getOmTypeId()`

Get the service type definition ID of this CI.

Returns:
Service type definition ID

See Also:
`setOmTypeId(String)`

getCmdbTypeId

`public java.lang.String getCmdbTypeId()`

Get the ID of this CI's type.

Returns:
CI type ID

See Also:
`setCmdbTypeId(String)`

hashCode

`public int hashCode()`

An implementation of an `ICi` must override the `hashCode()` to match the implementation of an ICis `equals(Object)` method.

Returns:
the hash code

See Also:
`Object.hashCode()`

hasMessages

`public boolean hasMessages()`

Check if there are any messages attached to this CI.

Returns:
`true`, if there is at least one message attached to this CI.

hasMessages

`public boolean hasMessages(CiMessageSeverity severity)`

Check if there are any messages attached to this CI with a severity that is equal to or higher than the given severity.

Parameters:
`severity` - Only check for messages that have this or a higher severity.

Returns:

(continued from last page)

`true`, if there is at least one message attached to this CI.

isChildOf

```
public boolean isChildOf(ICi parent)
```

Checks if the current CI is a child of the given CI.

Parameters:
`parent` - Suspected parent CI.

Returns:
`true`, if the suspected parent is a direct parent of the current CI.

isNode

```
public boolean isNode()
```

Checks if this CI is a node.

Returns:
`true`, if the CI is a node.

isParentOf

```
public boolean isParentOf(ICi child)
```

Checks if the current CI a parent of the given CI.

Parameters:
`child` - Suspected child CI.

Returns:
`true`, if the suspected child is a direct child of the current CI.

isRootCi

```
public boolean isRootCi()
```

Checks if this CI is the root CI.

Returns:
true, if this is the root CI.

isService

```
public boolean isService()
```

Checks if this CI will be imported as service.

Returns:
`true`, if the CI will be imported as service.

printCi

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

Print the short description of the CI.

(continued from last page)

Returns:
short description of the CI.

removeRelation

`public boolean removeRelation(ICiRelation relation)`

Remove the given relation. Returns `true`, if the relation has been found and it was successfully removed.

Parameters:
`relation` - the relation to a child CI.

Returns:
`true`, if successful.

removeRelation

`public boolean removeRelation(java.lang.String toCiId)`

Remove the relation to the CI with the given ID. Returns true, if the relation has been found and it was successfully removed.

Parameters:
`toCiId` - the ID of the CI to which the relation should be removed.

Returns:
`true`, if successful

setCaption

`public void setCaption(java.lang.String caption)`

Set the caption of the CI. This method sets the attribute `display_label`.

Parameters:
`caption` - the caption

See Also:
`getOmCaption()`

setOmId

`public void setOmId(java.lang.String id)`

Set the ID of the OM Service. The ID must be unique across all CIs. In general, the ID should not be changed after the CI has been initialized with all values, so use this method with care.

Parameters:
`id` - ID of the CI.

See Also:
`getOmId()`

setIsNode

`public void setIsNode(boolean isNode)`

Specify whether this CI is a Node.

(continued from last page)

Parameters:
`isNode` - Set to `null`, if the current CI is a Node.

See Also:
`isNode()`

setIsService

`public void setIsService(boolean isService)`

Specify whether this CI is a Service.

Parameters:
`isService` - Set to `null`, if the current CI is a Service.

See Also:
`isService()`

setOmTypeId

`public void setOmTypeId(java.lang.String serviceTypeDefinitionID)`

Set the service type definition of this CI.

Parameters:
`serviceTypeDefinitionID` - The service type definition ID.

See Also:
`getOmTypeId()`

setCmdbTypeId

`public void setCmdbTypeId(java.lang.String type)`

Set the ID of the CMDB CI type.

Parameters:
`type` - The CI type ID.

See Also:
`getCmdbTypeId()`

isExternalStub

`public boolean isExternalStub()`

Checks if the CI is an external stub. An external CI stub is not part of the synchronization data. It's only purpose is to serve as a representation of a CI that already exists in HP Operations Manager to which a CI relation can be defined.

Returns:
true, if the CI is external

getOmDependentRelations

`public java.util.Collection getOmDependentRelations()`

(continued from last page)

Gets all relations from other CIs that have a dependency to this CI. Relations get added to this collection, when the method `addOmRelation(ICiRelation)` is called on the dependent CI with a relation to this CI, where the relation type is not a containment type.

Do not remove items from the returned collection as this may harm the data model integrity. Always use the method `removeRelation(ICiRelation)` on the dependent CI.

Returns:
the dependent relations

See Also:
`removeRelation(ICiRelation)`
`removeRelation(String)`

---

getRootContainerOmId

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

Get the root container OM Id of this CI.

Returns `null` if there is no root container CI

Returns:
RootContainer CI. May return `null`.

---

setRootContainerOmId

```
public void setRootContainerOmId(java.lang.String rootContainerOmId)
```

Sets the root container OM id.

Parameters:
`rootContainerOmId` - the new root container service id

---

setCmdbId

```
public void setCmdbId(java.lang.String cmdbId)
```

Sets the uCMDB id.

Parameters:
`cmdbId` - the new uCMDB id

---

getCmdbId

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

Gets the uCMDB id.

Returns:
the uCMDB id

---

addCmdbToRelation

```
public void addCmdbToRelation(java.lang.String toOmId,
          java.lang.String toCMDBId,
          java.lang.String relationType)
```

Adds the CMDB relation to another CI with the specified OM ID.

(continued from last page)

Parameters:

`toOmId` - the OM ID of the target CI.
`toCMDBId` - the CMDB ID of the target CI.
`relationType` - the relation type

addCmdbFromRelation

```
public void addCmdbFromRelation(java.lang.String fromOmId,
         java.lang.String fromCMDBId,
         java.lang.String relationType)
```

Adds the CMDB relation from another CI with the specified OM ID.

Parameters:

`fromOmId` - the OM ID of the source CI.
`fromCMDBId` - the CMDB ID of the source CI.
`relationType` - the relation type

getCmdbToRelations

```
public java.util.Collection getCmdbToRelations()
```

Gets the CMDB relations to other CIs.

Returns:

the relations.

getCmdbFromRelations

```
public java.util.Collection getCmdbFromRelations()
```

Gets the CMDB relations from other CIs.

Returns:

the relations.

hasContexts

```
public boolean hasContexts()
```

Checks for assigned contexts.

A context is internally used for assigning a CI to certain mapping rules and filtering CIs.

Returns:

true, if at least one context has been assigned.

hasContext

```
public boolean hasContext(java.lang.String context)
```

Checks if the given context is assigned to this CI.

A context is internally used for assigning a CI to certain mapping rules and filtering CIs.

Parameters:

`context` - the context

Returns:

(continued from last page)

true, if successful

setContext

`public void setContext(java.lang.String context)`

Assigns this CI to the given context.

A context is internally used for assigning a CI to certain mapping rules and filtering CIs.

Parameters:

`context` - the new context

getContexts

`public java.util.Collection getContexts()`

Gets the collection of assigned contexts.

A context is internally used for assigning a CI to certain mapping rules and filtering CIs.

Returns:

collection of contexts

addOmIdAlias

`public void addOmIdAlias(java.lang.String omId)`

Sets an OMId as alias.

The aliases are written by the CMDB adapter to the ID mapping table in order to redirect messages from the alias OMId to the current CI in the CMDB

Parameters:

`omId` - the alias OMId

getOmIdAliases

`public java.util.Collection getOmIdAliases()`

Gets the collection of all OmId aliases assigned to this CI.

Returns:

collection of OmId aliases

getHostedOn

`public java.lang.String getHostedOn()`

Gets the node name where the CI is hosted on

Returns:

the nodename where the CI is hosted on

getHostedOnOmId

`public java.lang.String getHostedOnOmId()`

Gets the node OMId where the CI is hosted on

(continued from last page)

Returns:

the OMId where the CI is hosted on

setHostedOnOmId

```
public void setHostedOnOmId(java.lang.String hostedOnOmId)
```

Parameters:

`hostedOnOmId` - the hostedOnOmId to set

com.hp.opr.ts.interfaces.data.ci
Interface ICiMessage

public interface **ICiMessage**
extends

A CI message is a message that can be attached to a CI to describe status information of events that occurred during processing of that CI. A CI message has a severity, a category and the message string as properties.
See Also:
`CiMessageSeverity`, `CiMessageSeverity`

Method Summary

`CiMessageCategory`	`getCategory()` Gets the category of the CI message.
`java.lang.String`	`getMessage()` Gets the message of the CI message.
`CiMessageSeverity`	`getSeverity()` Gets the severity of the CI message.
`void`	`setCategory(CiMessageCategory category)` Sets the category of the CI message.
`void`	`setMessage(java.lang.String message)` Sets the message of the CI message.
`void`	`setSeverity(CiMessageSeverity severity)` Sets the severity of the CI message.

Methods

getCategory

`public CiMessageCategory getCategory()`

Gets the category of the CI message.

Returns:
the category

setCategory

`public void setCategory(CiMessageCategory category)`

Sets the category of the CI message.

Parameters:
`category` - the new category

(continued from last page)

getSeverity

`public CiMessageSeverity getSeverity()`

Gets the severity of the CI message.

Returns:
the severity

---

setSeverity

`public void setSeverity(CiMessageSeverity severity)`

Sets the severity of the CI message.

Parameters:
`severity` - the new severity

---

getMessage

`public java.lang.String getMessage()`

Gets the message of the CI message.

Returns:
the message

---

setMessage

`public void setMessage(java.lang.String message)`

Sets the message of the CI message.

Parameters:
`message` - the new message

com.hp.opr.ts.interfaces.data.ci
Interface ICiRelation

All Superinterfaces:
IRelation

public interface **ICiRelation**
extends IRelation

An `ICiRelation` models the relation from one CI instance to another.

In contrast to `IRelation` this relation type handles concrete instances of `ICi`. This relation type is used to model the dependencies between items of the OM model and will be used by the mapping engine to follow those relations from one ICi instance to another.

Method Summary

void	dispose() Dispose this CI relation.
ICi	getFrom() Get the CI from which this relation origins.
ICi	getTo() Get the CI to which this relation points to.
java.lang.String	getType() Returns the type of a CI relation
boolean	isContainmentRelation() Checks if this relation defines a containment relation.
boolean	isExternal() Checks if this relation exists between an external CI stub and a CI.

Methods inherited from interface com.hp.opr.ts.interfaces.data.ci.IRelation

getFromCMDBId, getFromId, getToCMDBId, getToId, getType, setFromCMDBId, setToCMDBId

Methods

dispose

`public void dispose()`

Dispose this CI relation. All references to other objects will be set to null, so that this CI relation can safely be disposed by the Garbage Collector. The references to this CI will be removed from the linked CIs. After disposing this relation, the CIs will have no relations to each other any more.

getFrom

`public ICi getFrom()`

Get the CI from which this relation origins.

(continued from last page)

Returns:
Source CI

getTo

`public ICi getTo()`

Get the CI to which this relation points to.

Returns:
Target CI

isContainmentRelation

`public boolean isContainmentRelation()`

Checks if this relation defines a containment relation. A containment relation is a parent-child relation. When added to a CI, the relation will be stored in the children collection.

The decision, whether this relation is a containment relation depends on the type that is set when the relation is created.

Returns:
true, if it is a containment relation

isExternal

`public boolean isExternal()`

Checks if this relation exists between an external CI stub and a CI.

Returns:
true, if it is an external relation

getType

`public java.lang.String getType()`

Returns the type of a CI relation

Returns:
type of the relation

com.hp.opr.ts.interfaces.data.ci
Interface INavigator

public interface **INavigator**
extends

A navigator allows to navigate through a CI topology tree. All results are returned relative to a context CI. When you use the method `ICi.getNavigator()` for instance, the CI on which you execute this method will become the context CI of the returned `INavigator` instance.

A navigator allows to return all ancestors and all descendants of the context CI. When using XPath expressions, you can directly access values of any CI that is contained in the same hierarchy as the context CI. For more information on XPath expressions, refer to the HP Operations Manager Dependency Mapping Automation Developer Guide.

Method Summary

ICi	getAncestorCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator) Gets the first ancestor CI that causes the specified operator to return true.
ICi	getAncestorCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator, java.lang.String relationType) Gets the first ancestor CI that causes the specified operator to return true.
java.util.List	getAncestors() Get all ancestors of the context CI.
java.util.List	getAncestors(java.lang.String cmdbType) Get all ancestors of the context CI that are of the given CMDB type.
ICi	getChild(java.lang.String cmdbType) Get the first child of the context CI that is of the given CMDB type.
ICi	getChildCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator) Gets the first child CI that causes the specified operator to return true.
ICi	getChildCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator, java.lang.String relationType) Gets the first child CI that causes the specified operator to return true.
java.util.List	getChildCiList(com.hp.opr.ts.interfaces.enrichment.IOperator operator) Gets the list of child CIs that cause the specified operator to return true.
java.util.List	getChildCiList(com.hp.opr.ts.interfaces.enrichment.IOperator operator, java.lang.String relationType) Gets the list of child CIs that cause the specified operator to return true.
java.util.List	getChildren(java.lang.String cmdbType) Get all children of the context CI that are of the given CMDB type.
ICi	getCiValue(java.lang.String xpath) Get a CI value from a Java property using an XPath expression starting at the CI given in the constructor of this CI navigator instance.
java.util.List	getCiValues(java.lang.String xpath) Select all CIs matching the given XPath expression starting at the CI given in the constructor of this CI navigator instance.

ICi	getDependencyCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator) Gets the first dependency CI that causes the specified operator to return true.
ICi	getDependencyCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator, java.lang.String relationType) Gets the first dependency CI that causes the specified operator to return true.
java.util.List	getDependencyCiList(com.hp.opr.ts.interfaces.enrichment.IOperator operator) Gets the list of dependency CIs that cause the specified operator to return true.
java.util.List	getDependencyCiList(com.hp.opr.ts.interfaces.enrichment.IOperator operator, java.lang.String relationType) Gets the list of dependency CIs that cause the specified operator to return true.
ICi	getDependentCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator) Gets the first dependent CI that causes the specified operator to return true.
ICi	getDependentCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator, java.lang.String relationType) Gets the first dependent CI that causes the specified operator to return true.
java.util.List	getDependentCiList(com.hp.opr.ts.interfaces.enrichment.IOperator operator) Gets the list of dependent CIs that cause the specified operator to return true.
java.util.List	getDependentCiList(com.hp.opr.ts.interfaces.enrichment.IOperator operator, java.lang.String relationType) Gets the list of dependent CIs that cause the specified operator to return true.
ICi	getDescendant(java.lang.String cmdbType) Get a descendant that is of the given CMDB type.
ICi	getDescendantCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator) Gets the first descendant CI that causes the specified operator to return true.
ICi	getDescendantCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator, java.lang.String relationType) Gets the first descendant CI that causes the specified operator to return true.
java.util.List	getDescendantCiList(com.hp.opr.ts.interfaces.enrichment.IOperator operator) Gets the list of descendant CIs that cause the specified operator to return true.
java.util.List	getDescendantCiList(com.hp.opr.ts.interfaces.enrichment.IOperator operator, java.lang.String relationType) Gets the list of descendant CIs that cause the specified operator to return true.
java.util.List	getDescendants() Get all descendants of the context CI.
java.util.List	getDescendants(java.lang.String cmdbType) Get all descendants that are of the given CMDB type.
java.lang.String	getStringValue(java.lang.String xpath) Get the String value of a Java property using an XPath expression using this CI as context.
java.util.List	getStringValueList(java.lang.String xpath) Select all properties matching the given XPath expression starting at the CI given in the constructor of this CI navigator instance.

java.lang.Object	getValue(java.lang.String xpath) Get the value of a Java property using an XPath expression having this CI as context.
java.util.List	getValues(java.lang.String xpath) Select all properties matching the given XPath expression using this CI as context.

Methods

getAncestors

```
public java.util.List getAncestors()
```

Get all ancestors of the context CI.

An ancestor is any CI that can be found in the parent hierarchy, i.e. the direct parent, the grandfather, the great-grandfather and so on.

Returns:
Ancestor CIs of the context CI.

getAncestors

```
public java.util.List getAncestors(java.lang.String cmdbType)
```

Get all ancestors of the context CI that are of the given CMDB type.

An ancestor is any CI that can be found in the parent hierarchy, i.e. the direct parent, the grandfather, the great-grandfather and so on.

Parameters:
`cmdbType` - The CMDB type of the ancestor CI to return.

Returns:
Ancestor CIs of the context CI that are of the given type.

getChild

```
public ICi getChild(java.lang.String cmdbType)
```

Get the first child of the context CI that is of the given CMDB type.

Parameters:
`cmdbType` - The CMDB type of the child to return.

Returns:
The first child that has the given type.

getChildren

```
public java.util.List getChildren(java.lang.String cmdbType)
```

Get all children of the context CI that are of the given CMDB type.

Parameters:
`cmdbType` - The CMDB type of the children to return.

(continued from last page)

Returns:
List of children that have the given type.

getCiValue

`public ICi getCiValue(java.lang.String xpath)`

Get a CI value from a Java property using an XPath expression starting at the CI given in the constructor of this CI navigator instance.

See `getValue(String)` for more information and examples.

Parameters:
`xpath` - XPath expression pointing to the String property

Returns:
Property as `ICi`. Returns `null`, if the path cannot be resolved due to wrong properties or syntax errors or if the returned type is not an instance of an `ICi`.

See Also:
`getValue(String)`

getCiValues

`public java.util.List getCiValues(java.lang.String xpath)`

Select all CIs matching the given XPath expression starting at the CI given in the constructor of this CI navigator instance.

See `getValue(String)` for more information and examples.

Parameters:
`xpath` - The XPath expression that selects the desired properties.

Returns:
List of `ICi` instances. `null`, if the xpath cannot be resolved due to wrong properties or syntax errors.

See Also:
`getValue(String)`

getDescendant

`public ICi getDescendant(java.lang.String cmdbType)`

Get a descendant that is of the given CMDB type.

A descendant is any CI that can be found in the hierarchy of children, i.e. direct children, grandchildren, great-grandchildren and so on.

Parameters:
`cmdbType` - The CMDB type of the descendant to return.

Returns:
Descendant CI of the given type.

getDescendants

`public java.util.List getDescendants()`

(continued from last page)

Get all descendants of the context CI.

A descendant is any CI that can be found in the hierarchy of this CI's children, i.e. direct children, grandchildren, great-grandchildren and so on.

Returns:
List of descendant CIs.

---

getDescendants

```
public java.util.List getDescendants(java.lang.String cmdbType)
```

Get all descendants that are of the given CMDB type.

A descendant is any CI that can be found in the hierarchy of children, i.e. direct children, grandchildren, great-grand children and so on.

Parameters:
`cmdbType` - The CMDB type of the descendants to return.

Returns:
List of descendant CIs of the given type.

---

getValue

```
public java.lang.Object getValue(java.lang.String xpath)
```

Get the value of a Java property using an XPath expression having this CI as context.

Example:

- The XPath expression /attributes/display_label returns the value of the CMDB attribute 'display_label', which would be the same as the XPath expression /label.
- //children[type='os']/attributes/display_label would return the display label of a child or a child of a child (and so on) that has the CMDB type 'os'.

See the JXPath users guide for more information on XPath expressions for Java object instances.

Parameters:
`xpath` - The XPath expression pointing to the bean property.

Returns:
A property object. null if the path cannot be resolved due to wrong properties or syntax errors.

See Also:
JXPath User's Guide
Java object queries using JXPath

---

getValues

```
public java.util.List getValues(java.lang.String xpath)
```

Select all properties matching the given XPath expression using this CI as context.

See `getValue(String)` for more information and examples.

(continued from last page)

Parameters:
`xpath` - XPath expression that selects the desired properties.

Returns:
List of properties. null if the XPath expression cannot be resolved due to wrong properties or syntax errors.

See Also:
`getValue(String)`

---

getAncestorCi

```
public ICi getAncestorCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator)
```

Gets the first ancestor CI that causes the specified operator to return `true`.

Returns `null` if no matching CI is found.

Parameters:
`operator` - the operator

Returns:
the found ancestor CI. `null`, if no matching CI is found.

---

getAncestorCi

```
public ICi getAncestorCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator,
    java.lang.String relationType)
```

Gets the first ancestor CI that causes the specified operator to return `true`. Only relations of the specified relation type will be followed. The algorithm stops following relations at the first relation that is not of the specified relation type.

Returns `null` if no matching CI is found.

Parameters:
`operator` - the operator
`relationType` - the relation type

Returns:
the found ancestor CI. `null`, if no matching CI is found.

---

getDescendantCi

```
public ICi getDescendantCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator)
```

Gets the first descendant CI that causes the specified operator to return `true`. Only relations of the specified relation type will be followed.

Descendants will be searched level for level to make sure that the found CI is contained in the closest possible level.

Returns `null` if no matching CI is found.

Parameters:
`operator` - the operator

Returns:
the found descendant CI. `null`, if no matching CI is found.

(continued from last page)

getDescendantCi

```
public ICi getDescendantCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator,
        java.lang.String relationType)
```

Gets the first descendant CI that causes the specified operator to return `true`. Only relations of the specified relation type will be followed. The algorithm stops following relations at the first relation that is not of the specified relation type.

Descendants will be searched level for level to make sure that the found CI is contained in the closest possible level.

Returns `null` if no matching CI is found.

Parameters:

`operator` - the operator
`relationType` - the relation type

Returns:

the found descendant CI. `null`, if no matching CI is found.

---

getDescendantCiList

```
public java.util.List
getDescendantCiList(com.hp.opr.ts.interfaces.enrichment.IOperator operator)
```

Gets the list of descendant CIs that cause the specified operator to return `true`. Only relations of the specified relation type will be followed.=

Descendants will be searched level for level. The CIs will be inserted into the returned list in the order they are found level after level.

Returns `null` if no matching CI is found.

Parameters:

`operator` - the operator=

Returns:

the list of found descendant CIs. `null`, if no matching CI is found.

---

getDescendantCiList

```
public java.util.List
getDescendantCiList(com.hp.opr.ts.interfaces.enrichment.IOperator operator,
         java.lang.String relationType)
```

Gets the list of descendant CIs that cause the specified operator to return `true`. Only relations of the specified relation type will be followed. The algorithm stops following relations at the first relation that is not of the specified relation type.

Descendants will be searched level for level. The CIs will be inserted into the returned list in the order they are found level after level.

Returns `null` if no matching CI is found.

Parameters:

`operator` - the operator
`relationType` - the relation type

Returns:

the list of found descendant CIs. `null`, if no matching CI is found.

---

getChildCi

```
public ICi getChildCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator)
```

(continued from last page)

Gets the first child CI that causes the specified operator to return `true`. Only relations of the specified relation type will be followed.

Returns `null` if no matching CI is found.

Parameters:

`operator` - the operator

Returns:

the found child CI. `null`, if no matching CI is found.

---

getChildCi

```
public ICi getChildCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator,
        java.lang.String relationType)
```

Gets the first child CI that causes the specified operator to return `true`. Only relations of the specified relation type will be followed.

Returns `null` if no matching CI is found.

Parameters:

`operator` - the operator
`relationType` - the relation type

Returns:

the found child CI. `null`, if no matching CI is found.

---

getChildCiList

```
public java.util.List getChildCiList(com.hp.opr.ts.interfaces.enrichment.IOperator
operator)
```

Gets the list of child CIs that cause the specified operator to return `true`.

Returns `null` if no matching CI is found.

Parameters:

`operator` - the operator

Returns:

the list of found child CIs. `null`, if no matching CI is found.

---

getChildCiList

```
public java.util.List getChildCiList(com.hp.opr.ts.interfaces.enrichment.IOperator
operator,
        java.lang.String relationType)
```

Gets the list of child CIs that cause the specified operator to return `true`. Only relations with the specified relation type are taken into account.

Returns `null` if no matching CI is found.

Parameters:

`operator` - the operator
`relationType` - the relation type

Returns:

the list of found child CIs. `null`, if no matching CI is found.

getDependencyCi

```
public ICi getDependencyCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator)
```

Gets the first dependency CI that causes the specified operator to return `true`.

Returns `null` if no matching CI is found.

Parameters:
`operator` - the operator

Returns:
the found dependency CI. `null`, if no matching CI is found.

getDependencyCi

```
public ICi getDependencyCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator,
        java.lang.String relationType)
```

Gets the first dependency CI that causes the specified operator to return `true`. Only relations of the specified relation type will be followed.

Returns `null` if no matching CI is found.

Parameters:
`operator` - the operator
`relationType` - the relation type

Returns:
the found dependency CI. `null`, if no matching CI is found.

getDependencyCiList

```
public java.util.List
getDependencyCiList(com.hp.opr.ts.interfaces.enrichment.IOperator operator)
```

Gets the list of dependency CIs that cause the specified operator to return `true`. Only relations with the specified relation type are taken into account.

Returns `null` if no matching CI is found.

Parameters:
`operator` - the operator

Returns:
the list of found dependency CIs. `null`, if no matching CI is found.

getDependencyCiList

```
public java.util.List
getDependencyCiList(com.hp.opr.ts.interfaces.enrichment.IOperator operator,
         java.lang.String relationType)
```

Gets the list of dependency CIs that cause the specified operator to return `true`. Only relations with the specified relation type are taken into account.

Returns `null` if no matching CI is found.

Parameters:
`operator` - the operator

(continued from last page)

`relationType` - the relation type

Returns:
the list of found dependency CIs. `null`, if no matching CI is found.

getStringValueList

```
public java.util.List getStringValueList(java.lang.String xpath)
```

Select all properties matching the given XPath expression starting at the CI given in the constructor of this CI navigator instance.

See `getValue(String)` for more information and examples.

Parameters:
`xpath` - XPath expression that selects the desired properties.

Returns:
List of strings. null, if the xpath cannot be resolved due to wrong properties or syntax errors.

See Also:
`getValue(String)`

getStringValue

```
public java.lang.String getStringValue(java.lang.String xpath)
```

Get the String value of a Java property using an XPath expression using this CI as context.

See `getValue(String)` for more information and examples.

Parameters:
`xpath` - XPath pointing to the String property

Returns:
Property as string. null, if the path cannot be resolved due to wrong properties or syntax errors.

See Also:
`getValue(String)`

getDependentCi

```
public ICi getDependentCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator)
```

Gets the first dependent CI that causes the specified operator to return `true`.

Returns `null` if no matching CI is found.

Parameters:
`operator` - the operator

Returns:
the found dependent CI. `null`, if no matching CI is found.

getDependentCi

```
public ICi getDependentCi(com.hp.opr.ts.interfaces.enrichment.IOperator operator,
        java.lang.String relationType)
```

(continued from last page)

Gets the first dependent CI that causes the specified operator to return `true`. Only relations of the specified relation type will be followed.

Returns `null` if no matching CI is found.

Parameters:

`operator` - the operator
`relationType` - the relation type

Returns:

the found dependent CI. `null`, if no matching CI is found.

---

getDependentCiList

```
public java.util.List getDependentCiList(com.hp.opr.ts.interfaces.enrichment.IOperator
operator)
```

Gets the list of dependent CIs that cause the specified operator to return `true`. Only relations with the specified relation type are taken into account.

Returns `null` if no matching CI is found.

Parameters:

`operator` - the operator

Returns:

the list of found dependent CIs. `null`, if no matching CI is found.

---

getDependentCiList

```
public java.util.List getDependentCiList(com.hp.opr.ts.interfaces.enrichment.IOperator
operator,
         java.lang.String relationType)
```

Gets the list of dependent CIs that cause the specified operator to return `true`. Only relations with the specified relation type are taken into account.

Returns `null` if no matching CI is found.

Parameters:

`operator` - the operator
`relationType` - the relation type

Returns:

the list of found dependent CIs. `null`, if no matching CI is found.

com.hp.opr.ts.interfaces.data.ci
Interface INode

All Superinterfaces:
ICi

public interface **INode**
extends ICi

INode describes a specific `ICi` that models a CI that represents a physical network node.

Method Summary

`void`	`addIPAddressList(java.util.List ipAddressList)` Add the list of IP addresses of this node.
`void`	`addMACAddressList(java.util.List macAddressList)` Add the list of MAC addresses of this node.
`void`	`addNodeGroup(java.lang.String nodeGroup)` Add a target node group to this node.
`java.lang.String`	`getDiscoveryDomain()` Gets the DiscoveryDomain of this node.
`java.util.List`	`getIPAddressList()` Gets the list of IP addresses of this node.
`java.util.List`	`getMACAddressList()` Gets the list of MAC addresses of this node.
`java.util.Collection`	`getNodeGroups()` Gets the node groups that are assigned to this node.
`java.util.Iterator`	`getNodeGroupsIterator()` Get an iterator that returns all node groups into which this node is currently assigned.
`boolean`	`isAssignedToAnyNodeGroup()` Checks if the node has any node groups assigned to it.
`boolean`	`isAssignedToNodeGroup(java.lang.String nodeGroup)` Checks if the node is contained in the given node group.

Methods inherited from interface `com.hp.opr.ts.interfaces.data.ci.ICi`

addCmdbFromRelation, addCmdbToRelation, addMessage, addMessage, addMessage, addOmIdAlias, addOmRelation, addOmRelation, dispose, equals, getCmdbAttributeNames, getCmdbAttributes, getCmdbAttributeValue, getCmdbFromRelations, getCmdbId, getCmdbToRelations, getCmdbTypeId, getContexts, getHostedOn, getHostedOnOmId, getMessages, getMessages, getMessages, getMessages, getNavigator, getOmAttributeNames, getOmAttributes, getOmAttributeValue, getOmCaption, getOmChildCis, getOmChildRelations, getOmDependencyRelations, getOmDependentRelations, getOmId, getOmIdAliases, getOmParentCi, getOmParentRelation, getOmRootCi, getOmTypeId, getRootContainerOmId, hasContext, hasContexts, hashCode, hasMessages, hasMessages, isChildOf, isExternalStub, isNode, isParentOf, isRootCi, isService, printCi, removeRelation, removeRelation, setCaption, setCmdbAttribute, setCmdbId, setCmdbTypeId, setContext, setHostedOnOmId, setIsNode, setIsService, setOmAttribute, setOmId, setOmTypeId, setRootContainerOmId

Methods

addNodeGroup

```
public void addNodeGroup(java.lang.String nodeGroup)
```

Add a target node group to this node. The node will be placed into the specified node group

Parameters:
`nodeGroup` - Target node group name.

addMACAddressList

```
public void addMACAddressList(java.util.List macAddressList)
```

Add the list of MAC addresses of this node.

Parameters:
`macAddressList` - The list of MAC addresses of this node.

getMACAddressList

```
public java.util.List getMACAddressList()
```

Gets the list of MAC addresses of this node.

Returns:
the list of MAC addresses of this node

getNodeGroups

```
public java.util.Collection getNodeGroups()
```

Gets the node groups that are assigned to this node.

Returns:
the assigned node groups

getNodeGroupsIterator

```
public java.util.Iterator getNodeGroupsIterator()
```

(continued from last page)

Get an iterator that returns all node groups into which this node is currently assigned.

Returns:
Iterator of target node groups.

isAssignedToAnyNodeGroup

`public boolean isAssignedToAnyNodeGroup()`

Checks if the node has any node groups assigned to it.

Returns:
`true` if at least one node group has been assigned to this node.

isAssignedToNodeGroup

`public boolean isAssignedToNodeGroup(java.lang.String nodeGroup)`

Checks if the node is contained in the given node group.

Parameters:
`nodeGroup` - The node group.

Returns:
`true` if the node is already assigned to this node group.

addIPAddressList

`public void addIPAddressList(java.util.List ipAddressList)`

Add the list of IP addresses of this node.

Parameters:
`ipAddressList` - The list of IP addresses of this node.

getIPAddressList

`public java.util.List getIPAddressList()`

Gets the list of IP addresses of this node.

Returns:
the list of IP addresses of this node

getDiscoveryDomain

`public java.lang.String getDiscoveryDomain()`

Gets the DiscoveryDomain of this node.

Returns:
the DiscoveryDomain of this node

com.hp.opr.ts.interfaces.data.ci
Interface IRelation

All Subinterfaces:
ICiRelation

public interface **IRelation**
extends

The Interface ICmdbRelation describes the loosely coupled relation between two CIs, which are identified by their service IDs, as they will be created in the CMDB by the CMDB Gateway.

Method Summary

`java.lang.String`	`getFromCMDBId()` Gets the ID of the CI from which this CI relation origins.
`java.lang.String`	`getFromId()` Gets the ID of the CI from which this CI relation origins.
`java.lang.String`	`getToCMDBId()` Gets the ID of the CI to which this CI relation points to.
`java.lang.String`	`getToId()` Gets the ID of the CI to which this CI relation points to.
`java.lang.String`	`getType()` Get the type of the relation.
`void`	`setFromCMDBId(java.lang.String cmdbToId)` Sets the ID of the CI to which this CI relation origins.
`void`	`setToCMDBId(java.lang.String cmdbToId)` Sets the ID of the CI to which this CI relation points to.

Methods

getFromId

`public java.lang.String **getFromId**()`

Gets the ID of the CI from which this CI relation origins.

Returns:
the OM ID of the origin CI.

getToId

`public java.lang.String **getToId**()`

Gets the ID of the CI to which this CI relation points to.

Returns:

(continued from last page)

the OM ID of the target CI.

getFromCMDBId

`public java.lang.String getFromCMDBId()`

Gets the ID of the CI from which this CI relation origins.

Returns:
the CMDB CI of the origin CI.

setFromCMDBId

`public void setFromCMDBId(java.lang.String cmdbToId)`

Sets the ID of the CI to which this CI relation origins.

Parameters:
`cmdbToId` - the CMDB CI ID of the source CI to set.

getToCMDBId

`public java.lang.String getToCMDBId()`

Gets the ID of the CI to which this CI relation points to.

Returns:
the CMDB CI ID of the target CI.

setToCMDBId

`public void setToCMDBId(java.lang.String cmdbToId)`

Sets the ID of the CI to which this CI relation points to.

Parameters:
`cmdbToId` - the CMDB CI ID of the target CI to set.

getType

`public java.lang.String getType()`

Get the type of the relation.

Returns:
CI relation type

Package
com.hp.opr.ts.interfaces.data.sync

Contains a data structure that holds all the data that is being synchronized.

com.hp.opr.ts.interfaces.data.sync
Interface ISyncData

public interface **ISyncData**
extends

The Interface `ISyncData` describes a data structure that contains the normalized data that is being synchronized. It includes configuration items like services in the HP Operations Manager Service Navigator as well as nodes and nodegroups.

The list of configuration items contains CIs that are potentially created as CIs in the uCMDB.

The list of nodes contains CIs that represent physical nodes in the node bank of HP Operations Manager. Please note that a node is a sub type of a CI and might also be included in the list of CIs, depending on the current synchronization settings. In case you wish to manipulate the attributes or relations of a node, ensure that you do that only once and not for its occurrence in the list of configuration items a second time.

Method Summary

void	`addConfigurationItem(ICi configurationItem)` Adds a ConfigurationItem to the synchronization data.
ICi	`getConfigurationItem(java.lang.String id)` Returns the configuration item with the given ID
java.util.Collection	`getConfigurationItems()` Returns all configuration items that will be added as nodes or services to HP Operations Manager.
java.util.Collection	`getConfigurationItemsWithMessages()` Gets all configuration items that have at least one message attached.
java.util.Collection	`getConfigurationItemsWithMessages(CiMessageSeverity severity)` Gets all configuration items that have at least one message attached and that have at least one message with a severity that is equal to or higher than the given severity.
java.util.Collection	`getNodes()` Returns all configuration items representing physical nodes that will be added to HP Operations Manager.
java.util.Collection	`getNodesWithMessages()` Gets all nodes that have at least one message attached.
java.util.Collection	`getNodesWithMessages(CiMessageSeverity severity)` Gets all nodes which have at least one message attached and which have at least one message with a severity that is equal to or higher than the given severity.
boolean	`hasConfigurationItemsWithMessages()` Check if there is at least one configuration item containing a message.
boolean	`hasConfigurationItemsWithMessages(CiMessageSeverity severity)` Check if there is at least one configuration item containing a message that has a severity equal to or higher than the given severity.
boolean	`hasNodesWithMessages()` Check if there is at least one node containing a message.

boolean	hasNodesWithMessages(CiMessageSeverity severity) Check if there is at least one node containing a message that has a severity equal to or higher than the given severity.
void	removeConfigurationItem(java.lang.String ci) Removes the configuration item.

Methods

addConfigurationItem

`public void addConfigurationItem(ICi configurationItem)`

Adds a ConfigurationItem to the synchronization data. If the given CI is a physical node, it will also be added to the list of nodes.

Parameters:
`configurationItem` - The ConfigurationItem that gets added

Throws:
`IllegalArgumentException` - if `configurationItem` is null

getConfigurationItem

`public ICi getConfigurationItem(java.lang.String id)`

Returns the configuration item with the given ID

Parameters:
`id` - ID of the configuration item

Returns:
configuration item with the given ID

getConfigurationItems

`public java.util.Collection getConfigurationItems()`

Returns all configuration items that will be added as nodes or services to HP Operations Manager.

Returns:
List of configuration items.

getConfigurationItemsWithMessages

`public java.util.Collection getConfigurationItemsWithMessages()`

Gets all configuration items that have at least one message attached.

Returns:
the configuration items with messages

getConfigurationItemsWithMessages

`public java.util.Collection getConfigurationItemsWithMessages(CiMessageSeverity severity)`

(continued from last page)

Gets all configuration items that have at least one message attached and that have at least one message with a severity that is equal to or higher than the given severity.

Parameters:
`severity` - the severity to filter the returned configuration items.

Returns:
the configuration items with messages

getNodes

```
public java.util.Collection getNodes()
```

Returns all configuration items representing physical nodes that will be added to HP Operations Manager.

Returns:
List of nodes

getNodesWithMessages

```
public java.util.Collection getNodesWithMessages()
```

Gets all nodes that have at least one message attached.

Returns:
List of nodes with messages

getNodesWithMessages

```
public java.util.Collection getNodesWithMessages(CiMessageSeverity severity)
```

Gets all nodes which have at least one message attached and which have at least one message with a severity that is equal to or higher than the given severity.

Parameters:
`severity` - the severity to filter the returned nodes.

Returns:
the nodes with messages

hasConfigurationItemsWithMessages

```
public boolean hasConfigurationItemsWithMessages()
```

Check if there is at least one configuration item containing a message.

Returns:
true, if successful

hasConfigurationItemsWithMessages

```
public boolean hasConfigurationItemsWithMessages(CiMessageSeverity severity)
```

Check if there is at least one configuration item containing a message that has a severity equal to or higher than the given severity.

Parameters:
`severity` - the severity

(continued from last page)

Returns:
true, if successful

hasNodesWithMessages

```
public boolean hasNodesWithMessages()
```

Check if there is at least one node containing a message.

Returns:
true, if successful

hasNodesWithMessages

```
public boolean hasNodesWithMessages(CiMessageSeverity severity)
```

Check if there is at least one node containing a message that has a severity equal to or higher than the given severity.

Parameters:
`severity` - the severity

Returns:
true, if successful

removeConfigurationItem

```
public void removeConfigurationItem(java.lang.String ci)
```

Removes the configuration item.

Parameters:
`ci` - the name of the CI

com.hp.opr.ts.interfaces.data.sync
Interface ISyncTask

public interface **ISyncTask**
extends com.hp.opr.ts.interfaces.config.IPersistableSettings

The interface ISyncTask describes a synchronization task. This interface acts as read-only interface for a concrete implementation.

Method Summary

`java.util.List`	`getActiveSyncPackages()` Gets a `java.util.List` containing the active synchronization packages.
`com.hp.opr.ts.interfaces.enrichment.IMappingSettings`	`getAttributeMappingSettings()` Gets the consolidated attribute mapping rules of all bundles.
`com.hp.opr.ts.interfaces.enrichment.IMappingSettings`	`getContextMappingSettings()` Gets the consolidated context mapping rules of all bundles.
`java.lang.String`	`getEncryptedHpOmWsPassword()` Gets the HP OM password for accessing the Service Engine via WS interface.
`java.lang.String`	`getEncryptedUCmdbPassword()` Gets the encrypted uCMDB password.
`java.lang.String`	`getHpOmWsEpr()` Gets the EPR for accessing the HP OM server via WS interface.
`java.lang.String`	`getHpOmWsUserName()` Gets the HP OM username for accessing the Service Engine via WS interface.
`java.lang.String`	`getName()` Gets the name of the task.
`java.lang.String`	`getOMCIdMappingFile()` Gets the filename of the OM classic ID to uCMDB ID mapping file.
`java.lang.String`	`getOMType()` Returns the OMType.
`com.hp.opr.ts.interfaces.enrichment.IMappingSettings`	`getRelationMappingSettings()` Gets the consolidated context mapping rules of all bundles.
`boolean`	`getResolveIPs()` Returns true if IPs should be resolved during sync.
`com.hp.opr.ts.interfaces.enrichment.IMappingSettings`	`getTypeMappingSettings()` Gets the consolidated type mapping rules of all bundles.
`java.lang.String`	`getUCmdbHostname()` Gets the uCMDB hostname.
`int`	`getUCmdbPort()` Gets the uCMDB TCP/IP port.

java.lang.String	getUcmdbProtocol() Gets the uCMDB protocol (https or http).
java.lang.String	getUCmdbUsername() Gets the uCMDB username.
boolean	isDumpingData() Checks, if the SyncData is dumped to the file system during synchronization.
boolean	isScriptingEnabled() Returns true if scripting is enabled.
boolean	load(boolean loadSyncPackages) Load the synchronization task settings, but optionally ignore the synchronization packages.
boolean	load(boolean loadSyncPackages, boolean loadMappings) Load the synchronization task settings, but optionally ignore the synchronization bundles.

Methods inherited from interface com.hp.opr.ts.interfaces.config.IPersistableSettings
load, load, loadFromSettingsManager, save, save

Methods

load

```
public boolean load(boolean loadSyncPackages)
```

Load the synchronization task settings, but optionally ignore the synchronization packages. This can be used in cases, where you only need the direct settings of the synchronization task.

Parameters:

loadSyncPackages - false, if the synchronization packages should not be loaded.

Returns:

true, if successful

load

```
public boolean load(boolean loadSyncPackages,
                    boolean loadMappings)
```

Load the synchronization task settings, but optionally ignore the synchronization bundles. This can be used in cases, where you only need the direct settings of the synchronization task.

Parameters:

loadSyncPackages - false, if the synchronization packages should not be loaded.

loadMappings - false, if synchronization task should not load the mapping settings.

Returns:

true, if successful

getName

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

Gets the name of the task.

(continued from last page)

Returns:
the task name

getUCmdbUsername

`public java.lang.String getUCmdbUsername()`

Gets the uCMDB username.

Returns:
the uCMDB username

getUcmdbProtocol

`public java.lang.String getUcmdbProtocol()`

Gets the uCMDB protocol (https or http).

Returns:
the uCMDB protocol

getEncryptedUCmdbPassword

`public java.lang.String getEncryptedUCmdbPassword()`

Gets the encrypted uCMDB password.

Returns:
the uCMDB password

getHpOmWsUserName

`public java.lang.String getHpOmWsUserName()`

Gets the HP OM username for accessing the Service Engine via WS interface.

Returns:
the WS username

getEncryptedHpOmWsPassword

`public java.lang.String getEncryptedHpOmWsPassword()`

Gets the HP OM password for accessing the Service Engine via WS interface.

Returns:
the WS password

getHpOmWsEpr

`public java.lang.String getHpOmWsEpr()`

Gets the EPR for accessing the HP OM server via WS interface.

Returns:
the EPR

getTypeMappingSettings

```
public com.hp.opr.ts.interfaces.enrichment.IMappingSettings getTypeMappingSettings()
```

Gets the consolidated type mapping rules of all bundles.

Returns:
the consolidated type mapping settings

getAttributeMappingSettings

```
public com.hp.opr.ts.interfaces.enrichment.IMappingSettings
getAttributeMappingSettings()
```

Gets the consolidated attribute mapping rules of all bundles.

Returns:
the consolidated attribute mapping settings

getContextMappingSettings

```
public com.hp.opr.ts.interfaces.enrichment.IMappingSettings
getContextMappingSettings()
```

Gets the consolidated context mapping rules of all bundles.

Returns:
the consolidated attribute mapping settings

getRelationMappingSettings

```
public com.hp.opr.ts.interfaces.enrichment.IMappingSettings
getRelationMappingSettings()
```

Gets the consolidated context mapping rules of all bundles.

Returns:
the consolidated attribute mapping settings

isDumpingData

```
public boolean isDumpingData()
```

Checks, if the SyncData is dumped to the file system during synchronization.

Returns:
true, if SyncData is dumped to file system.

getActiveSyncPackages

```
public java.util.List getActiveSyncPackages()
```

Gets a `java.util.List` containing the active synchronization packages. The bundles are sorted concerning their bundle priority settings.

Returns:
the collection of sync packages, null if the collection is not set

getUCmdbHostname

`public java.lang.String getUCmdbHostname()`

Gets the uCMDB hostname.

Returns:
the uCMDB hostname

getUCmdbPort

`public int getUCmdbPort()`

Gets the uCMDB TCP/IP port.

Returns:
the uCMDB TCP/IP port

getOMCIdMappingFile

`public java.lang.String getOMCIdMappingFile()`

Gets the filename of the OM classic ID to uCMDB ID mapping file.

Returns:
the filename of the OM classic ID to uCMDB ID mapping file

getResolveIPs

`public boolean getResolveIPs()`

Returns true if IPs should be resolved during sync.

Returns:
resolve IPs during sync.

isScriptingEnabled

`public boolean isScriptingEnabled()`

Returns true if scripting is enabled.

Returns:
true is scripting is enabled.

getOMType

`public java.lang.String getOMType()`

Returns the OMType.

Returns:
OM for Windows or OM for Unix

Package

com.hp.opr.ts.interfaces.scripting

com.hp.opr.ts.interfaces.scripting
Interface IScriptingGateway

public interface **IScriptingGateway**
extends

The Interface IScriptingGateway.

Method Summary

boolean	runPostUcmbdScripts(ISyncData syncData, ISyncTask syncTask) Run post ucmbd scripts.
boolean	runPreEnrichmentScripts(ISyncData syncData, ISyncTask syncTask) Run pre enrichtment scripts.
boolean	runPreUcmbdScripts(ISyncData syncData, ISyncTask syncTask) Run pre ucmbd scripts.

Methods

runPreUcmbdScripts

```
public boolean runPreUcmbdScripts(ISyncData syncData,
          ISyncTask syncTask)
```

Run pre ucmbd scripts.

Parameters:
syncData - the sync data
syncTask - the sync task

Returns:
true, if successful

runPostUcmbdScripts

```
public boolean runPostUcmbdScripts(ISyncData syncData,
          ISyncTask syncTask)
```

Run post ucmbd scripts.

Parameters:
syncData - the sync data
syncTask - the sync task

Returns:
true, if successful

runPreEnrichmentScripts

```
public boolean runPreEnrichmentScripts(ISyncData syncData,
          ISyncTask syncTask)
```

(continued from last page)

Run pre enrichtment scripts.

Parameters:

`syncData` - the sync data
`syncTask` - the sync task

Returns:

true, if successful

com.hp.opr.ts.interfaces.scripting
Interface IScriptingInterface

public interface **IScriptingInterface**
extends

The `IScriptingInterface` describes the interface to be used by a developer of synchronization scripts for TopologySync. An instance of this interface is bound to the `scriptInterface` variable, which is available in every script.

This interface serves as a gateway to common function calls like logging, creation of resources and control of the synchronization process.

Method Summary

void	`abortSync(java.lang.String message)` Abort the synchronization.
ICi	`createCi()`
ICi	`createCi(java.lang.String id)` Create a new CI.
void	`createCmdbRelation(ICi from, ICi to, java.lang.String type)` Create a relation between two CIs.
INode	`createNode(java.lang.String id)` Create a new node with the given unique ID.
ICiRelation	`createOmRelation(ICi from, ICi to, java.lang.String type)` Create a relation between two CIs.
java.lang.Process	`exec(java.lang.String command)` Execute the given command on the TopologySync system.
java.lang.Process	`exec(java.lang.String[] commandArray)` Execute a command with arguments.
ICi	`getExternalCiStub(java.lang.String id)` Gets a stub of an external CI.
boolean	`isAbortSyncOnError()` Checks if the synchronization shall abort on error.
void	`logDebug(java.lang.String message)` Log a new message to the applications log file with a severity level of FINE.
void	`logFatal(java.lang.String message)` Log a new message to the applications log file with a severity level of SEVERE.
void	`logInfo(java.lang.String message)` Log a new message to the applications log file with a severity level of INFO.
void	`logWarning(java.lang.String message)` Log a new message to the applications log file with a severity level of WARNING.

void	performEnrichment(ICi ci) Perform all data enrichments for the given configuration item.
void	removeCi(ICi ci) Remove the given CI.
void	removeCiWithDescendants(ICi ci) Remove the given CI with all its descendants.
void	setAbortSyncOnError(boolean abortOnError) Set, if the synchronization shall be aborted on unexpected errors.

Methods

abortSync

```
public void abortSync(java.lang.String message)
```

Abort the synchronization. Use this method in a pre-upload script when you have detected conditions in your script that will prevent a proper synchronization. Using this method in a post-upload script will stop the execution of further post-upload scripts.

Parameters:
`message` - the message of the abort reason. This will be logged with a severity of SEVERE in the application log file.

Throws:
`SynchronizationAbortException` - Thrown every time this method is called to abort the synchronization.

createCi

```
public ICi createCi(java.lang.String id)
```

Create a new CI. The CI that is going to be created will be synchronized and created as service in the service navigator.

After you have created the CI with the given unique ID and have set values as well as included the CI in a topology by adding relations, you should perform an data enrichment on this CI to add TopologySync required properties and attributes using the method `performEnrichment(ICi)`. However this step is not required as long as you ensure that required properties are set.

When you are going to perform an enrichment on the CI you must set the property CI Type using `ICi.setOmTypeId(String)`. Then set the relations to this CI as well as all other desired properties and attributes before calling `performEnrichment(ICi)`.

In the case you do not want to perform an enrichment on the created CI make sure that the following attribute and all CMDB key attributes for the CM respective CMDB Type is set:

- Attribute: The CMDB Type Id `ICi.setCmdbTypeId(String)`

Parameters:
`id` - the unique ID of the CI

Returns:
the instance of the CI.

See Also:
`performEnrichment(ICi)`

createCi

```
public ICi createCi()
```

createNode

```
public INode createNode(java.lang.String id)
```

Create a new node with the given unique ID. A node represents a physical network node. Depending on the synchronization settings of HP Operations Manager a node will be added to the node bank and/or added as service.

Note that a node is a sub type of an CI. Therefore all actions that require a CI can also be performed with a node.

After you have created the node and have changed values as well as included the node in a topology by setting relations, you should perform a data enrichment on this node. However this step is not required as long as you ensure that required properties are set.

When you are going to perform an enrichment on this created node you must set the following properties and attributes:

- Property: OM Type using `ICi.setOmTypeId(String)`.
- Property: OM ID with the lowest MAC address using `ICi.setOmId(String)`.
- Property: OM Caption `ICi.setCaption(String)`.
- For the node to be properly managed by the uCMDB:
  Attribute: `Name` using the method `ICi.setOmAttribute(String, Object)`

After that set the relations to this node as well as all other desired properties and attributes before calling `performEnrichment(ICi)`.

In the case you do not want to perform an enrichment on the created node make sure that the following properties and attributes are set:

- Property: CI Type using the method `ICi.setCmdbTypeId(String)`
- For the node to be properly managed by HPOM:
  Attribute: `host_key` using the method `ICi.setCmdbAttribute(String, Object)`Attribute: `host_dnsname` using the method `ICi.setCmdbAttribute(String, Object)`Attribute: `host_hostname` using the method `ICi.setCmdbAttribute(String, Object)`
- Add the node to at least one node group using the method `INode.addNodeGroup(String)`

Parameters:

`id` - the unique ID of the node.

Returns:

the instance of the node

See Also:

`performEnrichment(ICi)`

createOmRelation

```
public ICiRelation createOmRelation(ICi from,
        ICi to,
        java.lang.String type)
```

Create a relation between two CIs.

The relation will be automatically added to the CI from which the relation points to the other CI using the method `ICi.addOmRelation(ICiRelation)`. Depending on the type of the relation the relation will then become a containment relation, i.e. a parent-child relation, or a dependency relation, when the relation is added to a CI.

(continued from last page)

Parameters:

`from` - the CI from which the relation points to the other CI.
`to` - the CI to which the relations points from the other CI.
`type` - the type of the relation.

Returns:

the instance of the relation

---

createCmdbRelation

```
public void createCmdbRelation(ICi from,
          ICi to,
          java.lang.String type)
```

Create a relation between two CIs.

The relation will be automatically added to the CI from which the relation points to the other CI using the method `ICi.addCmdbFromRelation(String, String, String)`. Depending on the type of the relation the relation will then become a containment relation, i.e. a parent-child relation, or a dependency relation, when the relation is added to a CI.

Parameters:

`from` - the CI from which the relation points to the other CI.
`to` - the CI to which the relations points from the other CI.
`type` - the type of the relation.

---

exec

```
public java.lang.Process exec(java.lang.String command)
```

Execute the given command on the TopologySync system. In case you wish to specify arguments for the command use the method `exec(String[])`.

Parameters:

`command` - the command to execute

Returns:

instance of `java.lang.Process` for the executed command.

Throws:

`SynchronizationScriptException` - In case the script cannot be executed or errors occurred during execution of the script.

See Also:

`Runtime.exec(java.lang.String)`
`exec(String[])`

---

exec

```
public java.lang.Process exec(java.lang.String[] commandArray)
```

Execute a command with arguments. The first element of the given array is the name of the command and the remaining elements specify the list of runtime arguments to this command.

Parameters:

`commandArray` - the command array containing the command and its arguments

Returns:

instance of `java.lang.Process` for the executed command.

(continued from last page)

Throws:

`IOException` - Signals that an I/O exception has occurred during the execution of the command.

`SynchronizationScriptException` - In case the script cannot be executed or errors occurred during execution of the script.

See Also:

`Runtime.exec(java.lang.String[])`

getExternalCiStub

`public ICi getExternalCiStub(java.lang.String id)`

Gets a stub of an external CI. An external CI stub is a minimal representation of a service (external CI) that already exists in HP Operations Manager. Basically, the returned CI will only contain the ID of that external CI. Changes to properties or attributes of the returned CI will have no effect on that existing external CI.

Use an external CI stub as reference, when you need to define relations.

Parameters:

`id` - the ID of the external CI

Returns:

the external CI stub

isAbortSyncOnError

`public boolean isAbortSyncOnError()`

Checks if the synchronization shall abort on error.

See `setAbortSyncOnError(boolean)` for more information.

Returns:

true, if is abort sync on error

See Also:

`setAbortSyncOnError(boolean)`

logDebug

`public void logDebug(java.lang.String message)`

Log a new message to the applications log file with a severity level of FINE. Use this level for logging debug output. Messages will also be visible in the trace monitor application ovtrcmon.exe.

Parameters:

`message` - the message to log.

logInfo

`public void logInfo(java.lang.String message)`

Log a new message to the applications log file with a severity level of INFO.

Parameters:

`message` - the message to log.

(continued from last page)

logFatal

```
public void logFatal(java.lang.String message)
```

Log a new message to the applications log file with a severity level of SEVERE.

Parameters:
`message` - the message to log.

logWarning

```
public void logWarning(java.lang.String message)
```

Log a new message to the applications log file with a severity level of WARNING.

Parameters:
`message` - the message to log.

performEnrichment

```
public void performEnrichment(ICi ci)
```

Perform all data enrichments for the given configuration item.

This function should be called on a CI after a script has changed its values or the topology in which the CI is contained.

Parameters:
`ci` - the configuration item to be enriched.

removeCi

```
public void removeCi(ICi ci)
```

Remove the given CI. The removed CI will no longer by synchronized and will be removed from the CI topology. The children of the removed CI will not be removed. Each child will have the current CI as its parent removed and therefore will become root CIs. If you wish to remove the CI with all children and their descendants, use the method `removeCiWithDescendants(ICi)`.

Parameters:
`ci` - the CI to be removed.

See Also:
`removeCiWithDescendants(ICi)`

removeCiWithDescendants

```
public void removeCiWithDescendants(ICi ci)
```

Remove the given CI with all its descendants. Use this method to remove a whole subtree of a CI topology. All affected CIs will no longer be synchronized.

Parameters:
`ci` - the CI that and of which all descendants shall be removed.

setAbortSyncOnError

```
public void setAbortSyncOnError(boolean abortOnError)
```

(continued from last page)

Set, if the synchronization shall be aborted on unexpected errors.

The default value is `true`, which means, that the synchronization will be aborted as soon as an exception is thrown in the script.

When this is set to `false` calling the method `abortSync(String)` will still abort the synchronization while all other error conditions will lead to the rest of the current script being ignored but the next scripts will be executed.

Parameters:

`abortOnError` - `true` if the synchronization shall be aborted on an error. `false`, if the synchronization shall ignore the rest of the current script and continue.

See Also:

`setAbortSyncOnError(boolean)`