



OnCommand® Insight 7.2

Connect API Reference

March 2016 | 215-10387_A0
doccomments@netapp.com

Contents

OnCommand Insight Connect API overview	9
What can you do with OnCommand Insight Connect API?	9
Integration of storage data into a centralized IT dashboard	9
Integration of storage monitoring into a Help Desk Network Operations Center (NOC)	9
Development of specialized reports	10
Integration into your custom applications	10
OnCommand Insight Connect API changes	10
New Connect API methods	10
Deprecated Connect API methods	12
New Connect API classes	12
Changes to Connect API classes	13
OnCommand Insight Connect API prerequisites	14
How Connect API is licensed	14
What are the main Connect API object types?	14
How is Connect API authentication performed?	14
Connect API sessions and iterators	15
Opening a Connect API session	15
Closing a Connect API session	16
Accessing OnCommand Insight Connect API examples	16
Logging in to the OnCommand Insight Administration portal	16
Importing annotation values using OnCommand Insight Connect API	17
Where to find more information about OnCommand Insight	18
OnCommand Insight on the web	18
Locating OnCommand Insight documentation	18
OnCommand Insight Connect API example	19
Setting up your environment	19
Creating your project area	20
Creating your first connection to OnCommand Insight	21
Extracting data from OnCommand Insight	22
Extracting data with paged results	23
What's next in OnCommand Insight Connect API projects?	24
Connect API endpoint methods	25
addHostToApplication (deprecated)	25
addHostToApplicationById	26
addReservationRequests	26
addReservationRequirements	27
addVmToApplication	28
closeSession	28
dismissPathOutageViolations	29
findObjectByWwn	30

findObjectsByIp	30
getAllApplications	31
getAPIVersion	32
getApplication	33
getApplicationOfInternalVolume	33
getApplicationOfQtree	34
getApplicationOfShare	35
getApplicationOfVolume	36
getApplicationsOfHost	36
getApplicationsOfVm	37
getBusinessEntities	38
getBusinessEntity	39
getBusinessEntityOfPort	39
getBusinessEntityOfStorage	40
getBusinessEntityOfSwitch	41
getClosedViolations	41
getConnectedPorts	42
getDeviceGroup	43
getDeviceGroups	44
getDeviceGroupsByStorage	44
getDeviceGroupsByVolume	45
getDeviceGroupVolumes	46
getDRPathsByVolume	47
getFabricsByPorts	48
getGeneric	48
getGenerics	49
getHost	50
getHostPolicy	50
getHosts	51
getHostsOfApplication	52
getInternalVolume	53
getInternalVolumes	53
getInternalVolumesByStorageArray	54
getLogicalPortsByLogicalSwitch	55
getLogicalPortsByPhysicalPort	56
getLogicalPortsByPhysicalSwitch	57
getLogicalSwitchesByPhysicalSwitch	58
getNodesOfDevice	58
getPathPolicy	59
getPaths	60
getPathsByHost	61
getPendingPathReservations	62
getPhysicalPortByLogicalPort	62
getPolicies	63
getPort	64

getPorts	65
getPortsOfDevice	66
getPortsOfDeviceType	66
getPortsOfNode	67
getQtrees	68
getRecentDevice	69
getRecentHost	70
getRecentStorageArray	70
getRecentSwitch	71
getRecentTape	72
getRecentVolume	73
getReservationRequests	73
getReservationRequestsByStates	74
getReservationRequirement	75
getReservationRequirements	75
getReservationViolationTypeNames	76
getSANscreenVersion	77
getShares	77
getStorageArray	78
getStorageArrays	79
getSwitch	79
getSwitches	80
getTape	81
getTapes	82
getViolations	83
getViolationsByHost	83
getViolationsByRequests	84
getViolationsByReservations	85
getVirtualMachine	86
getVirtualMachines	86
getVolume	87
getVolumeMapsByStorageArray	88
getVolumeMasksByStorageArray	89
getVolumes	90
getVolumesByStorageArray	90
getVolumesSynchStatesByDRPath	91
getVolumesSynchStatesBySourceVolume	92
getZonesByPorts	93
openSession	94
removeHostFromApplication (deprecated)	94
removeHostFromApplicationById	95
removeHostPolicies	96
removeInternalVolumeFromApplication	97
removePolicies	97
removeQtreeFromApplication	98

removeReservationRequirements	99
removeShareFromApplication	99
removeVmFromApplication	100
removeVolumeFromApplication	101
setAnnotationValues	101
setApplication	102
setApplicationOfInternalVolume	103
setApplicationOfQtree	104
setApplicationOfShare	104
setApplicationOfVolume	105
setApplicationWithNoBusinessEntity	106
setBusinessEntity	107
setBusinessEntityOfPort	107
setBusinessEntityOfStorage	108
setBusinessEntityOfSwitch	109
setHostsPolicy	110
updateReservationRequests	111
updateReservationRequirements	111
getStoragePools	112
getStoragePoolsByStorageArray	113
getStoragePool	114
getVolumesByStoragePool	114
getInternalVolumesByStoragePool	115
Connect API classes	117
APIBadRequestException	117
Action	118
ActionResponse	119
AnnotationEnumValue	120
AnnotationType	121
AnnotationValue	123
APIException	124
APISessionContext	125
Application	126
ApplicationResponse	129
Attribute	129
BaseObject	130
BaseResponse	131
BusinessEntity	132
DeviceBase	133
DeviceGroup	134
DeviceGroupResponse	136
DRPath	136
DRPathResponse	138
Generic	139
GenericResponse	140

HistoryObject	141
Host	142
HostResponse	142
InternalVolume	143
InternalVolumeResponse	148
LogicalPort	149
LogicalPortResponse	149
Node	150
NodeResponse	152
Path	153
PathReservation	154
PathReservationResponse	155
PathResponse	156
Policy	157
PolicyResponse	159
Port	159
PortResponse	162
PortsByFabric	163
PortsByFabricResponse	163
PortsByZone	164
PortsByZoneResponse	165
Qtree	165
RequestIterator	167
ReservationRequest	169
ReservationRequestTarget	173
ReservationRequirement	174
ReservationViolation	175
ReservationViolationsByRequest	176
ReservationViolationsByRequestResponse	177
SANscreenAPIFactory	178
Share	179
StorageArray	180
StorageArrayResponse	182
StoragePool	183
StoragePoolResponse	188
Switch	189
SwitchResponse	190
SynchronizationResponse	191
Tape	192
TapeResponse	193
Task	194
TaskResponse	196
TaskViolationResponse	197
Violation	198
ViolationResponse	200

VirtualMachine	201
VirtualMachineResponse	204
Volume	205
VolumeMap	208
VolumeMapResponse	210
VolumeMask	211
VolumeMaskResponse	212
VolumeResponse	213
VolumesSynchronizationState	214
VolumesSynchronizationStateResponse	216
Web services WSDL	217
Accessing the WSDL specification	217
Connect API web services WSDL	217
OnCommand Insight Connect API code sample scripts and packages	218
Java examples	218
Perl examples	219
Copyright information	221
Trademark information	222
How to send comments about documentation and receive update notifications	223
Index	224

OnCommand Insight Connect API overview

OnCommand Insight Connect API enables NetApp customers and independent software vendors (ISVs) to integrate OnCommand Insight with other applications, such as monitoring and reporting systems. The Connect API lets you use your own applications and show OnCommand Insight data.

The Connect API is based on SOAP and web services. This provides the flexibility to use the API from many programming environments, including Java and Perl.

Note: Java is the only API supported by NetApp. Perl API and other examples are provided for reference only with no support.

What can you do with OnCommand Insight Connect API?

Using the OnCommand Insight Connect API, you can create many applications that perform tasks such as integrate storage data into a dashboard, integrate storage monitoring into a Help Desk Operations Center, create specialized reports, and integrate Connect API into other applications.

You can use the Connect API to initiate a session with the OnCommand Insight server, issue queries, retrieve data, and then close the session.

Integration of storage data into a centralized IT dashboard

You can use the OnCommand Insight Connect API to integrate storage data into a centralized IT dashboard to improve visibility and troubleshooting.

IT dashboards often display information from a centralized database, sometimes referred to as a CMDB for IT assets, configurations, status and relationships. These dashboards can extend their reach to include storage information from the server to the data.

The OnCommand Insight Connect API can help you provide:

- Consolidated reports including cost, utilization, and activity
- Consolidated chargeback and capacity planning
- Information for end-to-end troubleshooting

Integration of storage monitoring into a Help Desk Network Operations Center (NOC)

You can use the OnCommand Insight Connect API in the NOC to monitor the near real-time status of the storage environment. With the API, you can query outstanding violations and notify a monitoring framework system of any service violation such as outage, redundancy, or cluster sharing problem.

OnCommand Insight includes alternative methods for violation alerting using SNMP and email notification. However, using the API provides the required flexibility to add organization-specific logic around the violation notification.

Development of specialized reports

You can use the OnCommand Insight Connect API to provide frequent reports on the allocation of storage to different servers, enabling chargeback to the storage users based on their actual usage of production and replicated capacity.

Integration into your custom applications

Software vendors can use the OnCommand Insight Connect API to build applications that retrieve storage information into their own IT management software to provide visibility from their own applications into storage services, problems, and root-cause analysis. Such vendors include IT management framework developers, database management systems, and others.

OnCommand Insight Connect API changes

In each release, you can use this reference section to find out the new or changed methods, deprecated methods, and updated classes.

New Connect API methods

The following methods were added.

Added in OnCommand Insight 6.4

getFabricByPorts

Retrieves fabric by port IDs.

getInternalVolumesByStoragePool

Retrieves all internal volumes for a given storage pool.

getPendingPathReservations

Retrieves all pending path reservations.

getReservationRequirement

Retrieves a storage requirement object with the given ID.

getReservationViolationTypeNames

Retrieves a list of possible reservation violation types.

getStoragePool

Retrieves storage pool by storage pool ID.

getStoragePools

Retrieves all storage pools.

getStoragePoolsByStorageArray

Retrieves all storage pools for a given storage array.

getViolationsByRequests

Retrieves violations by request IDs.

getViolationsByReservations

Retrieves all the violations for given reservation

getVolumesByStoragePool

Retrieves all the volumes for a given storage array.

getZonesByPorts

Retrieves zones by port IDs.

Added in OnCommand Insight 6.2**addHostToApplicationById**

Associates a host with an application.

addVmToApplication

Associates a virtual machine with an application.

getApplicationOfInternalVolume

Retrieves the application that is associated with a given internal volume.

getApplicationOfQtree

Retrieves the application that is associated with a given qtree.

getApplicationOfShare

Retrieves the application that is associated with a given share.

getApplicationOfVolume

Retrieves the application that is associated with a given volume.

getApplicationsOfVm

Retrieves all the applications that are associated with a given virtual machine.

getBusinessEntities

Finds all the business entities.

getBusinessEntity

Finds the business entity with the given id.

getBusinessEntityOfPort

Gets the business entity of a port.

getBusinessEntityOfStorage

Gets the business entity of a storage.

getBusinessEntityOfSwitch

Gets the business entity of a switch.

removeHostFromApplicationById

Removes the association between a host and an application.

removeInternalVolumeFromApplication

Removes an association between an internal volume and an application.

removeQtreeFromApplication

Removes the association between a qtree and an application.

removeShareFromApplication

Removes the association between a share and an application.

removeVmFromApplication

Removes the association between a virtual machine and an application.

removeVolumeFromApplication

Removes the association between a volume and an application.

setApplicationOfInternalVolume

Associates an internal volume with an application.

setApplicationOfQtree

Associates a qtree with an application.

setApplicationOfShare

Associates a share with an application.

setApplicationOfVolume

Associates a volume with an application.

setApplicationWithNoBusinessEntity

Adds a new application or updates the attributes of an existing application.

setBusinessEntity

Adds a business entity if it does not already exist.

setBusinessEntityOfPort

Sets or unsets the business entity of a port.

setBusinessEntityOfStorage

Sets or unsets the business entity of a storage.

setBusinessEntityOfSwitch

Sets or unsets the business entity of a switch.

Added in OnCommand Insight 6.x

getQtrees

Retrieves all the qtrees.

getShares

Retrieves all the shares.

Deprecated Connect API methods

The following methods have been deprecated and are no longer supported.

Deprecated since OnCommand Insight 6.x

addHostToApplication

Associates a host with an application.

Instead, use `addHostToApplicationById`.

removeHostFromApplication

Removes an association between a host and an application.

Instead, use `removeHostFromApplicationById`.

Deprecated since SANscreen 5.1.2

findVirtualMachinesByName

Finds Virtual Machines based on the regular expression for their names.

Instead, use new methods that get virtual machines like `getVirtualMachines(...)`.

New Connect API classes

The following classes were added.

Added in OnCommand Insight 6.4

PathReservation

Contains metadata for path reservation.

PathReservationResponse

Contains an array of PathReservation objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

PortsByFabric

Contains Fabric objects and their associated portIds.

PortsByFabricResponse

Contains an array by PortsByFabric objects.

PortsByZone

Contains Zone object and its associated portIds.

PortsByZoneResponse

Contains an array of PortsByZone objects.

ReservationViolation

Contains metadata for reservationViolation.

ReservationViolationsByRequest

Contains requestId and its associated ReservationViolation objects.

ReservationViolationsByRequestResponse

Contains an array of ReservationViolationsByRequest objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

SANscreenAPIFactory

Simplifies the creation of API connections and sessions for interaction with the OnCommand Insight server.

StoragePool

Retrieves the storage pool object.

StoragePoolResponse

Contains an array of storage pools.

Changes to Connect API classes

Connect API classes are sometimes added or updated.

Changes since OnCommand Insight 6.2

Application

New attributes and methods and revisions to existing methods.

Volume

New methods.

Qtree

New

Share

New

BusinessEntity

New

OnCommand Insight Connect API prerequisites

To use OnCommand Insight Connect API, some prerequisites are required.

The following software requirements are necessary to develop programs that operate with the OnCommand Insight Connect API environment:

- The latest JDK downloaded and installed.
- Apache ANT downloaded and installed. This is required to use `build.xml`, which is part of the provided Java examples.
- Environment variables for ANT and JDK configured to run together.
The `java_home`, `path` set up for quick access to ANT.
- Ability to configure environment variables.

Note: OnCommand Insight does not actively support languages other than Java for use with the API.

The following item is recommended:

- A Perl interpreter installed to use the Perl-based examples.
ActivePerl is available for free download with registration. Perl examples have a `readme.txt` file that explains which Perl libraries must be downloaded and installed.

How Connect API is licensed

Using the OnCommand Insight Connect API requires the OnCommand Insight Discover software license key and an API license agreement.

What are the main Connect API object types?

You can work with object types as available in Connect API

Using the Connect API, the following objects are accessible:

- Devices: hosts, generic (HBA) devices, tapes, storage arrays, switches
- Device configuration: volumes, mapping, masking, zones, fabrics, VSAN
- Paths, policies, and violations
- Tasks, actions (ST)

How is Connect API authentication performed?

Authentication is based on HTTPS, which sends the user name and password over an encrypted communication channel.

The user name and password must correlate with a user entered in the OnCommand Insight User Management views in the OnCommand Insight Administration portal. The user must have administration privileges.

For security purposes, every action performed using the API is recorded in the OnCommand Insight audit table.

Connect API sessions and iterators

You can use the API to initiate a session with the OnCommand Insight server, issue queries, retrieve data, and then close the session. Opening a session provides a session context that is required for all queries. The API requires the use of sessions and iterators.

The API is designed to support large, ever-changing environments. To use the API, you use sessions and iterators similar to the way you use connections and cursors when accessing a database.

Use sessions and iterators in the following ways:

- A session maintains data consistency between multiple queries.
To assure consistency, queries to the same session respond with data that is consistent with a specific point in time.
- An iterator divides data when the data returned from a query exceeds a user-defined maximum.

Opening a Connect API session

Using the OnCommand Insight Connect API requires that you connect to the server and open a session. Opening a session provides a session context that is a required argument for all other queries.

About this task

A session is a snapshot of OnCommand Insight data at the time the session is created. The purpose of using a data snapshot rather than live data is to guarantee that multiple API queries provide a consistent view of OnCommand Insight data.

Note: Because the session uses a data snapshot rather than live data, any queries that you make to a session after opening it do not reflect changes that might have occurred after opening the session.

The session context variable is a required argument for every method that queries the OnCommand Insight server.

Steps

1. Connect to the server.

Depending on the platform used, connecting to the server yields the `SANscreenAPIEndPoint` end point that you can use to call API methods.

2. Open a session by entering the following:

```
SANscreenAPIFactory.openCurrentTimeSessionContext
(SANscreenAPIEndPoint apiEndPoint, int responseSizeLimit)
```

3. Issue queries by calling any of the `get*` methods.

The primary `get*` methods are listed in the `SANscreenAPIEndPoint` class.

Example

For example, call the following method to get storage array information:

```
getStorageArrays (APISessionContext sessionContext,
RequestIterator requestIterator)
```

Result

Queries for multiple values matching a given criterion returns the following:

- A response object encapsulating a subset of the values matching a given query.
- The number of values limited by the limit argument specified when a session is opened.
- An iterator that is used for determining if there are more values to be queried. Information about the iterator is located in the examples section.
If the iterator indicates that further queries should be issued, the query is reissued and data is provided along with the iterator in every new request.

Closing a Connect API session

Although you can close a session at any time, you should at least close one after the client finishes querying the server. Closing a session releases resources associated with a session so that they can be used in later calls.

Step

1. Call the `SANScreenAPIEndPoint` method: `closeSession(APISessionContext sessionContext)`

Accessing OnCommand Insight Connect API examples

You can review OnCommand Insight Connect API examples provided in the OnCommand Insight web portal.

About this task

The examples are provided in a zipped file.

Steps

1. Log into the OnCommand Insight Administration web portal as an administrator.
2. From the Advanced menu, select **Insight Connect API**.
3. From the OnCommand Insight Connect API page, select **Examples**.

Logging in to the OnCommand Insight Administration portal

You can access the OnCommand Insight Administration portal using a web browser.

About this task

The following table lists the default user name and password. You must change these default values as soon as possible after the installation.

Data	Value
Default user name	admin
Default password	admin123

If you do not initiate any activity within a time specified in Settings, then five minutes before the end of that interval, a message is displayed, warning that you will be logged out in five minutes if no activity occurs.

Steps

1. Do one of the following:
 - If you are accessing the system from the OnCommand Insight Server, enter:
`http://localhost`
 - If you are accessing the system from any location, enter the following URL:
`http://<OnCommand_Insight_Server_hostname>:<OnCommand_Insight_Server_port>`
The port number is either 80 or 8080, as configured when the Server was installed. The port number defaults to 80 if you do not specify it in the URL.
2. In the OnCommand Insight Administration portal, click one of the menu options.
3. In the **Login** window, enter your user name and password and click **OK**.

Importing annotation values using OnCommand Insight Connect API

If you maintain annotations on SAN objects (such as storage, hosts, and virtual machines) in a CSV file, you can import that information into Insight. You can import applications, business entities, or annotations such as tier and building.

About this task

The following rules apply:

- If an annotation value is empty, that annotation is removed from the object.
- When annotating volumes or internal volumes, the object name is a combination of storage name and volume name using the dash and arrow "-">" separator. For example:

```
<storage_name>-><volume_name>
```

- When storage, switches, or ports are annotated, the Application column is ignored.
- The columns of Tenant, Line_of_Business, Business_Unit, and Project make up a business entity. Any of the values can be left empty. If an application is already related with a business entity different from the input values, the application is assigned to the new business entity.

The following object types are supported in the import utility:

Type	Name or Key
Host	<Name> or <IP>
VM	<Name>
Internal Volume	<Storage_name>-> or <Internal_volume_name>
Volume	<Storage_name>-> or <Volume_name>
Storage	<Name> or <IP>
Switch	<Name> or <IP>
Port	<WWN>

The CSV file should use the following format:

```
, , <Annotation Type> [ , <Annotation Type> ...]
[ , Application] [ , Tenant] [ , Line_Of_Business] [ ,
Business_Unit] [ , Project]

<Object Type Value 1>, <Object Name or Key 1>, <Annotation Value> [ ,
<Annotation Value> ...] [ , <Application>] [ , <Tenant>] [ ,
<Line_Of_Business>] [ , <Business_Unit>] [ , <Project>]

<Object Type Value 2>, <Object Name or Key 2>, <Annotation Value> [ ,
<Annotation Value> ...] [ , <Application>] [ , <Tenant>] [ ,
<Line_Of_Business>] [ , <Business_Unit>] [ , <Project>]

<Object Type Value 3>, <Object Name or Key 3>, <Annotation Value> [ ,
<Annotation Value> ...] [ , <Application>] [ , <Tenant>] [ ,
<Line_Of_Business>] [ , <Business_Unit>] [ , <Project>]

...

<Object Type Value N>, <Object Name or Key N>, <Annotation Value> [ ,
<Annotation Value> ...] [ , <Application>] [ , <Tenant>] [ ,
<Line_Of_Business>] [ , <Business_Unit>] [ , <Project>]
```

Steps

1. From the OnCommand Insight web portal, select **Insight Connect API**.
2. From the Insight Connect API menu, select **Annotation Import Utility**.
3. Open the zip file.
4. Read the readme.txt file for additional information and samples.
5. In the command line window, enter the following:

```
java -jar import-utility.jar [-u<username>] [-p<password>]
[-a<server name or IP address>] <csv filename>
```

Where to find more information about OnCommand Insight

You can find more information about OnCommand Insight on the NetApp Support Site and in other OnCommand Insight documentation.

OnCommand Insight on the web

For comprehensive, up-to-date information about OnCommand Insight, use these NetApp web site resources.

- OnCommand Insight product web site at www.netapp.com/oncommandinsight
- The NetApp Support Site at: mysupport.netapp.com
- The OnCommand Insight data source Interoperability Matrix at mysupport.netapp.com/matrix.

Locating OnCommand Insight documentation

You can access the OnCommand Insight guides at the the NetApp Support Site mysupport.netapp.com/documentation/productsatoz/index.html to learn how to use the product.

OnCommand Insight Connect API example

This section guides you through an example of creating a simple project that uses the OnCommand Insight Connect API to extract information from OnCommand Insight.

In this example, the names of all storage arrays exist within an instance of OnCommand Insight.

The example uses command line tools to build and execute your first program. The same example can be done using any of the available Java IDEs.

The example includes the following major steps, each explained within the example:

1. Setting up your environment
2. Creating your project area
3. Creating your first connection to OnCommand Insight
4. Extracting data from OnCommand Insight
5. Extracting data with paged results
6. What's next?

Setting up your environment

Before you can create the project, you must set up the tools that allow you to build your project.

About this task

The example project requires the following tools:

- Java SE 6.0 JDK or later
- Apache Ant

Steps

1. Download and install Java SE 6.0 JDK or later from the following location:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

The location of the Java installation is referred to in this example as <JAVA_INSTALLDIR>

2. Download and install Apache Ant from the following location:

<http://ant.apache.org/>

3. Open a command line and enter the following commands:

```
set PATH=%PATH%;<ANT_INSTALL_DIR>\bin
set JAVA_HOME=<JAVA_INSTALL_DIR>
```

This sets your build environment.

4. To ensure that your environment is running properly, enter the following command:

```
ant -version
```

Note: When opening a new command line, you must run the commands above again to build any project.

Creating your project area

As part of creating a project, you need to know how to create a project area to allow you to build your application code.

About this task

You will create a build file that uses these targets:

Build target

Compiles your classes under `src` and generates a Java jar file for them under `build\dist`.

Run target

Executes your `OCI_HelloWorld` class within the generated Java jar file.

To be able to execute this jar file, all Java jar files under `lib` were included in your execution classpath.

Steps

1. Create a directory for your project.

The rest of the instructions refer to this project directory as `<PROJECT_DIR>`.

2. Download the API examples .zip from your OnCommand Insight portal page.

For instructions on how to access the examples, see "Accessing OnCommand Insight Connect API examples."

3. Uncompress the .zip file.

4. Create a `lib` directory in `PROJECT_DIR` and copy the contents of `java/lib` into this `lib` directory.

5. Create a `build.xml` file in your `<PROJECT_DIR>` with the following content using your preferred text editor:

```
<!-- Project for the OCI HelloWorld -->
<project name="OCI HelloWorld" default="build" basedir=". ">

    <!-- define the classpath for building the project -->
    <path id="build.classpath">
        <!-- include jar files under lib directory -->
        <fileset dir="lib" includes="*.jar" />
    </path>

    <!-- define the classpath for running the project -->
    <path id="run.classpath">
        <!-- include jar files under lib directory -->
        <fileset dir="lib" includes="*.jar" />

        <!-- include the jar file resulting from building the project -->
        <pathelement path="build/dist/OCI_HelloWorld.jar" />
    </path>

    <!-- build the jar file for the project -->
    <target name="build" description="Build the HelloWorld program">

        <!-- first compile your java class code -->
        <mkdir dir="build/classes" />
        <javac srcdir="src" destdir="build/classes">
            <classpath refid="build.classpath" />
        </javac>

        <!-- now assemble a jar file from the compiled classes -->
        <mkdir dir="build/dist" />
        <jar jarfile="build/dist/OCI_HelloWorld.jar">
            <fileset dir="build/classes" includes="**/*.*" />
        </jar>
    </target>
```

```

<!-- run the project jar file -->
<target name="run" description="Run the HelloWorld program">
  <!-- run java on the resulting code -->
  <java classname="OCI_HelloWorld">
    <classpath refid="run.classpath" />
  </java>
</target>

<!-- reset build environment to starting point -->
<target name="clean" description="Cleans up your project build artifacts">
  <delete dir="build" />
</target>

</project>

```

This build file will compile your sources and create a Java .jar file for executing your code.

6. Create an `src` directory for your sources.

All your source code will be placed here.

Creating your first connection to OnCommand Insight

As part of creating a project, you need to know how to create your first program interacting with OnCommand Insight. This step outlines the typical skeleton of interaction with OnCommand Insight and outlines how you can connect to OnCommand Insight.

Before you begin

Your command line must already be initialized by using the `set PATH` and `set JAVA_HOME` commands.

About this task

This example incorporates the following points:

- All access to the server is performed within a `try` or `finally` block. In the `finally` section, all open connections are closed.
- From the connection to the server, this code opens a session for server interaction. This session is used for keeping interaction state (meaning the paging information). In a later step, the meaning of the 10000 is explained.
- All interaction with server is done through the `apiEndPoint` and `apiSessionContext` objects.

This code creates the following:

- A `SANScreenAPIEndPoint` object using the server's hostname, user name, and password.
- A hostname, user name, and password with privileges to access your server.

Steps

1. Create a new file `src\OCI_HelloWorld.java` with the following content using your preferred text editor:

```

import com.onaro.sanscreen.wsapi.APISessionContext;
import com.onaro.sanscreen.wsapi.SANScreenAPIEndPoint;
import com.onaro.sanscreen.wsapi.utils.SANScreenAPIFactory;

/**
 * OnCommand Insight API Hello World:
 * Retrieving names of all storage arrays in the remote system
 *
 * @author Netapp
 * @since December 7, 2012
 */
public class OCI_HelloWorld {

```

```

public static void main(String[] args) throws Exception {
    SANscreenAPIEndPoint apiEndPoint = null;
    APISessionContext apiSessionContext = null;
    try {
        // create API end point for communication
        apiEndPoint = SANscreenAPIFactory.getEndPoint
            ("YOURHOSTNAME", "YOURUSERNAME", "YOURUSERPASSWORD");
        // create interaction session context for communication
        apiSessionContext =
            SANscreenAPIFactory.openCurrentTimeSessionContext(apiEndPoint, 10000);

        System.out.println("HelloWorld has connected to OCI server successfully!");

    } finally {
        // house cleanup, close session
        if (apiEndPoint != null && apiSessionContext != null) {
            apiEndPoint.closeSession(apiSessionContext);
        }
    }
}

```

2. Replace *YOURHOSTNAME* by your specific host.
3. Replace *YOURUSERNAME* and *YOURUSERPASSWORD* with your specific credential information.
4. Compile your code by running the following command:

```
ant clean build
```

5. Verify that code build completes successfully.
6. Run the build project by running the following command:

```
ant run
```

If your program is running, you will see the connection success message.

Extracting data from OnCommand Insight

In this step, you learn how to use the connection to OnCommand Insight and start extracting some information from OnCommand Insight.

About this task

This section of code applies the following points:

- After connecting to the server, the code now retrieves all storage systems by calling `getStorageArraysMethod` and then looping over the results to display the storage array names.
- The import statements for new classes were added to code.

Steps

1. Using your preferred text editor, edit the `src\OCI_HelloWorld.java` code by entering the following:

```

import com.onaro.sanscreen.wsapi.APISessionContext;
import com.onaro.sanscreen.wsapi.SANscreenAPIEndPoint;
import com.onaro.sanscreen.wsapi.StorageArray;
import com.onaro.sanscreen.wsapi.StorageArrayResponse;
import com.onaro.sanscreen.wsapi.utils.SANscreenAPIFactory;

/**
 * OnCommand Insight API Hello World:
 * Retrieving names of all storage arrays in the remote system
 *
 * @author Netapp
 * @since March 7, 2013

```

```

*/
public class OCI_HelloWorld {

    public static void main(String[] args) throws Exception {
        SANscreenAPIEndPoint apiEndPoint = null;
        APISessionContext apiSessionContext = null;
        try {
            // create API end point for communication
            apiEndPoint = SANscreenAPIFactory.getEndPoint
                ("YOURHOSTNAME", "YOURUSERNAME", "YOURUSERPASSWORD");
            // create interaction session context for communication
            apiSessionContext =
                SANscreenAPIFactory.openCurrentTimeSessionContext(apiEndPoint, 10000);

            // call for retrieval of all storage arrays in remote system
            StorageArrayResponse response =
                apiEndPoint.getStorageArrays(apiSessionContext, null);
            // loop through response, print each storage array name
            for (StorageArray oneStorageArray : response.getArrays()) {
                System.out.println(oneStorageArray.getName());
            }

        } finally {
            // house cleanup, close session
            if (apiEndPoint != null && apiSessionContext != null) {
                apiEndPoint.closeSession(apiSessionContext);
            }
        }
    }
}

```

2. Return to your command line, rebuild, and execute your program by entering the following command:

```
ant clean build run
```

This time the program is extracting information from your environment. You should see a printout of all the names of storage arrays found in your system.

Result

You have completed your first information extraction from the system.

Extracting data with paged results

In this section, you learn how to build upon the previous data extraction example to extract information from larger environments. In particular, you learn how to extract data from the system incrementally using the concept of paging.

About this task

Paging becomes extremely important in cases where the results being retrieved from the OnCommand Insight server are too large to handle all in memory at the same time. In these cases, the processing of the results needs to be broken up into several pages of results allowing each page to be handled within your program's limits.

This section of code applies the following points:

- When opening a session, the code is now changing the second parameter of the call. This parameter sets the maximum number of results you will get back from any request to the server. In the initial example, this was set to 10,000, but the example in this topic limits the result to three.
- The handling of the results is now enclosed in a do-while loop that ensures you have access to all pages of the results. As the code loops through the request, the code accesses the latest request iterator information to pass along to a call for the next page of results.

Step

1. Using your preferred text editor, edit the `src\OCI_HelloWorld.java` code by entering the following:

```
import com.onaro.sanscreen.wsapi.APISessionContext;
import com.onaro.sanscreen.wsapi.RequestIterator;
import com.onaro.sanscreen.wsapi.SANscreenAPIEndPoint;
import com.onaro.sanscreen.wsapi.StorageArray;
import com.onaro.sanscreen.wsapi.StorageArrayResponse;
import com.onaro.sanscreen.wsapi.utils.SANscreenAPIFactory;

/**
 * OnCommand Insight API Hello World:
 * Retrieving names of all storage arrays in the remote system
 *
 * @author Netapp
 * @since March 7, 2013
 */
public class OCI_HelloWorld {

    public static void main(String[] args) throws Exception {
        SANscreenAPIEndPoint apiEndPoint = null;
        APISessionContext apiSessionContext = null;
        try {
            // create API end point for communication
            apiEndPoint = SANscreenAPIFactory.getEndPoint(
                ("YOURHOSTNAME", "YOURUSERNAME", "YOURUSERPASSWORD"));
            // create interaction session context for communication, get back 3 results
            // max at a time
            apiSessionContext =
                SANscreenAPIFactory.openCurrentTimeSessionContext(apiEndPoint, 3);

            // call for retrieval of all storage arrays in remote system
            StorageArrayResponse response;
            // iterator to use when retrieving objects
            RequestIterator iterator = null;
            do {
                response = apiEndPoint.getStorageArrays(apiSessionContext, iterator);
                // loop through response, print each storage array name
                for (StorageArray oneStorageArray : response.getArrays()) {
                    System.out.println(oneStorageArray.getName());
                }
                // retrieve iterator from response
                iterator = response.getRequestIterator();

                // some feedback, retrieving more objects from system
                System.out.println("Got 3 results, retrieving more...");
            } while (iterator.isHasMore());
        } finally {
            // house cleanup, close session
            if (apiEndPoint != null && apiSessionContext != null) {
                apiEndPoint.closeSession(apiSessionContext);
            }
        }
    }
}
```

Result

You have completed your first paged information extraction from the system.

What's next in OnCommand Insight Connect API projects?

Using the `OCI_HelloWorld` example as guidance, you can build new programs to interact with and extract information from OnCommand Insight.

You can consult the endpoint documentation to see which other interaction and extraction methods are available for OnCommand Insight information extraction.

Connect API endpoint methods

You can use OnCommand Insight Connect API endpoint methods in your programs.

addHostToApplication (deprecated)

This method associates a host with an application. This method is deprecated; instead, use `addHostToApplicationById`.

```
public boolean addHostToApplication (APISessionContext sessionContext,  
String hostIp, String hostName, String applicationId)
```

Parameters

This method has the following parameters.

sessionContext

Mandatory. The session identifies the snapshot of the data to be used, thus the response will show the hosts that were known at the time the session was created.

hostIp

IP of the host to be associated with the application.

hostName

Name of the host to be associated with the application.

applicationId

ID of the application to associate with the host.

Returns

True upon successful completion of the operation.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Deprecated since 6.x.

addHostToApplicationById

Associates a host with an application using a host ID.

```
public boolean addHostToApplicationById(
    APISessionContext sessionContext,
    String hostId,
    String applicationName)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

hostId

ID of the host to be associated with the application.

applicationName

Name of the application to associate with the host.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

addReservationRequests

This method adds new request objects for Insight Plan, the capacity management product.

```
public String[] addReservationRequests (
    APISessionContext sessionContext,
    String ReservationRequest[] states)
```

Parameters

This method has the following parameters.

sessionContext

Requests are not historical data.

states

States of the reservation.

Returns

Returns array of IDs of created reservation requests.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

addReservationRequirements

This method adds the requirement objects of the request for Insight Plan, the capacity management product.

```
public ReservationRequests[] addReservationRequirements (
    APISessionContext sessionContext,
    String requestId, ReservationRequirement[] requirements)
```

Parameters

This method has the following parameters.

sessionContext

The session context to use.

requestId

The ID of request for which requirements will be added.

requirements

The array of requirements to add to the request.

Returns

True upon successful completion of operation.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

addVmToApplication

Associates a virtual machine with an application.

```
public boolean addVmToApplication(  
    APISessionContext sessionContext,  
    String vmId,  
    String applicationName)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

vmId

ID of the virtual machine.

applicationName

Name of the application to associate with the virtual machine.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

closeSession

Closes an API session, discarding any resources being used by the server.

```
public boolean closeSession(APISessionContext sessionContext)
```

Parameter

This method has the following parameter.

sessionContext

Identifies the session to close. The session context is returned by the openSession method.

Returns

True upon successful completion of operation.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred. Closes an API session, discarding any resources being used by the server.

New Since Version

Before 6.0

Related references

[openSession](#) on page 94

dismissPathOutageViolations

This method dismisses path outage violations.

```
public boolean dismissPathOutageViolations(
    APISessionContext sessionContext,
    String [] violationIds)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

violationIds

IDs of path outage violations to be dismissed.

Returns

True upon successful completion of operation.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or the iterator requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

6.1

findObjectByWwn

This method finds SAN devices based on their WWNs. A device can be a switch or WWN of the host adapter, a WWN of the port of the host adapter, a WWN of the protocol controller, or a WWN of the port of the protocol controller. Fabrics are excluded from search.

```
public DeviceBase findObjectByWwn (
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String wwn)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent nodes for a query that exceeded the session limit. If null, begins a new query.

wwn

WWN value to use in search

Returns

The object matching the specified WWN if found. Null otherwise.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or the iterator requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

findObjectsByIp

This method finds SAN devices based on their IP address/DNS name, which is regular expression.

```
public DeviceResponse findObjectsByIp (
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String ip)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent nodes for a query that exceeded the session limit. If null, begins a new query.

ip

Regular expression for device IP.

Returns

A device response object. The response includes an array of devices up to the session limit together with an RequestIterator which is used to fetch the rest of the nodes by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

getAllApplications

This method retrieves all the applications known to OnCommand Insight. Each application contains a list of the hosts that are associated with it. Optional attributes describing the application, such as priority, may also be included.

```
public ApplicationResponse getAllApplications (
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent nodes for a query that exceeded the session limit, if null, begins a new query.

Returns

Response object for applications. The response includes an array of Applications up to the session limit together with an RequestIterator which is used to fetch the rest of the nodes by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Application](#) on page 126

getAPIVersion

This method gets the version of the API deployed in the OnCommand Insight server to which the client is connected.

```
public String getAPIVersion()
```

Returns

The version of API deployed in the OnCommand Insight server.

Throws

This method throws the following exceptions.

java.rmi.RemoteException

Sent if there was a failure connecting to the server.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

Before 6.0

getApplication

This method retrieves a single application based on its name. The application contains a list of the hosts that are associated with it. Optional attributes describing the application, such as priority, may also be included.

```
public ApplicationResponse getApplication (
    APISessionContext sessionContext,
    String applicationName)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

applicationName

Unique name of the application.

Returns

The application object matching the specified application name.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Application](#) on page 126

getApplicationOfInternalVolume

Retrieves the application that is associated with a given internal volume. The application contains a list of the objects (such as hosts, virtual machines, and volumes) that are associated with it. Optional attributes describing the application, such as priority, can be included, too.

```
public Application getApplicationOfInternalVolume(
    APISessionContext sessionContext,
    String internalVolumeId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

internalVolumeId

ID of the internal volume.

Returns

A response object for an application associated with the internal volume. If an application is associated with the internal volume, it will be returned. Otherwise, null is returned.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

Related references

[Application](#) on page 126

getApplicationOfQtree

Retrieves the application that is associated with a given qtree. The application contains a list of the objects (such as hosts, virtual machines, and volumes) that are associated with it. Optional attributes describing the application, such as priority, can be included, too.

```
public Application getApplicationOfQtree(
    APISessionContext sessionContext,
    String qtreeId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

qtreeId

ID of the qtree.

Returns

A response object for an application associated with the qtree. If an application is associated with the qtree, it will be returned. Otherwise, null is returned.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

Related references

[Application](#) on page 126

getApplicationOfShare

Retrieves the application that is associated with a given share. The application contains a list of the objects (such as hosts, virtual machines, and volumes) that are associated with it. Optional attributes describing the application, such as priority, can be included, too.

```
public Application getApplicationOfShare(
    APISessionContext sessionContext,
    String shareId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

shareId

ID of the share.

Returns

A response object for an application associated with the share. If an application is associated with the share, it will be returned. Otherwise, null is returned.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

Related references

[Application](#) on page 126

getApplicationOfVolume

Retrieves the application that is associated with a given volume. The application contains a list of the objects (such as hosts, virtual machines, and volumes) that are associated with it. Optional attributes describing the application, such as priority, can be included, too.

```
public Application getApplicationOfVolume(  
    APISessionContext sessionContext,  
    String volumeId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

volumeId

ID of a volume.

Returns

A response object for an application associated with a volume. If an application is associated with the volume, it will be returned. Otherwise, null will be returned.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

Related references

[Application](#) on page 126

getApplicationsOfHost

This method retrieves all the applications that are associated with a given host. Each application contains a list of the hosts that are associated with it. Optional attributes describing the application, such as priority, may also be included.

```
public ApplicationResponse getApplicationsOfHost(  
    APISessionContext sessionContext,  
    RequestIterator requestIterator,  
    String hostId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent nodes for a query that exceeded the session limit. If null, begins a new query.

hostId

ID of host.

Returns

Response object for applications. The object includes an array of applications that are associated with the given host, up to the session limit together with an RequestIterator, which is used to fetch the rest of the nodes by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

Related references

[Application](#) on page 126

getApplicationsOfVm

Retrieves all the applications that are associated with a given virtual machine. Each application contains a list of the objects (such as hosts, virtual machines, and volumes) that are associated with it. Optional attributes describing the application, such as priority, can be included, too.

```
public ApplicationResponse getApplicationsOfVm(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String vmId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent nodes for a query that exceeded the session limit. If null, begins a new query.

vmId

ID of a virtual machine.

Returns

A response object for applications associated with a virtual machine. The object includes an array of applications that are associated with the given virtual machine, up to the session limit together with a `RequestIterator`, which is used to fetch the rest of the nodes by subsequent invocations of this method.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

Related references

[Application](#) on page 126

getBusinessEntities

Finds all the business entities.

```
public BusinessEntityResponse getBusinessEntities(
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent nodes for a query that exceeded the session limit. If null, begins a new query.

Returns

A response object for business entities. This object includes an array of `BusinessEntity` up to the session limit together with a `RequestIterator`, which is used to fetch the rest of the nodes by subsequent invocations of this method.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

Related references[BusinessEntity](#) on page 132

getBusinessEntity

Finds the business entity with the given ID.

```
public BusinessEntity getBusinessEntity(
    APISessionContext sessionContext,
    String businessEntityId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

businessEntityId

ID of the business entity.

Returns

A response object for business entities. This object includes an array of BusinessEntity up to the session limit together with RequestIterator.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

getBusinessEntityOfPort

Gets the business entity of a port. Returns null if no business entity is set on the port.

```
public BusinessEntity getBusinessEntityOfPort(
    APISessionContext sessionContext,
    String portId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

portId

ID of the port.

Returns

A response object for a business entity associated with a port. Returns null if no business entity is set on the port.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

getBusinessEntityOfStorage

Sets or unsets the business entity of a switch. To unset the existing value, pass an empty string ("") for businessEntityId.

```
public BusinessEntity getBusinessEntityOfStorage(
    APISessionContext sessionContext,
    String storageId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

storageId

ID of the storage system.

Returns

A response object for business entity. Returns null if no business entity is set on the storage.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

getBusinessEntityOfSwitch

Gets the business entity of a switch. Returns null if no business entity is set on the switch.

```
public BusinessEntity getBusinessEntityOfSwitch(
    APISessionContext sessionContext,
    String switchId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

switchId

ID of the switch.

Returns

A response object for business entity. Returns null if no business entity is set on the switch.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

getClosedViolations

This method retrieves all the closed violations in the SAN starting from a specified time.

```
public ViolationResponse getClosedViolations(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    long time)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent nodes for a query that exceeded the session limit. If null, begins a new query.

time

Starting time from which closed violations are retrieved.

Returns

A response object for closed violations. The object includes an array of closed violations starting from a specified time up to the session limit together with a `RequestIterator` which is used to fetch the rest of the closed violations by subsequent invocations of this method.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.0

Related references

[Violation](#) on page 198

getConnectedPorts

Retrieves all connected logical ports associated with the specified session context. You can use `LogicalPort` class methods to access logical port data, such as physical port ID and status.

```
public LogicalPortResponse getConnectedPorts(
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent ports for a query that exceeded the session limit. If null, begins a new query.

Returns

Response object for ports. The object includes an array of logical ports, up to the session limit, together with a `RequestIterator`, used to fetch the rest of the logical ports by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

getDeviceGroup

This method retrieves a Symmetrix device group by its ID.

```
public DeviceGroup getDeviceGroup(
    APISessionContext sessionContext,
    String deviceGroupId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

deviceGroupId

ID of the device group.

Returns

Device group object or null if none was found.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[DeviceGroup](#) on page 134

getDeviceGroups

This method retrieves all the Symmetrix device groups.

```
public DeviceGroupResponse getDeviceGroups(  
    APISessionContext sessionContext,  
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent device groups for a query that exceeded the session limit. If null, begins a new query.

Returns

Response object for device groups. The object includes an array of DeviceGroups up to the session limit together with an RequestIterator, which is used to fetch the rest of the device groups by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[DeviceGroup](#) on page 134

getDeviceGroupsByStorage

This method retrieves all the Symmetrix device groups that have volumes for given storage.

```
public DeviceGroupResponse getDeviceGroupsByStorage(  
    APISessionContext sessionContext,  
    RequestIterator requestIterator,  
    String storageId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent nodes for a query that exceeded the session limit. If null, begins a new query.

storageId

ID of the storage associated with the device groups.

Returns

Response object for device groups. The response includes an array of DeviceGroups up to the session limit together with an RequestIterator, which is used to fetch the rest of the device groups by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[DeviceGroup](#) on page 134

getDeviceGroupsByVolume

This method retrieves all the Symmetrix device groups included in a volume.

```
public DeviceGroupResponse getDeviceGroupsByVolume(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String volumeId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent device groups for a query that exceeded the session limit. If null, begins a new query.

volumeId

ID of the volume for which to get device groups.

Returns

Response object for device groups. The object includes an array of DeviceGroups up to the session limit together with an RequestIterator, which is used to fetch the rest of the device groups by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[DeviceGroup](#) on page 134

getDeviceGroupVolumes

This method retrieves all the volumes that are part of Symmetrix device groups.

```
public VolumeResponse getDeviceGroupVolumes (
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String deviceGroupId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent volumes for a query that exceeded the session limit. If null, begins a new query.

deviceGroupId

ID of device group.

Returns

Response object for volumes. The object includes an array of volumes up to the session limit together with an `RequestIterator`, which is used to fetch the rest of the volumes by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

Related references

[DeviceGroup](#) on page 134

[Volume](#) on page 205

getDRPathsByVolume

This method retrieves all the DR (disaster recovery) paths detected by OnCommand Insight for a given volume. Volume ID can refer to any volume from DRPath for DRPath to be returned in this call.

```
public DRPathResponse getDRPathsByVolume(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String volumeId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent DR-paths for a query that exceeded the session limit. If null, begins a new query.

volumeId

ID of the volume for which to return DR paths.

Returns

Response object for DRPaths. The response includes an array of DRPaths up to the session limit together with an `RequestIterator`, which is used to fetch the rest of the DR-path by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

Related references

[DRPath](#) on page 136

getFabricsByPorts

This method retrieves fabrics by port IDs detected by OnCommand Insight.

```
public PortsByFabricResponse getFabricsByPorts
(APISessionContext sessionContext, String[] portIds)
throws APIException, RemoteException
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

portIds

An array of IDs of the ports for which to get fabric. Cannot be null or empty string.

Returns

PortsByFabric Response, which includes an array of PortsByFabric response object.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error or invalid parameters being passed into the method.

New Since Version

6.4

Related references

[Port](#) on page 159

getGeneric

This method retrieves a generic device. A generic device is a SAN device that is not yet identified.

```
public Generic getGeneric(
    APISessionContext sessionContext,
    String id)
```


Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

id

Unique identifier for the generic device.

Returns

Response object for generic device. A generic device if it exists; otherwise, NULL.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

getGenerics

This method retrieves all generic devices. Generic devices are those that are known to OnCommand Insight (including inactive ones), but not yet identified.

```
public GenericResponse getGenerics(
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent devices for a query that exceeded the session limit; if null, begins a new query. The session limit is set when a session is opened.

Returns

Response object for generic devices. The objects includes an array of generic devices up to the session limit together with an RequestIterator, which is used to fetch the rest of the devices by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

getHost

This method retrieves a host server.

```
public Host getHost(
    APISessionContext sessionContext,
    String hostId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

hostId

The ID of the host that needs to be fetched.

Returns

Response object for a host. A single instance of a host, if it exists. Otherwise, returns NULL.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

getHostPolicy

This method retrieves the policy defined for a host.

```
public Policy getHostPolicy(
    APISessionContext sessionContext,
    String hostId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

hostId

ID of host associated with the policy.

Returns

Returns a single host policy.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Policy](#) on page 157

getHosts

This method retrieves all the hosts or servers known to OnCommand Insight, including inactive hosts.

```
public HostResponse getHosts(
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent hosts for a query that exceeded the session limit. If null, begins a new query. The session limit is set when a session is first opened.

Returns

Response object for hosts. The object includes an array of hosts up to the session limit together with an RequestIterator, which is used to fetch the rest of the nodes by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Host](#) on page 142

getHostsOfApplication

This method retrieves all the host servers (see [Host](#)) that are associated with a given application.

```
public HostResponse getHostsOfApplication(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String applicationName)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent hosts for a query that exceeded the session limit. If null, begins a new query.

applicationName

Unique name of an application.

Returns

Response object for hosts. The object includes an array of hosts up to the session limit together with a RequestIterator, which is used to fetch the rest of the hosts by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

Related references

[Host](#) on page 142

getInternalVolume

This method retrieves an internal volume by ID.

```
public InternalVolume getInternalVolume(
    APISessionContext sessionContext,
    String id)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

Id

ID of the internal volume.

Returns

An internal volume, if it exists. Otherwise, null.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

6.0

Related references

[InternalVolume](#) on page 143

getInternalVolumes

This method retrieves all internal volumes known to OnCommand Insight.

```
public InternalVolumeResponse getInternalVolumes(
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent internal volumes for a query that exceeded the session limit. If null, begins a new query.

Returns

Response object for internal volumes. The object includes an array of internal volumes up to the session limit together with a RequestIterator, which is used to fetch the rest of the internal volumes by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

6.0

Related references

[InternalVolume](#) on page 143

getInternalVolumesByStorageArray

This method retrieves internal volumes for a given storage array.

```
public InternalVolumeResponse getInternalVolumesByStorageArray(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String storageArrayId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent internal nodes for a query that exceeded the session limit. If null, begins a new query.

storageArrayId

Unique ID of the Storage Array for which internal volumes are requested. Use `getStorageArrays(APISessionContext,RequestIterator)` to get a list of all the arrays.

Returns

Response object for internal volumes. The object includes an array of internal volumes up to the session limit together with a RequestIterator, which is used to fetch the rest of the internal volumes by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

6.0

Related references

[InternalVolume](#) on page 143

getLogicalPortsByLogicalSwitch

This method retrieves the logical ports associated with the specified switch. You can use the LogicalPort class methods to access logical port data, such as physical port ID and status.

```
public LogicalPortResponse getLogicalPortsByLogicalSwitch(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String logicalSwitchId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent ports for a query that exceeded the session limit. If null, begins a new query.

logicalSwitchId

Uniquely identifies the switch whose logical ports are requested.

Returns

Response object for logical ports. The object includes an array of ports, up to the session limit, together with a RequestIterator, which is used to fetch the rest of the ports by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[LogicalPort](#) on page 149

getLogicalPortsByPhysicalPort

This method retrieves all logical ports associated with the specified physical port. You can use LogicalPort class methods to access logical port data, such as physical port ID and status.

```
public LogicalPortResponse getLogicalPortsByPhysicalPort (
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String physicalPortId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent ports for a query that exceeded the session limit. If null, begins a new query.

physicalPortID

Uniquely identifies the physical port whose logical ports are requested.

Returns

Response object for logical ports. The object includes an array of logical ports, up to the session limit, together with a RequestIterator, used to fetch the rest of the logical ports by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[LogicalPort](#) on page 149

getLogicalPortsByPhysicalSwitch

This method retrieves the logical ports associated with the specified switch. You can use the LogicalPort class methods to access logical port data, such as physical port ID and status.

```
public LogicalPortResponse getLogicalPortsByPhysicalSwitch (
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String switchId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent ports for a query that exceeded the session limit. If null, begins a new query.

switchId

Uniquely identifies the switch for which logical ports are requested.

Returns

Response object for logical ports. The object includes an array of ports, up to the session limit together with an RequestIterator, which is used to fetch the rest of the ports by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

getLogicalSwitchesByPhysicalSwitch

This method retrieves the logical switches associated with the specified physical switch. You can use Switch class methods to access switch data, such as WWN and status.

```
public SwitchResponse getLogicalSwitchesByPhysicalSwitch(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String switchId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent switches for a query that exceeded the session limit. If null, begins a new query.

switchId

Uniquely identifies the physical switch whose logical switches are requested.

Returns

Response object for logical switches. The response includes an array of switches, up to the session limit, together with a RequestIterator, which is used to fetch the rest of the switches by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Switch](#) on page 189

getNodesOfDevice

This method retrieves the nodes of a device, such as a host's HBA or the controller of a storage-device. You can use Node class methods to access node data, such as WWN and device ID.

```
public NodeResponse getNodesOfDevice(
    APISessionContext sessionContext,
```

```
RequestIterator requestIterator,  
String deviceId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent nodes for a query that exceeded the session limit. If null, begins a new query.

deviceId

Uniquely identifies the device whose nodes are requested.

Returns

Response object for nodes. The response includes an array of nodes, up to the session limit, together with a RequestIterator, used to fetch the rest of the nodes by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Node](#) on page 150

getPathPolicy

This method retrieves the policy defined for path.

```
public Policy getPathPolicy(  
    APISessionContext sessionContext,  
    Path path)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

path

The logical path for which policy will be returned.

Returns

Response object for policy. Policy for the path.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Policy](#) on page 157

getPaths

This method retrieves all the logical paths detected by OnCommand Insight in the SAN being monitored. You can use Path class methods to access path data, such as host ID and array ID.

```
public PathResponse getPaths(
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent logical paths for a query that exceeded the session limit. If null, begins a new query.

Returns

Response object for paths. The object includes an array of paths, up to the session limit, together with a RequestIterator, which is used to fetch the rest of the logical path by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Path](#) on page 153

getPathsByHost

This method retrieves all the logical paths detected by OnCommand Insight in the SAN being monitored for a given host.

```
public PathResponse getPathsByHost(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String hostId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent logical paths for a query that exceeded the session limit. If null, begins a new query.

hostId

ID of the host.

Returns

Response object for paths. The object includes an array of paths, up to the session limit together with an RequestIterator, which is used to fetch the rest of the logical paths by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Path](#) on page 153

getPendingPathReservations

This method retrieves all pending path reservations detected by OnCommand Insight.

```
public PathReservationResponse getPendingPathReservations
(APISessionContext sessionContext, RequestIterator iterator)
throws APIException, RemoteException
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent pending path reservations for a query that exceeded the session limit. If null, begins a new query.

Returns

Response object for pathReservations. The object includes 1) an array of pathReservation objects limited to the session limit and 2) a RequestIterator, which is used to fetch the rest of the logical paths by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used).

New Since Version

6.4

getPhysicalPortByLogicalPort

This method retrieves the physical port associated with the specified logical port. You can use Port class methods to access physical port data, such as WWN and status.

```
public PortResponse getPhysicalPortByLogicalPort(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String logicalPortId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent ports for a query that exceeded the session limit. If null, begins a new query.

logicalPortId

Uniquely identifies the logical port whose physical port is requested.

Returns

Response object for physical port. The object includes an array of ports, up to the session limit, together with a RequestIterator, used to fetch the rest of the ports by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Port](#) on page 159

getPolicies

This method retrieves all the policies defined for the SAN.

```
public PolicyResponse getPolicies(
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent policies for a query that exceeded the session limit. If null, begins a new query.

Returns

Response object for policies. The object includes an array of policies up to the session limit together with an `RequestIterator`, which is used to fetch the rest of the policies by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Policy](#) on page 157

getPort

This method retrieves a port. You can use Port methods to access port data, such as WWN and status.

```
public Port getPort (
    APISessionContext sessionContext,
    String id)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

id

Unique identifier for the port.

Returns

Response object for port. The port, if it exists; otherwise Null.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Port](#) on page 159

getPorts

This method retrieves all the ports in the SAN. You can use Port methods to access port data, such as WWN and status.

```
public PortResponse getPorts (
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent ports for a query that exceeded the session limit. If null, begins a new query.

Returns

Response object for ports. The object includes an array of ports, up to the session limit, together with a RequestIterator, which is used to fetch the rest of the ports by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Port](#) on page 159

getPortsOfDevice

This method retrieves all the ports of a device in all the nodes of the device. You can use Port methods to access port data, such as WWN and status.

```
public PortResponse getPortsOfDevice (
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String deviceId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent ports for a query that exceeded the session limit. If null, begins a new query.

deviceId

Uniquely identifies the device whose ports are requested.

Returns

Response object of ports. The object includes an array of ports, up to the session limit, together with a RequestIterator, which is used to fetch the rest of the ports by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Port](#) on page 159

getPortsOfDeviceType

This method retrieves ports on all the devices of the specified type. You can use Port methods to access port data, such as WWN and status.

```
public PortResponse getPortsOfDeviceType (
    APISessionContext sessionContext,
```

```
RequestIterator requestIterator,
String type)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent ports for a query that exceeded the session limit. If null, begins a new query.

type

Type of the device.

Returns

Response object of ports. The object includes an array of ports, up to the session limit, together with a RequestIterator, which is used to fetch the rest of the ports by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Port](#) on page 159

getPortsOfNode

This method retrieves all the ports in a single node of a device. You can use Port methods to access port data, such as WWN and status.

```
public PortResponse getPortsOfNode (
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String nodeId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent ports for a query that exceeded the session limit. If null, begins a new query.

nodeId

Uniquely identifies the node and device whose ports are requested.

Returns

Response object of ports. The object includes an array of ports, up to the session limit together with an `RequestIterator`, which is used to fetch the rest of the ports by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Port](#) on page 159

getQtrees

Retrieves all the qtrees.

```
public QtreeResponse getQtrees(
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent volumes for a query that exceeded the session limit. If null, begins a new query.

Returns

Response object for qtrees. This object includes an array of qtrees up to the session limit together with a RequestIterator, which is used to fetch the rest of the qtrees by subsequent invocations of this method.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

Related references

[Qtree](#) on page 165

getRecentDevice

This method gets the most recent SAN device in the history.

```
public DeviceBase getRecentDevice (
    APISessionContext sessionContext,
    String id)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

id

Device ID.

Returns

Response object for devices. The response returns the most recent device in the history.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

getRecentHost

This method retrieves the most recent host server in history.

```
public Host getRecentHost(
    APISessionContext sessionContext,
    String hostId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the hosts that were known at the time the session was created.

hostId

The ID of the host that needs to be fetched.

Returns

A single instance of host, if one exists. Null otherwise.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Host](#) on page 142

getRecentStorageArray

This method retrieves the most recent storage array in history.

```
public StorageArray getRecentStorageArray (
    APISessionContext sessionContext,
    String id)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

id

ID of storage array.

Returns

Response object for storage arrays. The object includes an array for StorageArray if it exists. Null otherwise.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[StorageArray](#) on page 180

getRecentSwitch

This method retrieves the most recent switch device in history.

```
public Switch getRecentSwitch (
    APISessionContext sessionContext,
    String id)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

id

ID of switch.

Returns

Response object for switches, if one exists. Null otherwise.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Switch](#) on page 189

getRecentTape

This method retrieves the most recent tape device.

```
public Tape getRecentTape(
    APISessionContext sessionContext,
    String id)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

id

ID of tape.

Returns

Response object of tape if one exists. Null Otherwise.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Tape](#) on page 192

getRecentVolume

This method retrieves the volume that is most recent in history.

```
public Volume getRecentVolume (
    APISessionContext sessionContext,
    String id)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

id

ID of the volume.

Returns

Response object of a volume, if one exists. Null otherwise.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Volume](#) on page 205

getReservationRequests

This method gets all request objects for capacity management product.

```
public ReservationRequest[] getReservationRequests (
    APISessionContext sessionContext)
```

Parameters

This method has the following parameter.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

Returns

An array of reservation requests.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[ReservationRequest](#) on page 169

getReservationRequestsByStates

This method gets request objects from Insight Plan, the capacity management product, and filters them by states.

```
public ReservationRequest[] getReservationRequestsByStates(
    APISessionContext sessionContext,
    String[] states)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

States

Request states to get requests for: NEW, OPEN, CANCELLED, REJECTED, etc.

Returns

Array of reservation requests.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[ReservationRequest](#) on page 169

getReservationRequirement

This method gets a storage requirement object with the given ID.

```
public ReservationRequirement[] getReservationRequirement
(APISessionContext sessionContext, String[] requirementId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created. This parameter is not essential, because requests are not historical.

requirementId

Unique ID of the requirement.

Returns

Reservation requirement, if it exists; otherwise null.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.4

getReservationRequirements

This method gets all requirement objects of the request for Insight Plan, the capacity management product.

```
public ReservationRequirements[] getReservationRequirements (
    APISessionContext sessionContext,
    String[] requestId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestId

Unique ID of the reservation request to get reservation requirements for.

Returns

Array of reservation requirements.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[ReservationRequirement](#) on page 174

getReservationViolationTypeNames

This method retrieves a list of possible reservation violation types.

```
public String[] getReservationViolationTypeNames
(APISessionContext sessionContext)
```

Parameter

This method has the following parameter.

sessionContext

Not applicable for this request.

Returns

An array of violation types.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.4

getSANscreenVersion

This method gets the version of the OnCommand Insight server to which the client is connected. Use `setSessionContext` to retrieve the version number from the server before calling this method.

```
public String getSANscreenVersion()
```

Returns

The version of the OnCommand Insight server.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

getShares

Retrieves all the shares.

```
public ShareResponse getShares(
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent shares for a query that exceeded the session limit. If null, begins a new query.

Returns

A response object for shares. This object includes an array of Shares up to the session limit together with a RequestIterator, which is used to fetch the rest of the shares by subsequent invocations of this method.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

Related references

[Share](#) on page 179

getStorageArray

This method retrieves a storage array. After you retrieve a storage array, you can use `StorageArray` methods to access array data such as vendor name, model name, and raw capacity.

```
public StorageArray getStorageArray(
    APISessionContext sessionContext,
    String id)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

id

Unique identifier for the storage array.

Returns

Response object for storage array, if it exists. Otherwise, `NULL`.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[StorageArray](#) on page 180

getStorageArrays

This method retrieves all the storage arrays, including inactive ones, known to OnCommand Insight. After you retrieve a storage array, you can use `StorageArray` methods to access array data such as vendor name, model name, and raw capacity.

```
public StorageArrayResponse getStorageArrays(
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent arrays for a query that exceeded the session limit. If null, begins a new query.

Returns

Response object of storage arrays. The response includes an array of `StorageArrays`, up to the session limit, together with a `RequestIterator`, which is used to fetch the rest of the arrays by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[StorageArray](#) on page 180

getSwitch

This method retrieves a switch device.

```
public Switch getSwitch (
    APISessionContext sessionContext,
    String id)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

id

Unique identifier for the switch.

Returns

Response object for switch, if it exists. Otherwise, NULL.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Switch](#) on page 189

getSwitches

This method retrieves all the switch devices, including inactive ones, that are known to OnCommand Insight.

```
public SwitchResponse getSwitches (
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent devices for a query that exceeded the session limit. If null, begins a new query. The session limit is set when a session is opened.

Returns

Response object of switches. The object includes an array of switches, up to the session limit, together with a RequestIterator, which is used to fetch the rest of the switches by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Switch](#) on page 189

getTape

This method retrieves a tape device. You can use Tape methods to access tape device data, such as vendor name and model name.

```
public Tape getTape (
    APISessionContext sessionContext,
    String id)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

id

Unique identifier for the tape.

Returns

Response object for tape, if it exists. Otherwise, NULL.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Tape](#) on page 192

getTapes

This method retrieves all the tape devices, including inactive ones, known to OnCommand Insight. You can use Tape methods to access tape data, such as vendor name and model name.

```
public TapeResponse getTapes (
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent devices for a query that exceeded the session limit. If null, begins a new query.

Returns

Response object for tapes. The object includes an array of type Tape, up to the session limit, together with a RequestIterator, which is used to fetch the rest of the tapes by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Tape](#) on page 192

getViolations

This method retrieves all the violations in the SAN.

```
public ViolationResponse getViolations(
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent violations for a query that exceeded the session limit. If null, begins a new query.

Returns

Response object for violations. The object includes an array of violations up to the session limit together with a RequestIterator, which is used to fetch the rest of the violations by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Violation](#) on page 198

getViolationsByHost

Retrieves all the violations for given host.

```
public ViolationResponse getViolationsByHost(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String hostId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent violations for a query that exceeded the session limit. If null, begins a new query.

hostId

ID of the host that has the violations.

Returns

Response object for violations. The object includes an array of violations up to the session limit together with an RequestIterator, which is used to fetch the rest of the violations by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Violation](#) on page 198

getViolationsByRequests

This method retrieves violations using request IDs detected by OnCommand Insight.

```
public ReservationViolationsByRequestResponse getViolationsByRequests
(APISessionContext sessionContext, RequestIterator iterator, String[] requestIds)
throws APIException, RemoteException
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent violations for a query that exceeded the session limit. If null, begins a new query.

requestIds

An array of unique IDs of the reservation requests to get reservation violations for.

Returns

Response object for ReservationViolations. The object includes 1) an array of ReservationViolationByRequest objects limited to the session limit and 2) a RequestIterator, which is used to fetch the rest of the logical paths by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used).

New Since Version

6.4

Related references

[Violation](#) on page 198

getViolationsByReservations

This method retrieves all the violations for given reservation.

```
public ReservationViolationsByReservationResponse[]
getViolationsByReservations
(APISessionContext sessionContext, RequestIterator requestIterator,
String[] reservationIds)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created. This parameter is not essential, because requests are not historical.

requestIterator

Optional. If specified, used to fetch subsequent violations for a query that exceeded the session limit. If null, begins a new query.

reservationIds

An array of non-null reservationIds to get violations for.

Returns

Response object of violations. The response includes an array of Violations, up to the session limit, together with a RequestIterator, which is used to retrieve the rest of the violations by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.4

getVirtualMachine

This method gets virtual machine (VM) data by ID.

```
public VirtualMachine getVirtualMachine (
    APISessionContext sessionContext,
    String id)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

id

ID of the virtual machine.

Returns

Response object for a virtual machine. If not found, returns null.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

6.0

Related references

[VirtualMachine](#) on page 201

getVirtualMachines

This method retrieves all virtual machines.

```
public VirtualMachineResponse getVirtualMachines (
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent virtual machines for a query that exceeded the session limit. If null, begins a new query.

Returns

Response object of virtual machines. An array of virtual machines up to the session limit together with a RequestIterator, which is used to fetch the rest of the virtual machines by subsequent invocations of this method.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[VirtualMachine](#) on page 201

getVolume

Retrieves a volume. You can use Volume class methods to access volume data, such as disk type and capacity.

```
public Volume getVolume (
    APISessionContext sessionContext,
    String id)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

id

Unique identifier for the volume.

Returns

Response object of a volume, if it exists. Otherwise, NULL.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Volume](#) on page 205

getVolumeMapsByStorageArray

Retrieves all the volume maps known to OnCommand Insight on the specified storage array. You can use VolumeMap class methods to access volume map data, such as storage port ID and protocol controller.

```
public VolumeMapResponse getVolumeMapsByStorageArray(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String storageArrayId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent volume maps for a query that exceeded the session limit. If null, begins a new query.

storageArrayId

Unique name of the storage array whose volume maps are requested. You can get a list of storage arrays by using the `getStorageArrays` method.

Returns

Response object for volume maps. The response includes an array of volume maps, up to the session limit, together with a RequestIterator, used to fetch the rest of the volume maps by subsequent invocations of this method. The session limit is set at the time a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[VolumeMap](#) on page 208

getVolumeMasksByStorageArray

Retrieves all the volume masks known to OnCommand Insight on the specified storage array. You can use VolumeMask class methods to access volume-mask data, such as storage port WWN and ID.

```
public VolumeMaskResponse getVolumeMasksByStorageArray (
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String storageArrayId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent volume masks for a query that exceeded the session limit. If null, begins a new query.

storageArrayId

Uniquely identifies the storage array whose volume masks are requested. You can get a list of storage arrays by using the getStorageArrays method.

Returns

Response object of volume masks. The object includes an array of volume masks, up to the session limit, together with a RequestIterator, used to retrieve the rest of the volume masks by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

Related references

[VolumeMask](#) on page 211

getVolumes

Retrieves all the volumes in the SAN. You can use Volume class methods to access volume data, such as disk type and capacity.

```
public VolumeResponse getVolumes(
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent volumes for a query that exceeded the session limit. If null, begins a new query.

Returns

Response object of volumes. The object includes an array of volumes, up to the session limit, together with a RequestIterator, which is used to fetch the rest of the volumes by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Volume](#) on page 205

getVolumesByStorageArray

Retrieves all volumes known to OnCommand Insight on the specified storage array. You can use Volume class methods to access volume data, such as disk type and capacity.

```
public VolumeResponse getVolumesByStorageArray (
    APISessionContext sessionContext,
```

```
RequestIterator requestIterator,
String storageArrayId)
```

Parameters

This method has the following parameters.

sessionContext

The session context returned by the openSession method. The context corresponds to the Snapshot copy of data to be used, thus responses can contain only those volumes known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent volumes for a query that exceeded the session limit. If null, begins a new query.

storageArrayId

Uniquely identifies the storage array whose volumes are requested. You can get a list of storage arrays by using the getStorageArrays method.

Returns

Response object for volumes. The object includes an array of volumes, up to the session limit, together with a RequestIterator, which is used to fetch the rest of the volumes by subsequent invocations of this method. The session limit is set when a session is opened.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Volume](#) on page 205

getVolumesSynchStatesByDRPath

Returns synchronization states for the volumes of the DR Path.

```
public VolumesSynchronizationStateResponse getVolumesSynchStatesByDRPath(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String drPathId,
    long period)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent nodes for a query that exceeded the session limit. If null, begins a new query.

drPathId

ID of the DR path.

period

Timestamp indicating when path existed.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[VolumesSynchronizationState](#) on page 214

getVolumesSynchStatesBySourceVolume

Returns all synchronization states for given source volume.

```
public VolumesSynchronizationStateResponse getVolumesSynchStatesBySourceVolume(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String volumeId,
    long period)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent nodes for a query that exceeded the session limit. If null, begins a new query.

volumeId

ID of the volume to get data for.

period

Timestamp indicating when synchronization existed.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[VolumesSynchronizationState](#) on page 214

getZonesByPorts

This method retrieves all the zones by port IDs detected by OnCommand Insight.

```
public PortsByZoneResponse getZonesByPorts
(APISessionContext sessionContext, String[] portIds)
    throws APIException, RemoteException
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

portIds

An array of IDs of the ports to get zones for. Cannot be null or empty string.

Returns

PortsByZoneResponseobject. The object contains an array of PortsByZone objects

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error or invalid parameters being passed into the method.

New Since Version

6.4

Related references

[Port](#) on page 159

openSession

Creates the `APISessionContext`, which is required by the API server in order to provide a consistent view of the data. Opening a session also defines the API version being used by the client as well as the maximum number of values returned in a single query.

```
public APISessionContext openSession (
    String clientVersion,
    int limit)
```

A session defines a Snapshot copy of the data at the time it was created. Therefore, queries that are issued using the same session will reflect the data at the time the session was created.

When you open a session, the system verifies whether you have a valid API license. If a valid API license is not found, an exception is thrown.

Parameters

This method has the following parameters.

clientVersion

The version of the API client. The server will verify that the API client version number equals the supported server version. If they do not match, an exception is thrown.

limit

Maximum number of values in a response to a query. A `RequestIterator` fetches values exceeding the limit. This limit cannot exceed 10,000.

Returns

A new session.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

removeHostFromApplication (deprecated)

Removes association between a host and an application. This method is deprecated; instead, use `removeHostFromApplicationById`.

```
public boolean removeHostFromApplication (APISessionContext sessionContext,
    String hostIp, String hostName, String applicationId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

hostIp

IP of the host whose association with the application should be removed.

hostName

Name of the host whose association with the application should be removed.

applicationId

ID of the application to associate with the host.

Returns

True upon successful completion of operation.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

removeHostFromApplicationById

Removes the association between a host and an application.

```
public boolean removeHostFromApplicationById(
    APISessionContext sessionContext,
    String hostId,
    String applicationName)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

hostId

ID of the host.

applicationName

Name of the application to disassociate from the host.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

removeHostPolicies

Removes policies for host including path policies and host policy.

```
public boolean removeHostPolicies (
    APISessionContext sessionContext,
    String hostId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

hostId

ID of the host for which policies will be removed.

Returns

True upon successful completion of operation.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Policy](#) on page 157

removeInternalVolumeFromApplication

Removes the association between an internal volume and an application.

```
public boolean removeInternalVolumeFromApplication(
    APISessionContext sessionContext,
    String internalVolumeId,
    String applicationName)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

internalVolumeId

ID of the internal volume.

applicationName

Name of the application associated with the internal volume.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

removePolicies

Removes policies.

```
public boolean removePolicies (
    APISessionContext sessionContext,
    String[] policyIds)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

policyIds

IDs of the policies to remove.

Returns

True upon successful completion of operation.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Policy](#) on page 157

removeQtreeFromApplication

Removes the association between a qtree and an application.

```
public boolean removeQtreeFromApplication(
    APISessionContext sessionContext,
    String qtreeId,
    String applicationName)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

qtreeId

ID of the qtree.

applicationName

Name of the application associated with the qtree.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

removeReservationRequirements

Removes existing requirement objects of the request for capacity management product.

```
public boolean removeReservationRequirements (
    APISessionContext sessionContext,
    String requestId,
    String[] requirementIds)
```

Parameters

This method has the following parameters.

sessionContext

Requests are not historical data.

requestId

ID of the request for which requirements will be removed.

requirementIds

IDs of the requirements to remove.

Returns

True upon successful completion of operation.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[ReservationRequest](#) on page 169

[ReservationRequirement](#) on page 174

removeShareFromApplication

Removes the association between a share and an application.

```
public boolean removeShareFromApplication(
    APISessionContext sessionContext,
    String shareId,
    String applicationName)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

shareId

ID of the share.

applicationName

Name of the application associated with the share.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

removeVmFromApplication

Removes association between a virtual machine and an application.

```
public boolean removeVmFromApplication(
    APISessionContext sessionContext,
    String vmId,
    String applicationName)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

vmId

ID of the virtual machine

applicationName

Name of the application to disassociate from the virtual machine.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

removeVolumeFromApplication

Removes the association between a volume and an application.

```
public boolean removeVolumeFromApplication(
    APISessionContext sessionContext,
    String volumeId,
    String applicationName)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

volumeId

ID of a volume.

applicationName

Name of the application associated with the volume.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

setAnnotationValues

This method sets annotation values.

```
public boolean setAnnotationValues(
    APISessionContext sessionContext,
    AnnotationValue[] annotationValues)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

annotationValues

Annotation values to set.

Returns

True upon successful completion of operation.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

Related references

[Annotation Value](#) on page 123

setApplication

Adds or updates attributes of an application.

```
public boolean setApplication (
    APISessionContext sessionContext,
    String name,
    String priority,
    String businessEntityId,
    boolean ignoreSharing)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

name

Unique name of an application.

priority

Priority of the application. can be one of the following values: Critical, High, Medium, Low.

businessEntityId

ID of the business entity with which this application is associated.

ignoreSharing

Hosts associated with this application will be ignored when calculating volume sharing violation for other hosts.

Returns

True upon successful completion of operation.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[Application](#) on page 126

setApplicationOfInternalVolume

Associates an internal volume with an application.

```
public boolean setApplicationOfInternalVolume(
    APISessionContext sessionContext,
    String internalVolumeId,
    String applicationName)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

internalVolumeId

ID of the internal volume.

applicationName

Name of the application to associate with the internal volume.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

setApplicationOfQtree

Associates a qtree with an application

```
setApplicationOfQtree(
    APISessionContext sessionContext,
    String qtreeId,
    String applicationName)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

qtreeId

ID of the qtree.

applicationName

Name of the application to associate with the qtree.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

setApplicationOfShare

Associates a share with an application.

```
public boolean setApplicationOfShare(
    APISessionContext sessionContext,
    String shareId,
    String applicationName)
```


Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

shareId

ID of the share.

applicationName

Name of the application to associate with the share.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

setApplicationOfVolume

Associates a volume with an application.

```
public boolean setApplicationOfVolume(
    APISessionContext sessionContext,
    String volumeId,
    String applicationName)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

volumeId

ID of a volume.

applicationName

Name of the application to be associated with the volume.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

setApplicationWithNoBusinessEntity

Adds a new application or updates the attributes of an existing application. The resulting application will not be associated with any business entity. For new applications, creates an application without the business entity or for existing applications, it updates the application and disassociates from any business entity.

A business entity holds tenant, business unit, line of business (LOB), and project information.

```
public boolean setApplicationWithNoBusinessEntity(
    APISessionContext sessionContext,
    String name,
    String priority,
    boolean ignoreSharing)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

name

Unique name of an application.

priority

Priority of the application. Can be one of the following values : Critical, High, Medium, Low.

ignoreSharing

Hosts associated with this application will be ignored when calculating volume sharing violation for other hosts.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

setBusinessEntity

Adds a business entity if it does not already exist. A business entity holds tenant, line of business (LOB), business unit, and project information.

```
public String setBusinessEntity(
    APISessionContext sessionContext,
    String tenant,
    String lineOfBusiness,
    String businessUnit,
    String project)
```

Parameters

This method has the following parameters.

sessionContext

Mandatory. The session identifies the snapshot of the data to be used.

tenant

Primarily used by service providers to associate resources with a customer, for example, NetApp.

lineOfBusiness

A line of business within a company, for example "Hardware" or "Software."

businessUnit

A traditional business unit such as "Sales" or "Marketing."

project

A project in your business.

Returns

The ID of the new or existing business entity.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

setBusinessEntityOfPort

Sets or unsets the business entity of a port. To unset the existing value, pass empty string ("") for businessEntityId.

```
public boolean setBusinessEntityOfPort(
    APISessionContext sessionContext,
    String portId,
    String businessEntityId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

portId

ID of the port

businessEntityId

ID of the business entity to set. To unset the existing value, use an empty string.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

setBusinessEntityOfStorage

Sets or unsets the business entity of a storage. To unset the existing value, pass an empty string ("") for businessEntityId.

```
public boolean setBusinessEntityOfStorage(
    APISessionContext sessionContext,
    String storageId,
    String businessEntityId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

storageId

ID of the storage system.

businessEntityId

ID of the business entity to set. To unset the existing value, use an empty string.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

setBusinessEntityOfSwitch

Sets or unsets the business entity of a switch. To unset the existing value, pass empty string ("") for businessEntityId.

```
public boolean setBusinessEntityOfSwitch(
    APISessionContext sessionContext,
    String switchId,
    String businessEntityId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

switchId

ID of the switch.

businessEntityId

ID of the business entity to set. To unset the existing value, use an empty string.

Returns

True upon successful completion of operation.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.2

setHostsPolicy

Creates policy for several hosts.

```
public boolean setHostsPolicy (
    APISessionContext sessionContext,
    String[] hostIds,
    String redundancy,
    int hopsNumber,
    int hostPortsRedundancy,
    int storagePortsRedundancy,
    String volumeSharingScope)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

hostIds

IDs of the hosts on which to create policies.

redundancy

Policy attribute redundancy, possible values: none, no-spf, redundant

hopsNumber

Maximum number of hops between host and storage.

hostPortsRedundancy

Minimum number of ports per host adapter.

storagePortsRedundancy

Minimum number of storage ports per storage controller.

volumeSharingScope

Possible values NO, ANY, CLUSTER

Returns

True upon successful completion of operation.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references[Host](#) on page 142[Policy](#) on page 157

updateReservationRequests

Updates existing request objects for OnCommand Insight Plan, the capacity management product.

```
public boolean updateReservationRequests (
    APISessionContext sessionContext,
    ReservationRequest[] requests)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestId

ID of the reservation request to update.

requests

The requests to update.

Returns

True upon successful completion of operation.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references[ReservationRequest](#) on page 169

updateReservationRequirements

Updates existing requirement objects of the request for OnCommand Insight Plan, the capacity management product.

```
public boolean updateReservationRequirements(
    APISessionContext sessionContext,
```

```
String requestId,
ReservationRequirement[] requirements)
```

Parameters

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestId

ID of the reservation request to update.

requirements

The requirement to update.

Returns

True upon successful completion of operation.

Throws

This method throws the following exceptions.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

java.rmi.RemoteException

RMI or WSAPI failure occurred.

New Since Version

Before 6.0

Related references

[ReservationRequest](#) on page 169

[ReservationRequirement](#) on page 174

getStoragePools

This method retrieves all the storage pools known to OnCommand Insight.

```
public StoragePoolResponse getStoragePools(
    APISessionContext sessionContext,
    RequestIterator requestIterator)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent hosts for a query that exceeded the session limit. If null, begins a new query. The session limit is set when a session is first opened.

Returns

Response object for storage pools. The object includes an array of storage pools up to the session limit together with an `RequestIterator`, which is used to fetch the rest of the nodes by subsequent invocations of this method.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.4

getStoragePoolsByStorageArray

This method retrieves all storage pools for a given storage array.

```
public StoragePoolResponse getStoragePoolsByStorageArray(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String storageArrayId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent storage arrays for a query that exceeded the session limit. If null, begins a new query. The session limit is set when a session is first opened.

storageArrayId

ID of the storage array.

Returns

Response object for storage pools. The object includes an array of storage pools for a given storage array up to the session limit together with an `RequestIterator`, which is used to fetch the rest of the nodes by subsequent invocations of this method.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.4

getStoragePool

This method retrieves a storage pool.

```
public StoragePool getStoragePool(  
    APISessionContext sessionContext,  
    String id)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

String id

The ID of the storage pool that needs to be fetched.

Returns

Response object for a storage pool. A single instance of a storage pool, if it exists. Otherwise, returns NULL.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.4

getVolumesByStoragePool

This method retrieves all the volumes for a given storage pool.

```
public VolumeResponse getVolumesByStoragePool(  
    APISessionContext sessionContext,  
    RequestIterator requestIterator,  
    String storagePoolId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent volumes for a query that exceeded the session limit. If null, begins a new query. The session limit is set when a session is first opened.

storagePoolId

Uniquely identifies the storage pool whose volumes are requested.

Returns

Response object for volumes. The object includes an array of volumes up to the session limit together with an RequestIterator, which is used to fetch the rest of the nodes by subsequent invocations of this method.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.4

getInternalVolumesByStoragePool

This method retrieves all internal volumes for a given storage pool.

```
public InternalVolumesResponse getInternalVolumesByStoragePool(
    APISessionContext sessionContext,
    RequestIterator requestIterator,
    String storagePoolId)
```

Parameters

This method has the following parameters.

sessionContext

The session identifies the snapshot of the data to be used. The response shows the nodes that were known at the time the session was created.

requestIterator

Optional. If specified, used to fetch subsequent hosts for a query that exceeded the session limit. If null, begins a new query. The session limit is set when a session is first opened.

storagePoolId

The ID of the storage pool whose internal volumes need to be fetched.

Returns

Response object for internal volumes. The object includes an array of internal volumes up to the session limit together with an RequestIterator, which is used to fetch the rest of the nodes by subsequent invocations of this method.

Throws

This method throws the following exception.

APIException

Can be caused by the server experiencing an internal error, invalid parameters being passed into the method, or an iterator (if used) requesting nodes when there are no more nodes to retrieve.

New Since Version

6.4

Connect API classes

You can use Connect API classes to integrate OnCommand Insight into your applications. You might want to review the list of classes that are provided in alphabetical order.

APIBadRequestException

Thrown to indicate that a client application provided wrong or inconsistent information to an API method call. All the methods that can throw this type of exception are defined in the class `SANscreenAPIEndPoint`.

Superclass

`APIException`

Constructors

To be used by WSAPI only (to deserialize an exception).

```
public APIBadRequestException()
```

Methods

The following methods are available from this class:

- `clone`
- `equals`
- `fillInStackTrace`
- `finalize`
- `getCause`
- `getClass`
- `getLocalizedMessage`
- `getMessage`
- `getStackTrace`
- `setStackTrace`
- `hashCode`
- `initCause`
- `notify`
- `notifyAll`
- `printStackTrace`
- `toString`

Action

Gets action for given action ID.

Superclass

BaseObject

Constructors

```
public Action()
```

```
public Action (String id, String taskId, String type, int actionOrder,
String state, String owner, ActionParam[] params, String[] errors)
```

Methods

The following methods are available from this class:

getActionOrder

```
public int getActionOrder()
```

setActionOrder

Should be used by the WSAPI.

```
public void setActionOrder(int actionOrder)
```

getErrors

```
public String[] getErrors()
```

setErrors

Should be used by the WSAPI.

```
public void setErrors(String[] errors)
```

getId

```
public String getId()
```

setId

```
public void TaskId(String id)
```

getOwner

```
public String getOwner()
```

setOwner

Should be used by the WSAPI.

```
public void setOwner(String owner)
```

getParameters

```
public ActionParam[] getParameters()
```

setParameters

Should be used by the WSAPI.

```
public void setParameters(ActionParam[] parameters)
```

getState

```
public String getState()
```

setState

Should be used by the WSAPI.

```
public void setState(String state)
```

getTaskId

```
public String getTaskId()
```

setTaskId

Should be used by the WSAPI.

```
public void setTaskId(String taskId)
```

getType

```
public String getType()
```

setType

Should be used by the WSAPI.

```
public void setType(String type)
```

Also see

Inherited methods from superclass BaseObject.

ActionResponse

Gets actions for given task. A response to a request for Action objects. This class contains an array of Action objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

BaseResponse

Constructors

Initiates an `ActionResponse` object.

```
public ActionResponse()
```

Methods

The following methods are available from this class:

getActions

```
public Action[] getActions()
```

setActions

Should be used by the WSAPI.

```
public void setActions(Action[] actions)Parameters
```

```
public String getWwn()
```

Also see

Inherited methods from superclass `BaseResponse`.

Related references

[Action](#) on page 118

AnnotationEnumValue

Represents one of the enumeration annotation values.

Superclass

`BaseObject`

Constructors

```
public AnnotationEnumValue()
```

Methods

The following methods are available from this class:

getDescription

Description of the enumeration value. (May be empty.)

```
public String getDescription()
```

setDescription

```
public void setDescription(String description)
```

getDisplayName

The display name of the enumeration value. (May be empty. If so, the name should be used.)

```
public String getDisplayName()
```

setDisplayName

```
public void setDisplayName(String displayName)
```

getName

The name of the enumeration value.

```
public String getName()
```

setName

```
public void setName(String name)
```

isUserDefined

Indication of whether the enumeration is user-defined or system-defined. System-defined values cannot be removed.

```
public boolean isUserDefined()
```

setUserDefined

```
public void setUserDefined(boolean userDefined)
```

Also see

Inherited methods from superclass `BaseObject`.

AnnotationType

Gets annotation type by name (for example, Tier or Data Center).

Superclass

`BaseObject`

Constructors

```
public AnnotationType()
```

Methods

The following methods are available from this class:

getDataType

```
public String getDataType()
```

setDataType

```
public void setDataType(String dataType)
```

getDescription

```
public String getDescription()
```

setDescription

```
public void setDescription(String description)
```

getDisplayName

```
public String getDisplayName()
```

setDisplayName

```
public void setDisplayName(String displayName)
```

getEnumerationValues

```
public AnnotationEnumValue[] getEnumerationValues()
```

setEnumerationValues

```
public void setEnumerationValues(AnnotationEnumValue[]  
enumerationValues)
```

getName

```
public String getName()
```

Gets the name of this device (usually, this is the DNS alias of the IP address).

setName

```
public void setName(String name)
```

getOwner

```
public String getOwner()
```

setOwner

```
public void setOwner(String owner)
```

getSupportedObjectTypes

```
public String[] getSupportedObjectTypes()
```

setSupportedObjectTypes

```
public void setSupportedObjectTypes(String[]
supportedObjectTypes)
```

isAllowMultipleValues

```
public boolean isAllowMultipleValues()
```

setAllowMultipleValues

```
public void setAllowMultipleValues(boolean allowMultipleValues)
```

isUserDefined

```
public boolean isUserDefined()
```

setUserDefined

```
public void setUserDefined(boolean userDefined)
```

Also see

Inherited methods from superclass `BaseObject`.

AnnotationValue

Gets annotation values for object references and annotation types.

Superclass

`BaseObject`

Constructors

```
public AnnotationValue()
```

Methods

The following methods are available from this class:

getEnumId

```
public String getEnumId()
```

setEnumId

```
public void setEnumId(String enumId)
```

getObjectReference

```
public ObjectReference getObjectReference()
```

setObjectReference

```
public void setObjectReference(ObjectReference objectReference)
```

getSetTime

```
public long getSetTime()
```

setSetTime

```
public void setSetTime(long setTime)
```

getTypeId

```
public String getTypeId()
```

setTypeId

```
public void setTypeId(String typeId)
```

getValue

```
public String getValue()
```

setValue

```
public void setValue(String value)
```

Also see

Inherited methods from superclass BaseObject.

APIException

Thrown to indicate a failure in the API server while processing a request from an API client application. All the methods that can throw this type of exception are defined in the class SANscreenAPIEndPoint.

Superclass

java.lang.Exception

Constructors

Used by WSAPI only (for deserializing the exception).

```
public APIException()
```

Methods

The following methods are available in this class:

- clone
- equals

- fillInStackTrace
- finalize
- getCause
- getClass
- getLocalizedMessage
- getMessage
- getStackTrace
- setStackTrace
- hashCode
- initCause
- notify
- notifyAll
- printStackTrace
- toString

APISessionContext

A session context is required with every query to the API server in order to provide a consistent view of the data. It is also used to enforce the API version being used by the client, as well as the maximum number of values returned in a single query.

Each session context equates to a snapshot of the data that was available when the session was created. Therefore, queries to a session context do not reflect any data changes that may have occurred after a session was opened.

A session context should only be instantiated using the `SANscreenAPIEndPoint.openSession(String, int)` method.

Typically, the life cycle of a session context in a client program should be:

Open a New Session

The server returns an instance of a session context that should be used for further queries.

Issue Queries

The client passes the session context to the server with every query, thus maintaining a consistent view data.

Close the Session

After the client is done querying the server, it should close the session and discard the session context.

Superclass

None.

Constructors

Constructs an uninitialized session context. This constructor should be used only by the WSAPI.

```
public APISessionContext()
```

Methods

The following methods are available for this class.

getClientVersion

```
public String getClientVersion()
```

Returns

The version of the API client that is using the context.

getId

```
public String getId()
```

Returns

The unique identifier of the context.

getLimit

```
public int getLimit()
```

Returns

The maximum number of values in a response to a query.

getTime

```
public long getTime()
```

Returns

The timestamp indicating when a session was opened.

Application

Represents an application to which a host in the SAN can belong.

Superclass

BaseObject

Constructors

Initializes an Application object.

```
public Application()
```

```
public Application()
```

Methods

The following methods are available from this class:

getBusinessEntityId

```
public String getBusinessEntityId()
```

setBusinessEntityId

```
public void setBusinessEntityId(String businessEntityId)
```

getHostIds

```
public String[] getHostIds()
```

setHostIds

```
public void setHostIds(String[] hostIds)
```

setIgnoreSharing

```
public void setIgnoreSharing(boolean ignoreSharing)
```

getInternalVolumeIds

```
public String[] getInternalVolumeIds()
```

setInternalVolumeIds

```
public void setInternalVolumeIds(String[] internalVolumeIds)
```

getName

```
public String getName()
```

setName

```
public void setName(String name)
```

getPortIds

```
public String[] getPortIds()
```

setPortIds

```
public void setPortIds(String[] portIds)
```

getPriority

```
public String getPriority()
```

setPriority

```
public void setPriority(String priority)
```

getQtreeIds

```
public String[] getQtreeIds()
```

setQtreeIds

```
public void setQtreeIds(String[] qtreeIds)
```

getShareIds

```
public String[] getShareIds()
```

setShareIds

```
public void setShareIds(String[] shareIds)
```

getStorageIds

```
public String[] getStorageIds()
```

setStorageIds

```
public void setStorageIds(String[] storageIds)
```

getSwitchIds

```
public String[] getSwitchIds()
```

setSwitchIds

```
public void setSwitchIds(String[] switchIds)
```

getVmIds

```
public String[] getVmIds()
```

setVmIds

```
public void setVmIds(String[] vmIds)
```

getVolumeIds

```
public String[] getVolumeIds()
```

setVolumeIds

```
public void setVolumeIds(String[] volumeIds)
```


isIgnoreSharing

```
public boolean isIgnoreSharing()
```

Also see

Inherited methods from superclass `BaseObject`.

ApplicationResponse

A response to a request for `Application` objects. This class contains an array of `Application` objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed number of returned objects.

Superclass

`BaseResponse`

Constructors

Initializes an `Application` response object.

```
public ApplicationResponse()
```

Methods

The following methods are available from this class:

getApplications

```
public Application[] getApplications()
```

setApplications

Should be used by the WSAPI.

```
public void setApplications(Application[] applications)
```

Also see

Inherited methods from superclass `BaseResponse`.

Related references

[Application](#) on page 126

[BaseResponse](#) on page 131

Attribute

The attribute of a reservation request target.

Superclass

None

Constructors

```
public Attribute()
```

Methods

The following methods are available from this class:

getName

Gets the name of the attribute.

```
public String getName()
```

setName

Used by the WSAPI.

```
public void setName (String name)
```

getValue

Gets the value of the attribute.

```
public String getValue()
```

setValue

Used by the WSAPI.

```
public void setValue(String value)
```

BaseObject

The base class for all objects returned by the OnCommand Insight API.

Superclass

None

Constructors

Initializes a history object.

```
public BaseObject()
```

Methods

The following methods are available from this class.

getId

Gets the unique identifier of this object.

```
public String getId()
```

Returns

The unique identifier of this object.

setId

Should only be used by WSAPI.

```
public void setId(String id)
```

Parameter**id**

The unique identifier for this object.

BaseResponse

Base class for responding to request in OnCommand Insight API. Encapsulates both a subset of objects matching the query and a request iterator that can be used to fetch the remaining objects.

Superclass

None

Constructors

```
public BaseResponse()
```

Methods

The following methods are available from this class:

getRequestIterator

Gets the iterator that is used by the client to determine if there are more objects to be fetched in addition to the objects in this response. Used by the server to determine which objects to return in subsequent invocations.

```
public RequestIterator getRequestIterator()
```

Returns

The iterator for this response.

setRequestIterator

Used by the WSAPI.

```
public void setRequestIterator(RequestIterator iterator)
```

getSize

Gets the number of objects contained in this response. This is not the total number of objects matching a query.

```
public int getSize()
```

Returns

The number of objects returned in this response.

setSize

The number of objects in this response. Used by the WSAPI.

```
public void setSize(int size)
```

BusinessEntity

Represents a business entity in the environment.

Superclass

BaseObject

Constructors

public BusinessEntity

```
public BusinessEntity(String id, String tenant, String  
lineOfBusiness, String businessUnit,  
String project, String displayName, String  
defaultApplicationName)
```

Methods

The following methods are available from this class:

getBusinessUnit

```
public String getBusinessUnit()
```

setBusinessUnit

```
public void setBusinessUnit(String businessUnit)
```

getDefaultApplicationName

```
public String getDefaultApplicationName()
```

setDefaultApplicationName

```
public void setDefaultApplicationName(String  
defaultApplicationName)
```

getDisplayName

```
public String getDisplayName()
```

setDisplayName

```
public void setDisplayName(String displayName)
```

getLineOfBusiness

```
public String getLineOfBusiness()
```

setLineOfBusiness

```
public void setLineOfBusiness(String lineOfBusiness)
```

getProject

```
public String getProject()
```

setProject

```
public void setProject(String project)
```

getTenant

```
public String getTenant()
```

setTenant

```
public void setTenant(String tenant)
```

Also see

Inherited methods from superclass `BaseObject`

Related references

[BaseObject](#) on page 130

DeviceBase

The base class of all SAN devices returned by the API.

Superclass

`HistoryObject`

Constructors

```
public DeviceBase()
```

Methods

The following methods are available from this class:

getIp

The TCP/IP address of this device, which cannot be NULL.

```
public String getIp()
```

Returns

The TCP/IP address of this device.

setIp

Used by the WSAPI.

```
public void setIp(String ip)
```

ip

The TCP/IP address of this device, which cannot be NULL.

getName

Gets the name of this device (usually, this is the DNS alias of the IP address).

```
public String getName()
```

Returns

The name of this device.

setName

Should be used by the WSAPI.

```
public void setName(String name)
```

name

The name of this device.

isDead

Indicator if the device is currently visible to OnCommand Insight. A device is not visible to OnCommand Insight if no active data-source is currently reporting on it.

```
public boolean isDead()
```

Returns

True if the device is not currently visible to OnCommand Insight; otherwise set to false.

setDead

Used by the WSAPI.

```
public void setDead(boolean dead)
```

dead

True if the device is not currently visible to OnCommand Insight; otherwise false.

Also see

Inherited methods from superclass HistoryObject.

DeviceGroup

Device group corresponding to Symmetrix device groups.

Superclass

HistoryObject

Constructors

Initializes a device group object.

```
public DeviceGroup()
```

Methods

The following methods are available from this class.

getName

```
public String getName()
```

setName

Should be used by the WSAPI.

```
public void setName(String name)
```

getStorageManagementId

```
public String getStorageManagementId()
```

setStorageManagementId

Should be used by the WSAPI.

```
public void setStorageManagementId(String storageManagementId)
```

getStorageManagementName

```
public String getStorageManagementName()
```

setStorageManagementName

Should be used by the WSAPI.

```
public void setStorageManagementName(String  
storageManagementName)
```

getType

```
public String getType()
```

setType

Should be used by the WSAPI.

```
public void setType(String type)
```

Also see

Inherited methods from superclass `HistoryObject`.

DeviceGroupResponse

A response to a request for Device group objects. This class contains an array of Device Group objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed number of returned objects.

Superclass

BaseResponse

Constructors

Initializes a device group response object.

```
public DeviceGroupResponse()
```

Methods

The following methods are available from this class:

getDeviceGroups

```
public DeviceGroup[] getDeviceGroups()
```

setDeviceGroups

Should be used by the WSAPI.

```
public void setDeviceGroups(DeviceGroup[] deviceGroups)
```

Also see

Inherited methods from superclass BaseResponse.

Related references

[DeviceGroup](#) on page 134

[BaseResponse](#) on page 131

DRPath

Disaster Recovery Path model object.

Superclass

BaseObject

Constructors

Initiates a DRPath object.

```
public DRPath()
```

Methods

The following methods are available from this class.

getDRType

Sets the type of disaster recovery. Number between 1 and 9.

```
public int getDRType()
```

setDRType

```
public void setDRType (int DRType)
```

getHostId

```
public String getHostId()
```

setHostId

Should be used by the WSAPI.

```
public void setHostId(String hostId)
```

getRemoteHostId

```
public String getRemoteHostId()
```

setRemoteHostId

Should be used by the WSAPI.

```
public void setRemoteHostId(String remoteHostId)
```

getRemoteReplicaVolumeId

```
public String getRemoteReplicaVolumeId()
```

setRemoteReplicaVolumeId

Should be used by the WSAPI.

```
public void setRemoteReplicaVolumeId(String  
remoteReplicaVolumeId)
```

getRemoteStorageId

```
public String getRemoteStorageId()
```

setRemoteStorageId

Should be used by the WSAPI.

```
public void setRemoteStorageId(String remoteStorageId)
```

getReplicaVolumeId

```
public String getReplicaVolumeId()
```

setReplicaVolumeId

Should be used by the WSAPI.

```
public void setReplicaVolumeId(String replicaVolumeId)
```

getSecondaryReplicaVolumeId

```
public String getSecondaryReplicaVolumeId()
```

setSecondaryReplicaVolumeId

Should be used by the WSAPI.

```
public void setSecondaryReplicaVolumeId(String  
secondaryReplicaVolumeId)
```

getStorageId

```
public String getStorageId()
```

setStorageId

Should be used by the WSAPI.

```
public void setStorageId(String storageId)
```

getVolumeId

```
public String getVolumeId()
```

setVolumeId

Should be used by the WSAPI.

```
public void setVolumeId(String volumeId)
```

Also see

Inherited methods from superclass `BaseObject`.

DRPathResponse

A response to a request for Disaster Recovery Path objects. This class contains an array of `DRPath` objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

`BaseResponse`

Constructors

Initiates a `DRPath` response object.

```
public DRPathResponse()
```

Methods

The following methods are available from this class:

getDrPaths

```
public DRPath[] getDrPaths()
```

setDrPaths

Should be used by the WSAPI.

```
public void setDrPaths(DRPath[] drPaths)
```

Also see

Inherited methods from superclass `BaseResponse`.

Related references

[DRPath](#) on page 136

[BaseResponse](#) on page 131

Generic

Represents a generic device. Generic devices are those known to OnCommand Insight but not yet identified.

Superclass

`BaseObject`

Constructors

Initializes a generic object.

```
public Generic()
```

Methods

The following methods are inherited from this class:

getWwn

```
public String getWwn()
```

Returns

The World Wide Name.

setWwn

The World Wide Name. Should be used by the WSAPI.

```
public void setWwn(String wwn)
```

isDead

```
public boolean isDead()
```

Returns

True if this device is not currently visible to OnCommand Insight; otherwise false.

setDead

Indicates if a device is currently visible to OnCommand Insight. True = visible. False = not visible. Should be used by the WSAPI.

```
public void setDead(boolean dead)
```

Also see

Inherited methods from superclass BaseObject.

Related references

[GenericResponse](#) on page 140

GenericResponse

A response to a request for Generic objects. This class contains an array of Generic objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

BaseResponse

Constructors

The following constructor initializes a generic response object.

```
public GenericResponse()
```

Methods

The following methods are available from this class:

getGenerics

Generic devices are ports on unknown devices. An unknown device could be a switch, tape, disc storage controller, or an HBA with ports.

```
public Generic[] getGenerics()
```

Returns

Generic devices known to OnCommand Insight.

setGenerics

Should be used by the WSAPI.

```
public void setGenerics(Generic[] generics)
```

generics

Generic devices known to OnCommand Insight. Generic devices are ports on unknown devices. An unknown device could be a switch, tape, disc storage controller, or an HBA with ports.

Also see

Inherited methods from superclass `BaseResponse`.

Related references

[Generic](#) on page 139

[BaseResponse](#) on page 131

HistoryObject

The base class for any history-aware object.

Superclass

`BaseObject`

Constructors

Initializes a history object.

```
public HistoryObject()
```

Methods

The following methods are available from this class.

`getStartTime`

```
public long getStartTime()
```

`setStartTime`

Should be used by the WSAPI.

```
public void setStartTime(long startTime)
```

`getEndTime`

```
public long getEndTime()
```

`setEndTime`

Should be used by the WSAPI.

```
public void setEndTime(long endTime)
```

Also see

Inherited methods from superclass `BaseObject`.

Host

Represents a host or server in the SAN.

Superclass

DeviceBase

```
BaseObject
|__HistoryObject
|__DeviceBase
```

Constructors

Used by WSAPI only.

```
public Host()
```

Methods

All methods are available from the superclass.

Also see

Inherited methods from superclass DeviceBase.

Related references

[DeviceBase](#) on page 133

[getHosts](#) on page 51

HostResponse

A response to a request for Host objects. This class contains an array of Host objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

BaseResponse

Constructors

Used by WSAPI only.

```
public HostResponse()
```

Methods

The following methods are available from this class:

getHosts

```
public Host[] getHosts()
```

Returns

The hosts and servers contained in the host response.

setHosts

Should be used by the WSAPI.

```
public void setHosts(Host[] hosts)
```

hosts

The hosts in the host response.

Also see

Inherited methods from superclass `BaseResponse`.

Related references

[Host](#) on page 142

[BaseResponse](#) on page 131

InternalVolume

Represents an internal volume.

Superclass

`HistoryObject`

```
BaseObject
|__HistoryObject
```

Constructors

Initiates the internal volume object.

```
public InternalVolume()
```

Methods

The following methods are available from this class:

getCloneSourceId

Gets the ID of the clone source.

```
public String getCloneSourceId()
```

setCloneSourceId

Should be used by the WSAPI.

```
public void setCloneSourceId(String cloneSourceId)
```

getDataAllocatedCapacityMB

Capacity allocated for data.

```
public long DataAllocatedCapacityMB()
```

setDataAllocatedCapacityMB

```
public void setDataAllocatedCapacityMB(long  
DataAllocatedCapacityMB)
```

getDataUsedCapacityMB

Capacity used for data.

```
public long getDataUsedCapacityMB()
```

setDataUsedCapacityMB

```
public void setDataUsedCapacityMB(long DataUsedCapacityMB)
```

getDedupeSavings

Gets the savings data based on the use of deduplication.

```
public float getDedupeSavings()
```

setDedupeSavings

Should be used by the WSAPI.

```
public void setDedupeSavings(float dedupeSavings)
```

getHead

Gets the name of the head.

```
public String getHead()
```

setHead

Should be used by the WSAPI.

```
public void setHead(String head)
```

getOtherAllocatedCapacityMB

Allocated other capacity for internal volume (overhead attributed to different technologies).

```
public long getOtherAllocatedCapacityMB()
```

setOtherAllocatedCapacityMB

```
public void setOtherAllocatedCapacityMB(long  
OtherAllocatedCapacityMB)
```

getOtherUsedCapacityMB

Used other capacity for internal volume (overhead attributed to different technologies).

```
public long getOtherUsedCapacityMB()
```

setOtherUsedCapacityMB

```
public void setOtherUsedCapacityMB(long getOtherUsedCapacityMB)
```

getProtectionType

Gets the type of protection on the internal volume.

```
public String getProtectionType()
```

setProtectionType

Should be used by the WSAPI.

```
public void setProtectionType(String protectionType)
```

getRawToUsableRatio

Gets the ratio of raw-to-usable capacity on the device.

```
public float getRawToUsableRatio()
```

setRawToUsableRatio

Should be used by the WSAPI.

```
public void setRawToUsableRatio(float rawToUsableRatio)
```

getSnapshotAllocatedCapacityMB

Capacity allocated for Snapshot information.

```
public long getSnapshotAllocatedCapacityMB()
```

setSnapshotAllocatedCapacityMB

```
public void setSnapshotAllocatedCapacityMB(long  
SnapshotAllocatedCapacityMB)
```

getSnapshotUsedCapacityMB

Capacity used for Snapshot information.

```
public long setSnapshotUsedCapacityMB()
```

setSnapshotUsedCapacityMB

```
public void setSnapshotUsedCapacityMB(long  
SnapshotUsedCapacityMB)
```

getSpaceGuarantee

Gets a string indicating the space guarantee.

```
public String getSpaceGuarantee()
```

setSpaceGuarantee

Should be used by the WSAPI.

```
public void setSpaceGuarantee(String spaceGuarantee)
```

getStatus

Gets the status of the internal volume.

```
public String getStatus()
```

setStatus

Should be used by the WSAPI.

```
public void setStatus(String status)
```

getStorageId

Gets the ID of the storage object.

```
public String getStorageId()
```

setStorageId

Should be used by the WSAPI.

```
public void setStorageId(String storageId)
```

getStoragePool

Gets the ID of the storage pool.

```
public String getStoragePoolId()
```

setStoragePoolId

Should be used by the WSAPI.

```
public void setStoragePoolId(String storagePoolId)
```

getTotalAllocatedCapacityMB

Total allocated capacity for internal volume.

```
public long getTotalAllocatedCapacityMB()
```

setTotalAllocatedCapacityMB

```
public void setTotalAllocatedCapacityMB(long  
TotalAllocatedCapacityMB)
```

getTotalCloneSavedCapacityMB

Clone saved capacity of the internal volume.

```
public long getTotalCloneSavedCapacityMB()
```

getTotalCloneSavedCapacityMB

```
public void getTotalCloneSavedCapacityMB(long
getTotalCloneSavedCapacityMB)
```

getTotalUsedCapacityMB

Total used capacity for internal volume.

```
public long getTotalUsedCapacityMB()
```

setTotalUsedCapacityMB

```
public void setTotalUsedCapacityMB(long TotalUsedCapacityMB)
```

getVirtualStorage

Gets the name of the virtual storage.

```
public String getVirtualStorage()
```

setVirtualStorage

Should be used by the WSAPI.

```
public void setVirtualStorage(String virtualStorage)
```

isDedupeEnabled

True if deduplication is enabled. Otherwise, **False**.

```
public boolean isDedupeEnabled() {
```

setDedupeEnabled

Should be used by the WSAPI.

```
public void setDedupeEnabled(boolean dedupeEnabled)
```

isThinProvisioned

```
public boolean isThinProvisioned()
```

True if thin provisioned. Otherwise, **False**.

setThinProvisioned

Should be used by the WSAPI.

```
public void setThinProvisioned(boolean thinProvisioned)
```

isThinProvisioningSupported

Indicates whether thin provisioning is supported.

```
public boolean isThinProvisioningSupported()
```

setThinProvisioningSupported

Should be used by the WSAPI.

```
public void setThinProvisioningSupported(boolean
thinProvisioningSupported)
```

Also see

Inherited methods from superclass HistoryObject.

New Since Version

6.0

InternalVolumeResponse

The response to a request for InternalVolume objects. This class contains an array of InternalVolume objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed number of returned objects.

Superclass

BaseResponse

Constructors

Used by WSAPI only.

```
public InternalVolumeResponse()
```

Methods

The following methods are available from this class:

getInternalVolumes

```
public InternalVolume[] getInternalVolumes()
```

Returns

internal volumes - The internal volumes contained in the host response.

setInternalVolumes

Should be used by the WSAPI.

```
public void setInternalVolumes(InternalVolume[] internalVolumes)
```

Parameter

internalVolumes

The internal volumes in the host response.

Also see

Inherited methods from superclass BaseResponse.

New Since Version

6.0

Related references

[Internal Volume](#) on page 143

LogicalPort

Represents a logical port in the SAN. Enables you to retrieve port information.

Superclass

Port

Constructors

```
public LogicalPort()
```

Initiates the logical port object

Methods

The following methods are available from this class:

getPhysicalPortId

```
public String getPhysicalPortId()
```

Returns

The physical port corresponding to the logical port.

setPhysicalPortId

```
public void setPhysicalPortId(String physicalPortId)
```

Used by the WSAPI.

Also see

Inherited methods from superclass Port.

LogicalPortResponse

A response to a request for LogicalPort objects. This class contains an array of LogicalPort objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

BaseResponse

It is returned by the following methods:

- SANscreenAPIEndPoint.getConnectionedPorts(APISessionContext, RequestIterator)
- SANscreenAPIEndPoint.getLogLogicalPortsByPhysicalPort(APISessionContext RequestIterator)
- SANscreenAPIEndPoint.getLogLogicalPortsByLogicalSwitch(APISessionContext, RequestIterator)
- SANscreenAPIEndPoint.getLogLogicalPortsByPhysicalSwitch(APISessionContext, RequestIterator)

Constructors

Initiates the logical port response object

```
public LogicalPortResponse()
```

Methods

The following methods are available from this class:

getLogicalPorts

```
public LogicalPort[] getLogicalPorts()
```

Returns

The logical ports known to OnCommand Insight.

setLogicalPorts

Should be used by the WSAPI.

```
public void setLogicalPorts(LogicalPort[] logicalPorts)
```

Also see

Inherited methods from superclass BaseResponse.

Related references

[LogicalPort](#) on page 149

[getConnectedPorts](#) on page 42

[getLogicalPortsByPhysicalPort](#) on page 56

[getLogicalPortsByLogicalSwitch](#) on page 55

[getLogicalPortsByPhysicalSwitch](#) on page 57

[BaseResponse](#) on page 131

Node

Represents a node in the SAN. You can retrieve known nodes using the method call SANscreenAPIEndPoint.getNodesOfDevice(APISessionContext,RequestIterator).

Superclass

BaseObject

Constructors

```
public Node()
```

Initiates the node object

Methods

The following methods are available from this class:

getDeviceID

```
public String getDeviceID()
```

Returns

The ID of the device where a node resides.

setDeviceID

```
public void setDeviceID(String deviceId)
```

Should be used by the WSAPI.

getWwn

```
public String getWwn()
```

Returns

The World Wide Name of the node.

setWwn

```
public void setWwn(String wwn)
```

Used by the WSAPI.

Parameter

wwn

The World Wide Name of the node.

isDead

```
public boolean isDead()
```

Returns

True if this device is not currently visible to OnCommand Insight; otherwise false.

setDead

Indicates whether a device is currently visible to OnCommand Insight. True = visible.
False = not visible.

```
public void setDead(boolean dead)
```

Used by the WSAPI.

Also see

Inherited methods from superclass BaseObject.

Related references

[*getNodeOfDevice*](#) on page 58

NodeResponse

A response to a request for Node objects. This class contains an array of Node objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

BaseResponse

Constructors

Initiates a node response object.

```
public NodeResponse()
```

Methods

The following methods are available from this class:

getNodes

Gets the nodes of a device, such as a host's HBA or the controller of a storage device.

```
public Node[] getNodes()
```

Returns

The nodes of a device, such as a host's HBA or the controller of a storage device.

setNodes

```
public void setNodes(Node[] nodes)
```

Should be used by the WSAPI.

Also see

Inherited methods from superclass BaseResponse.

Related references

[Node](#) on page 150

[getNodesOfDevice](#) on page 58

[BaseResponse](#) on page 131

Path

Represents a path in the SAN. You can retrieve known paths using the `SANscreenAPIEndPoint.getPaths(APISessionContext,RequestIterator)` method call.

Superclass

HistoryObject

```
BaseObject
|__HistoryObject
```

Constructors

Initiates a path object

```
public Path()
```

Methods

The following methods are available from this class:

getArrayId

```
public String getArrayId()
```

Returns

The name of the storage device where the data for this path resides.

setArrayId

Should be used by the WSAPI. The name of the storage device where the data for this path resides.

```
public void setArrayId(String arrayId)
```

getHostId

```
public String getHostId()
```

Returns

The name of the host from which the path originates.

setHostId

Should be used by the WSAPI. The name of the host from which the path originates.

```
public void setHostId(String hostId)
```

getVolumeId

```
public String getVolumeId()
```

Returns

The name of the volume where the data for this path resides.

setVolumeId

Should be used by the WSAPI. The name of the volume where the data for this path resides.

```
public void setVolumeId(String volumeId)
```

Also see

Inherited methods from superclass `HistoryObject`.

Related references

[getPath](#) on page 60

[HistoryObject](#) on page 141

PathReservation

This class contains metadata for a path reservation.

Superclass

`BaseResponse`

Constructors

Initiates a path reservation object.

```
public PathReservation()
```

Methods

The following methods are available from this class:

getRequestId

```
public String getRequestId()
```

Returns

Returns requestId of type `String`.

getRequirementId

```
public String getRequirementId()
```

Returns

Returns requirementId of type `String`.

getReservationId

```
public String getReservationId()
```

Returns

Returns reservationId of type `String`.

getStorage

```
public DeviceBase getStorage()
```

Returns

Returns storage object associated with storage ports of type DeviceBase.

getStoragePorts

```
public Port () getStoragePorts()
```

Returns

Returns an array of reserved storage ports of type Port.

getVolumes

```
public Volume () getVolumes()
```

Returns

Returns an array of reserved volumes of type Volume.

Also see

Inherited methods from superclass BaseResponse.

New Since Version

6.4

Related references

[BaseResponse](#) on page 131

PathReservationResponse

This class contains an array of PathReservation objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed number of objects returned.

Superclass

BaseResponse

Constructors

Initiates a path reservation response object.

```
public PathReservationResponse()
```

Method

The following method is available from this class:

getPathReservations

```
public PathReservation () getPathReservations()
```

Returns

An array of PathReservation.

Also see

Inherited methods from superclass `BaseResponse`.

New Since Version

6.4

Related references

[BaseResponse](#) on page 131

PathResponse

A response to a request for Path objects. This class contains an array of Path objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed number of returned objects.

Superclass

`BaseResponse`

Constructors

Initiates a path response object.

```
public PathResponse()
```

Methods

The following methods are available from this class:

getPaths

```
public Path[] getPaths()
```

Returns

The logical paths detected by OnCommand Insight in the SAN being monitored.

setPaths

Should be used by the WSAPI.

```
public void setNodes(Path[] paths)
```

Also see

Inherited methods from superclass `BaseResponse`.

Related references

[Path](#) on page 153

[getPaths](#) on page 60

[BaseResponse](#) on page 131

Policy

Represents a policy.

Superclass

BaseObject

Constructors

Initiates a policy object.

```
public Policy()
```

Methods

The following methods are available from this class:

getHostId

```
public String getHostId()
```

setHostId

Should be used by the WSAPI.

```
public void setHostId(String hostId)
```

getArrayId

```
public String getArrayId()
```

setArrayId

Should be used by the WSAPI.

```
public void setArrayId(String arrayId)
```

getHopsNumber

```
public int getHopsNumber()
```

setHopsNumber

Should be used by the WSAPI.

```
public void setHopsNumber(int hopsNumber)
```

getHostPortsRedundancy

```
public int getHostPortsRedundancy()
```

setHostPortsRedundancy

Should be used by the WSAPI.

```
public void setHostPortsRedundancy(int hostPortsRedundancy)
```

getOwner

```
public String getOwner()
```

setOwner

Should be used by the WSAPI.

```
public void setOwner(String owner)
```

getRedundancy

```
public String getRedundancy()
```

setRedundancy

Should be used by the WSAPI.

```
public void setRedundancy(String redundancy)
```

getStoragePortRedundancy

```
public String getStoragePortRedundancy()
```

setStoragePortRedundancy

```
public void setStoragePortRedundancy(int StoragePortRedundancy)
```

getVolumeId

```
public String getVolumeId()
```

setVolumeId

Should be used by the WSAPI.

```
public void setVolumeId(String volumeId)
```

getVolumeSharingScope

```
public String getVolumeSharingScope()
```

setVolumeSharingScope

Should be used by the WSAPI.

```
public void setVolumeSharingScope(String volumeSharingScope)
```

Also see

Inherited methods from superclass BaseObject.

PolicyResponse

A response to a request for Policy objects. This class contains an array of Policy objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

BaseResponse

Constructors

Initiates a policy object.

```
public PolicyResponse()
```

Methods

The following methods are available from this class:

getPolicies

```
public Policy[] getPolicies()
```

setPolicies

Should be used by the WSAPI.

```
public void setPolicies(Policy[] policies)
```

Also see

Inherited methods from superclass BaseResponse.

Related references

[Policy](#) on page 157

[getPolicies](#) on page 63

[BaseResponse](#) on page 131

Port

Represents a port in the SAN. You can retrieve known ports.

Superclass

BaseObject

Direct Known Subclasses

LogicalPort

Constructors

Initiates a Port object.

```
public Port()
```

Methods

The following methods are available from this class:

getConnectedPortId

```
public String getConnectedPortId()
```

Returns

The ID of the port to which the other port is connected.

setConnectedPortId

The ID of the port to which the other port is connected. Should be used by the WSAPI.

```
public void setConnectedPortId(String connectedPortId)
```

getDeviceId

```
public String getDeviceId()
```

Returns

The ID of the device to which the node belongs.

setDeviceId

Should be used by the WSAPI.

```
public void setDeviceId(String deviceId)
```

getName

```
public String getName()
```

Returns

Name of the port.

setName

Name of the port. Should be used by the WSAPI.

```
public void setName(String name)
```

getNodeId

```
public String getNodeId()
```

Returns

ID of the device that corresponds to the node to which this port belongs. For example, HBA of the host or protocol controller of the storage device.

setNodeId

Sets the ID of the device that corresponds to the node to which this port belongs. For example, HBA of the host or protocol controller of the storage device. Should be used by the WSAPI.

```
public void setNodeId(String nodeId)
```

getReservationStatus

```
public String getReservationStatus()
```

Returns

The reservation status for the port (one of "FREE", "USED", or "RESERVED").

setReservationStatus

Should be used by the WSAPI.

```
public void setReservationStatus(String reservationStatus)
```

getState

```
public String getState()
```

Returns

Port state

setState

Should be used by the WSAPI.

```
public void setState(String state)
```

getStatus

```
public String getStatus()
```

Returns

Port Status - either connected or disconnected.

setStatus

Should be used by the WSAPI.

```
public void setStatus(String status)
```

getWwn

```
public String getWwn()
```

Returns

The World Wide Name of the port.

setWwn

Should be used by the WSAPI.

```
public void setWwn(String wwn)
```

isDead

```
public boolean isDead()
```

Returns

True if this device is not currently visible to OnCommand Insight.

setDead

True if this device is not currently visible to OnCommand Insight. Should be used by the WSAPI.

```
public void setDead(boolean dead)
```

Also see

Inherited methods from superclass `BaseObject`.

PortResponse

A response to a request for Port objects. A partial response to any of several method calls that retrieve ports known to OnCommand Insight in the SAN being monitored. This class contains an array of Port objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed number of returned objects.

Superclass

`BaseResponse`

Constructors

Initiates a Port Response object.

```
public PortResponse()
```

Methods

The following methods are available from this class:

getPorts

Returns the request iterator for this response.

```
public Port[] getPorts()
```

setPorts

Should be used by the WSAPI.

```
public void setPorts(Port[] ports)
```

Also see

Inherited methods from superclass `BaseResponse`.

Related references

[Port](#) on page 159

[BaseResponse](#) on page 131

PortsByFabric

This class contains a Fabric object and its associated ports.

Superclass

BaseObject

Constructors

Initiates a fabric response object.

```
public PortsByFabric()
```

Methods

The following methods are available from this class:

getFabric

```
public Fabric getFabric()
```

Returns

Returns Fabric object.

getPortIds

```
public String() getPortIds()
```

Returns

Returns portIds of type String.

Also see

Inherited methods from superclass BaseObject.

New Since Version

6.4

Related references

[BaseResponse](#) on page 131

PortsByFabricResponse

A response to a request for PortsByFabric objects. A partial response to any of several method calls that retrieve ports known to OnCommand Insight in the SAN being monitored. This class contains an array of PortsByFabric objects, and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed number of returned objects.

Superclass

BaseResponse

Constructors

Initiates a PortsByFabricResponse object.

```
public PortsByFabricResponse()
```

Method

The following method is available from this class:

getPortsByFabric

Returns the request iterator for this response.

```
public PortsByFabric[] getPortsByFabric()
```

Also see

Inherited methods from superclass BaseResponse.

New Since Version

6.4

PortsByZone

This class contains Zone object and its associated port IDs.

Superclass

BaseObject

Constructors

Initiates a path response object.

```
public PortsByZone()
```

Methods

The following methods are available from this class:

getPortIds

```
public String () getPortIds()
```

Returns

Returns portIds of type String.

getZone

```
public Zone getZone()
```

Returns

Returns Zone object.

Also see

Inherited methods from superclass BaseObject.

New Since Version

6.4

Related references[BaseResponse](#) on page 131

PortsByZoneResponse

A response to a request for PortsByZone objects. A partial response to any of several method calls that retrieve ports known to OnCommand Insight in the SAN being monitored. This class contains an array of PortsByZone objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed number of returned objects.

Superclass

BaseResponse

Constructors

Initiates a PortsByZoneResponse object.

```
public PortsByZoneResponse()
```

Method

The following method is available from this class:

getPortsByZone

Returns the PortsByZone array.

```
public PortsByZone[] getPortsByZone()
```

Also see

Inherited methods from superclass BaseResponse.

New Since Version

6.4

Qtree

Represents a qtree in the environment.

Superclass

HistoryObject

```
BaseObject
|__HistoryObject
```

Constructors

```
public Qtree()
```

Initializes a qtree object.

```
public Qtree (String id, String identifier,
String storageId, String name, String internalVolumeId,
String type, String securityStyle,
String status, boolean oplocks,
long quotaHardCapacityLimitMB, long quotaSoftCapacityLimitMB,
long startTime, long endTime)
```

Methods

The following methods are available from this class:

getIdentifier

```
public String getIdentifier()
```

setIdentifier

```
public void setIdentifier(String identifier)
```

getInternalVolumeId

```
public String getInternalVolumeId()
```

setInternalVolumeId

```
public void setInternalVolumeId(String internalVolumeId)
```

getName

```
public String getName()
```

setName

```
public void setName(String name)
```

getQuotaHardCapacityLimitMB

```
public long getQuotaHardCapacityLimitMB()
```

setQuotaHardCapacityLimitMB

```
public void setQuotaHardCapacityLimitMB(long
quotaHardCapacityLimitMB)
```

getQuotaSoftCapacityLimitMB

```
public long getQuotaSoftCapacityLimitMB()
```

setQuotaSoftCapacityLimitMB

```
public void setQuotaSoftCapacityLimitMB(long
quotaSoftCapacityLimitMB)
```

getSecurityStyle

```
public String getSecurityStyle()
```

setSecurityStyle

```
public void setSecurityStyle(String securityStyle)
```

getStatus

```
public String getStatus()
```

setStatus

```
public void setStatus(String status)
```

getStorageId

```
public String getStorageId()
```

setStorageId

```
public void setStorageId(String storageId)
```

getType

```
public String getType()
```

setType

```
public void setType(String type)
```

isOplocks

```
public boolean isOplocks()
```

setOplocks

```
public void setOplocks(boolean oplocks)
```

Also see

Inherited methods from superclass `HistoryObject`.

Related references

[HistoryObject](#) on page 141

RequestIterator

A `RequestIterator` is used by OnCommand Insight API to track and manage client query responses. Clients use the iterator to determine if there are more objects to be fetched by invoking the query

again. The server uses the iterator to determine which objects should be returned in subsequent invocations of the query.

Constructors

To be used by WSAPI only (for deserializing the iterator).

```
public RequestIterator()
```

Methods

The following methods are available from this class.

getNextKey

Gets the identifier of objects which should be returned on the next invocation of the query. Used internally by the API server.

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

Returns

The identifier of the objects that should be returned on the next invocation of the query.

setNextKey

To be used by WSAPI only (for deserializing this iterator).

```
public void setNextKey(java.lang.String nextKey)
```

Parameter

nextKey

Identifies the objects that will be returned on the next invocation of the query.

isHasMore

Tells the client if there are more objects to be fetched by subsequent invocations of the query.

```
public boolean isHasMore()
```

Returns

True if the client should invoke the query again using this iterator to get the next bulk of objects matching the query.

setHasMore

To be used by WSAPI only (for deserializing this response).

```
public void setHasMore(boolean hasMore)
```

Parameter

hasMore

True if the client should invoke the query again using this iterator to get the next bulk of objects matching the query).

getMethodName

Gets the name of the API query for which this iterator is used as part of the response.

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

Returns

The name of the query for which this iterator was created.

setMethodName

To be used by WSAPI only (for deserializing this response).

```
public void setMethodName(java.lang.String methodName)
```

Parameter

methodName

The name of the query for which this iterator was created.

ReservationRequest

Used to generate reservation request for capacity.

Superclass

BaseObject

Constructors

```
public ReservationRequest()
```

Methods

The following methods are available from this class:

getBusinessEntityId

```
public long getBusinessEntityId()
```

setBusinessEntityId

Should be used by the WSAPI.

```
public void setBusinessEntityId(long BusinessEntityId)
```

getComments

```
public String getComments()
```

setComments

Should be used by the WSAPI.

```
public void setComments(String comments)
```

getCompletionDate

```
public long getCompletionDate()
```

setCompletionDate

Should be used by the WSAPI.

```
public void setCompletionDate(long completionDate)
```

getCreationDate

```
public long getCreationDate()
```

setCreationDate

Should be used by the WSAPI.

```
public void setCreationDate(long creationDate)
```

getDataCenterId

```
public long getDataCenterId()
```

setDataCenterId

Should be used by the WSAPI.

```
public void setDataCenterId(long dataCenterId)
```

getDescription

```
public String getDescription()
```

setDescription

Should be used by the WSAPI.

```
public void setDescription(String description)
```

getDisplayName

```
public String getDisplayName()
```

setDisplayName

Should be used by the WSAPI.

```
public void setDisplayName(String displayName)
```

getDueDate

```
public long getDueDate()
```

setDueDate

Should be used by the WSAPI.

```
public void setDueDate(long dueDate)
```

getErrorCount

```
public int getErrorCount()
```

setErrorCount

Should be used by the WSAPI.

```
public void setErrorCount(int errorCount)
```

getJustification

```
public String getJustification()
```

setJustification

Should be used by the WSAPI.

```
public void setJustification(String justification)
```

getOwner

```
public String getOwner()
```

setOwner

Should be used by the WSAPI.

```
public void setOwner(String owner)
```

getProjectId

```
public long getProjectId()
```

setProjectId

Should be used by the WSAPI.

```
public void setProjectId(long projectId)
```

getRequester

```
public String getRequester()
```

setRequester

Should be used by the WSAPI.

```
public void setRequester(String requester)
```

getRequesterEmail

```
public String getRequesterEmail()
```

setRequesterEmail

Should be used by the WSAPI.

```
public void setRequesterEmail(String requesterEmail)
```

getRequestTargets

```
public ReservationRequestTarget[] getRequestTargets()
```

setRequestTargets

Should be used by the WSAPI.

```
public void setRequestTargets(ReservationRequestTarget[]  
requestTargets)
```

getSanId

```
public long getSanId()
```

setSanId

Should be used by the WSAPI.

```
public void setSanId(long sanId)
```

getState

Possible values: OPEN, NEW, COMPLETED, REJECTED, CANCELED.

```
public String getState()
```

setState

Should be used by the WSAPI.

```
public void setState(String state)
```

getTargetType

Possible values: HOST, OTHER, NONE

```
public String getTargetType()
```

setTargetType

Should be used by the WSAPI.

```
public void setTargetType(String targetType)
```

getTicket

```
public String getTicket()
```

setTicket

Should be used by the WSAPI.

```
public void setTicket(String ticket)
```

getWarningCount

```
public int getWarningCount()
```

setWarningCount

Should be used by the WSAPI.

```
public void setWarningCount(int warningCount)
```

isSatisfied

```
public boolean isSatisfied()
```

setSatisfied

Should be used by the WSAPI.

```
public void setSatisfied(boolean satisfied)
```

Also see

Inherited methods from superclass `BaseObject`.

Related references

[ReservationRequestTarget](#) on page 173

[BaseObject](#) on page 130

ReservationRequestTarget

The target for your reservation request.

Superclass

None

Constructors

```
public ReservationRequestTarget()
```

Methods

The following methods are available from this class:

getAttributes

Gets the attributes for resource being requested (for example, `IP_ADDRESS` or `HOST_NAME`).

```
public Attribute[] getAttributes()
```

setAttributes

Used by the WSAPI.

```
public void setAttributes(Attribute[] attributes)
```

getType

Gets the type of resource being requested.

```
public String getType()
```

setType

Used by the WSAPI.

```
public void setType(String type)
```

ReservationRequirement

Represents requirement for Reservation Request.

Superclass

ReservationRequestTarget

Constructors

```
public ReservationRequirement()
```

Methods

The following methods are available from this class:

isSatisfied

```
public boolean issatisfied()
```

setSatisfied

Should be used by the WSAPI.

```
public void setSatisfied(boolean satisfied)
```

getNote

```
public String getNote()
```

setNote

Should be used by the WSAPI.

```
public void setNote(String note)
```

Also see

Inherited methods from superclass ReservationRequestTarget.

Related references

[ReservationRequest](#) on page 169

[BaseObject](#) on page 130

ReservationViolation

This class contains metadata for reservationViolation.

Superclass

BaseObject

Constructors

Initiates a ReservationViolation object.

```
public ReservationViolation()
```

Methods

The following methods are available from this class:

getDetails

```
public String getDetails()
```

Returns

Returns the details of the violation of type String.

getId

```
public String getId()
```

Returns

Returns the ID of the reservation violation of type String.

getRequestId

```
public String getRequestId()
```

Returns

Returns the requestId of the violation of type String.

getReservationId

```
public String getReservationId()
```

Returns

Returns reservationId of the violation of type String.

getSince

```
public long getSince()
```

Returns

Returns the date since the violation was triggered of type Long.

getTargetDevice

```
public String getTargetDevice()
```

Returns

Returns target device of type String.

getTargetDeviceType

```
public String getTargetDeviceType()
```

Returns

Returns the target device type, such as fabric, zone, host, or storage, of type String.

getViolationType

```
public String getViolationType()
```

Returns

Returns the type of violation, for example, "POTENTIALLY_INCORRECT_VOLUME_RESERVATION" of type String.

Also see

Inherited methods from superclass BaseObject.

New Since Version

6.4

Related references

[BaseResponse](#) on page 131

ReservationViolationsByRequest

This class contains requestId and its associated ReservationViolation objects.

Superclass

BaseObject

Constructors

Initiates reservation violations response object.

```
public ReservationViolationsByRequest()
```

Methods

The following methods are available from this class:

getRequestId

```
public String getRequestId()
```

Returns

Returns requestId of type String.

getViolations

```
public ReservationViolation () getViolations()
```

Returns

Returns an array of violation objects associated with the request ID.

Also see

Inherited methods from superclass BaseObject.

New Since Version

6.4

Related references

[BaseResponse](#) on page 131

ReservationViolationsByRequestResponse

This class contains an array of ReservationViolationsByRequest objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

BaseResponse

Constructors

Initiates a path response object.

```
public PathViolationsByRequestResponse()
```

Methods

The following methods are available from this class:

getReservationViolationsForRequest

```
public ReservationViolationsByRequest ()  
getReservationViolationsForRequest()
```

Returns

An array of ReservationViolationsByRequest

Also see

Inherited methods from superclass BaseResponse.

New Since Version

6.4

Related references[BaseObject](#) on page 130

SANscreenAPIFactory

This class simplifies the creation of API connections and sessions for interaction with the OnCommand Insight server.

Methods

The following methods are available from this class:

getEndPoint

Gets access to a SANscreen API endpoint for server interaction and returns the end point.

```
public static SANscreenAPIEndPoint getEndPoint
(String serverHost, String userName, String password) throws Exception
```

Returns

Returns the end point.

openCurrentTimeSessionContext

Opens a session context for interaction with the server. Session is associated to current time upon opening of session. This session encapsulates interaction information and includes the creation time timestamp from which results are consistently retrieved. All queries using this context see only objects that existed at the session creation time. Object addition or removal is returned by queries using this session.

```
public static APISessionContext openCurrentTimeSessionContext
(SANscreenAPIEndPoint apiEndPoint, int maxResponseLimit)
throws RemoteException, APIException
```

Returns

Returns the session context to interact with the connection.

setTrustManager

Allows setting a default trust manager for interaction with the server (optional). Default implementation ships with a trust manager not requiring checking of certificates or host names. The default is trust-all.

```
public static void setTrustManager
(TrustManager trustManager)
```

Returns

Returns the TrustManager.

New Since Version

6.4

Share

Represents a Share in the environment.

Superclass

HistoryObject

```
BaseObject
|__HistoryObject
```

Constructors

public Share

```
public Share(String id, String name, String fileShareId, String
protocol,
String ipInterfaces, String description, long startTime, long
endTime)
```

Methods

The following methods are available from this class:

getDescription

```
public String getDescription()
```

setDescription

```
public void setDescription(String description)
```

getFileShareId

```
public String getFileShareId()
```

setFileShareId

```
public void setFileShareId(String fileShareId)
```

getIpInterfaces

```
public String getIpInterfaces()
```

setIpInterfaces

```
public void setIpInterfaces(String ipInterfaces)
```

getName

```
public String getName()
```

setName

```
public void setName(String name)
```

getProtocol

```
public String getProtocol()
```

setProtocol

```
public void setProtocol(String protocol)
```

Also see

Inherited methods from superclass HistoryObject.

Related references

[HistoryObject](#) on page 141

StorageArray

Represents a storage array in the SAN. You can retrieve storage device information.

Superclass

DeviceBase

```
BaseObject
|__HistoryObject
|__DeviceBase
```

Constructors

Initiates a StorageArray object

```
public StorageArray()
```

Methods

The following methods are available from this class:

getCapacityGB

```
public double getCapacityGB()
```

setCapacityGB

```
public void setCapacityGB(double capacityGB)
```

getMicrocodeVersion

```
public String getMicrocodeVersion()
```

Returns

The microcode version.

setMicrocodeVersion

Should be used by the WSAPI.

```
public void setMicrocodeVersion(String microcodeVersion)
```

getModel

```
public String getModel()
```

Returns

The vendor model.

setModel

The vendor model. Should be used by the WSAPI.

```
public void setModel(String model)
```

getRawCapacityGB

```
public double getRawCapacityGB()
```

Returns

The raw capacity of the storage.

setRawCapacityGB

The raw capacity of the storage. Should be used by the WSAPI.

```
public void setRawCapacityGB(double capacityGB)
```

getSerialNumber

```
public String getSerialNumber()
```

Returns

The serial number of the storage.

setSerialNumber

The serial number of the storage. Should be used by the WSAPI.

```
public void setSerialNumber(String serialNumber)
```

getVendor

```
public String getVendor()
```

Returns

The vendor name, such as EMC or HDS.

setVendor

The vendor name, such as EMC or HDS, etc. Should be used by the WSAPI.

```
public void setVendor(String vendor)
```

Also see

Inherited methods from superclass DeviceBase.

Related references

[DeviceBase](#) on page 133

StorageArrayResponse

A response to a request for StorageArray objects. This class contains an array of StorageArray objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

BaseResponse

Constructors

Initializes a StorageArrayResponse object.

```
public StorageArrayResponse()
```

Methods

The following methods are available from this class:

getArrays

```
public StorageArray[] getArrays()
```

Returns

An array of storages

setArrays

Should be used by the WSAPI.

```
public void setArrays(StorageArray[] arrays)
```

Parameters**arrays**

An array of storages

Also see

Inherited methods from superclass BaseResponse.

Related references

[StorageArray](#) on page 180

[BaseResponse](#) on page 131

StoragePool

This class retrieves the storage pool object for interaction via web services.

Superclass

HistoryObject

```
BaseObject
|__HistoryObject
```

Constructors

Initiates a StoragePool object.

```
public StoragePool()
```

Methods

The following methods are available from this class:

getDataAllocatedCapacityMB

```
public long getDataAllocatedCapacityMB()
```

setDataAllocatedCapacityMB

```
public void setDataAllocatedCapacityMB(long
dataAllocatedCapacityMB)
```

getDataUsedCapacityMB

```
public long getDataUsedCapacityMB()
```

setDataUsedCapacityMB

```
public void setDataUsedCapacityMB(long dataUsedCapacityMB)
```

getDedupeSavings

```
public float getDedupeSavings()
```

setDedupeSavings

```
public void setDedupeSavings(long dedupeSavings)
```

getName

```
public String getName()
```

setName

```
public void setName(String name)
```

getOtherAllocatedCapacityMB

```
public long getOtherAllocatedCapacityMB()
```

setOtherAllocatedCapacityMB

```
public void setOtherAllocatedCapacityMB(long  
otherAllocatedCapacityMB)
```

getOtherUsedCapacityMB

```
public long getOtherUsedCapacityMB()
```

setOtherUsedCapacityMB

```
public void setOtherUsedCapacityMB(long otherUsedCapacityMB)
```

getPhysicalDiskCapacityMB

```
public long getPhysicalDiskCapacityMB()
```

setPhysicalDiskCapacityMB

```
public void setPhysicalDiskCapacityMB(long  
physicalDiskCapacityMB)
```

getRawToUsableRatio

```
public float getRawToUsableRatio()
```

setRawToUsableRatio

```
public void setRawToUsableRatio(float rawToUsableRatio)
```

getRedundancy

```
public String getRedundancy()
```

setRedundancy

```
public void setRedundancy(String redundancy)
```

Returns

Sets the type of redundancy in the storage pool.

getReservedCapacityMB

```
public long getReservedCapacityMB()
```


setReservedCapacityMB

```
public void setReservedCapacityMB(long ReservedCapacityMB)
```

getSnapshotAllocatedCapacityMB

```
public long getSnapshotAllocatedCapacityMB()
```

setSnapshotAllocatedCapacityMB

```
public void setSnapshotAllocatedCapacityMB(long  
snapshotAllocatedCapacityMB)
```

getSnapshotUsedCapacityMB

```
public long getSnapshotUsedCapacityMB()
```

setSnapshotUsedCapacityMB

```
public void setSnapshotUsedCapacityMB(long  
snapshotUsedCapacityMB)
```

getStatus

```
public String getStatus()
```

Returns

The status (for example, offline or online) of the storage pool.

setStatus

```
public void setStatus(String status)
```

getStorageId

```
public String getStorageId()
```

Returns

The ID of the storage system.

setStorageId

The vendor model. Should be used by the WSAPI.

```
public void setStorageId(String storageId)
```

getTotalAllocatedCapacityMB

```
public long getTotalAllocatedCapacityMB()
```

setTotalAllocatedCapacityMB

```
public void setTotalAllocatedCapacityMB(long  
totalAllocatedCapacityMB)
```

getTotalUsedCapacityMB

```
public long getTotalUsedCapacityMB()
```

setTotalUsedCapacityMB

```
public void setTotalUsedCapacityMB(long totalUsedCapacityMB)
```

getType

```
public String getType()
```

Returns

The type of storage pool, for example, thin provisioning, RAID group, Backend RAID Group, or Hybrid-Aggregate.

setType

Should be used by the WSAPI.

```
public void setType(String type)
```

getVendorTier

```
public String getVendorTier()
```

setVendorTier

```
public void setVendorTier(String vendorTier)
```

Returns

Flag to indicate the vendor-specific tier.

isAutoTiering

```
public boolean isAutoTiering()
```

Returns

Flag that indicates whether the storage pool participates in auto tiering (EMC case).

setAutoTiering

```
public void setAutoTiering(boolean autoTiering)
```

isDedupeEnabled

```
public boolean isDedupeEnabled()
```

Returns

Indicates whether the deduplication technology is enabled on the storage pool.

setDedupeEnabled

```
public void setDedupeEnabled(boolean dedupeEnabled)
```

isIncludeInDwhCapacity

```
public boolean isIncludeInDwhCapacity()
```

Returns

Flag that controls which storage pools are computed from Acquisition.

setIncludeInDwhCapacity

```
public void setIncludeInDwhCapacity(boolean includeInDwhCapacity)
```

isRaidGroup

```
public boolean isRaidGroup()
```

Returns

Flag that indicates whether the this is a RAID group.

setRaidGroup

```
public void setRaidGroup(boolean raidGroup)
```

isThinProvisioningSupported

```
public boolean isThinProvisioningSupported()
```

Returns

Indicates that the storage pool supports thin provisioning. This means that volumes on top of the storage pool can be thin provisioned.

setThinProvisioningSupported

```
public void setThinProvisioningSupported(boolean  
thinProvisioningSupported)
```

isUsesSSDCache

```
public boolean isUsesSSDCache()
```

Returns

Flag that indicates whether the storage pool uses Flash pools (NetApp case).

setUsesSSDCache

```
public void setUsesSSDCache(boolean usesSSDCache)
```

isVirtual

```
public boolean isVirtual()
```

Returns

Flag that indicates whether the storage pool is virtual.

setVirtual

```
public void setVirtual(boolean virtual)
```

Also see

Inherited methods from superclass HistoryObject.

New Since Version

6.4

Related references

[HistoryObject](#) on page 141

StoragePoolResponse

A response to a request for storage pool objects. This class contains an array of storage pool objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum returned number of objects allowed.

Superclass

BaseResponse

Constructors

Used by WSAPI only.

```
public StoragePoolResponse()
```

Methods

The following methods are available from this class:

getStoragePools

```
public StoragePool[] getStoragePools()
```

Returns

An array of storage pools.

setStoragePools

Should be used by the WSAPI.

```
public void setStoragePools(StoragePool[] StoragePools)
```

Also see

Inherited methods from superclass BaseResponse.

New Since Version

6.4

Related references

[StoragePool](#) on page 183

[BaseResponse](#) on page 131

Switch

Represents a switch in the SAN. You can retrieve switch information.

Superclass

DeviceBase

```
BaseObject
|__HistoryObject
|__DeviceBase
```

Constructors

Initializes a switch object.

```
public Switch()
```

Methods

The following methods are available from this class:

getModel

```
public String getModel()
```

Returns

The model of the switch.

setModel

The model of the switch. Should be used by the WSAPI.

```
public void setModel(String model)
```

getStatus

```
public String getStatus()
```

Returns

The current status of the switch.

setStatus

The current status of the switch. Should be used by the WSAPI.

```
public void setStatus(String status)
```

getVendor

```
public String getVendor()
```

Returns

The vendor of the switch.

setVendor

The vendor of the switch. Should be used by the WSAPI.

```
public void setVendor(String vendor)
```

getWwn

```
public String getWwn()
```

Returns

The World Wide Name of the switch.

setWwn

The World Wide Name of the switch. Should be used by the WSAPI.

```
public void setWwn(String wwn)
```

Also see

Inherited methods from superclass DeviceBase.

Related references

[DeviceBase](#) on page 133

SwitchResponse

A response to a request for Switch objects. This class contains an array of Switch objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed number of returned objects.

Superclass

BaseResponse

Constructors

Initializes a switch response object.

```
public SwitchResponse()
```

Methods

The following methods are available from this class:

getFabricId

```
public String getFabricId()
```

setFabricId

```
public void setFabricId(String fabricId)
```

getFirmwareVersion

```
public String getFirmwareVersion()
```

setFirmwareVersion

```
public void setFirmwareVersion(String firmwareversion)
```

getSwitches

```
public Switch[] getSwitches()
```

Returns

An array of switches.

setSwitches

The switch object. Should be used by the WSAPI.

```
public void setSwitches(Switch[] switches)
```

Also see

Inherited methods from superclass `BaseResponse`.

Related references

[Switch](#) on page 189

[BaseResponse](#) on page 131

SynchronizationResponse

A response to a request for Synchronization objects. This class contains an array of Synchronization objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

`BaseResponse`

Constructors

```
public SynchronizationResponse()
```

Methods

The following methods are available from this class:

getSynchronizations

```
public Synchronization[] getSynchronizations()
```

setSynchronizations

Should be used by the WSAPI.

```
public void setSynchronizations(Synchronization[]
synchronizations)
```

Also see

Inherited methods from superclass `BaseResponse`.

Tape

Represents a tape storage device in the SAN.

Superclass

`DeviceBase`

```
BaseObject
|__HistoryObject
|__DeviceBase
```

Constructors

Initializes a tape object.

```
public Tape()
```

Methods

The following methods are available from this class:

getModel

```
public String getModel()
```

Returns

The model of the tape device.

setModel

Should be used by the WSAPI.

```
public void setModel(String model)
```

Parameter

model

The model of the tape device.

getSerialNumber

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

Returns

The serial number of the tape device.

setSerialNumber

The serial number of the tape device. Should be used by the WSAPI.

```
public void setSerialNumber(String serialNumber)
```

getVendor

```
public String getVendor()
```

Returns

The vendor of the tape device.

setVendor

The vendor of the tape device. Should be used by the WSAPI.

```
public void setVendor(String vendor)
```

Also see

Inherited methods from superclass DeviceBase.

TapeResponse

A response to a request for Tape objects. This class contains an array of Tape objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

BaseResponse

Constructors

Initializes a violation object.

```
public TapeResponse()
```

Methods

The following methods are available from this class:

getTapes

```
public Tape[] getTapes()
```

Returns

An array of tape objects.

setTapes

Should be used by the WSAPI.

```
public void setTapes(Tape[] tapes)
```

Also see

Inherited methods from superclass BaseResponse.

Related references

[Tape](#) on page 192

[getTapes](#) on page 82

[BaseResponse](#) on page 131

Task

A task data object.

Superclass

BaseObject

Constructors

```
public Task()
```

Methods

The following methods are available from this class:

getComments

```
public String getComments()
```

setComments

Should be used by the WSAPI.

```
public void setComments(String comments)
```

getDependedTaskId

```
public String getDependedTaskId()
```

setDependedTaskId

Should be used by the WSAPI.

```
public void setDependedTaskId(String dependedTaskId)
```

getEndTime

```
public long getEndTime()
```

setEndTime

Should be used by the WSAPI.

```
public void setEndTime(long endTime)
```

getErrorsCount

```
public int getErrorsCount()
```

setErrorsCount

Should be used by the WSAPI.

```
public void setErrorsCount(int errorsCount)
```

getName

```
public String getName()
```

setName

Should be used by the WSAPI.

```
public void setName(String name)
```

getOwner

```
public String getOwner()
```

setOwner

Should be used by the WSAPI.

```
public void setOwner(String owner)
```

getReferenceNumber

```
public String getReferenceNumber()
```

setReferenceNumber

Should be used by the WSAPI.

```
public void setReferenceNumber(String referenceNumber)
```

getStartTime

```
public long getStartTime()
```

setStartTime

Should be used by the WSAPI.

```
public void setStartTime(long startTime)
```

getState

```
public String getState()
```

setState

Should be used by the WSAPI.

```
public void setState(String state)
```

getStatus

```
public String getStatus()
```

setStatus

Should be used by the WSAPI.

```
public void setStatus(String status)
```

getUserTaskId

```
public String getUserTaskId()
```

setUserTaskId

Should be used by the WSAPI.

```
public void setUserTaskId(String userTaskId)
```

getViolationsCount

```
public int getViolationsCount()
```

setViolationsCount

Should be used by the WSAPI.

```
public void setViolationsCount(int violationsCount)
```

Also see

Inherited methods from superclass `BaseObject`.

Related references

[BaseObject](#) on page 130

TaskResponse

Gets all planning tasks. A response to a request for Task objects. This class contains an array of Task objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

`BaseResponse`

Constructors

```
public TaskResponse()
```

Methods

The following methods are available from this class:

getTasks

```
public String getTasks()
```

setTasks

Should be used by the WSAPI.

```
public void setTasks(String details)
```

Also see

Inherited methods from superclass `BaseResponse`.

Related references

[Task](#) on page 194

[BaseResponse](#) on page 131

TaskViolationResponse

Gets future violations for a given task. A response to a request for `TaskViolation` objects. This class contains an array of `TaskViolation` objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

`BaseResponse`

Constructors

```
public TaskViolationResponse()
```

Methods

The following methods are available from this class:

getTaskViolations

```
public TaskViolation[] getTaskViolations()
```

setTaskViolations

Should be used by the WSAPI.

```
public void setTaskViolations(TaskViolation[] taskViolations)
```

Also see

Inherited methods from superclass `BaseResponse`.

Related references

[BaseResponse](#) on page 131

Violation

A SAN violation.

Superclass

HistoryObject

```
BaseObject
|__HistoryObject
```

Constructors

Initializes a violation object.

```
public Violation()
```

Methods

The following methods are available from this class:

getArrayId

```
public String getArrayId()
```

setArrayId

Should be used by the WSAPI.

```
public void setArrayId(String arrayId)
```

getHostId

```
public String getHostId()
```

setHostId

Should be used by the WSAPI.

```
public void setHostId(String hostId)
```

setType

Should be used by the WSAPI.

```
public void setType(String type)
```

getType

Gets the Violation Type.

```
public String getType()
```

The following Violation Types are included:

getVolumeId

```
public String getVolumeId()
```

setVolumeId

Should be used by the WSAPI.

```
public void setVolumeId(String volumeId)
```

isBackend

```
public boolean isBackend()
```

setBackend

```
public void setBackend (boolean backend)
```

The following Violation Types are included:

Also see

Inherited methods from superclass HistoryObject.

Violation Type	Corresponding UI Field	Description
EXIST_NOT_APPROVED	Unauthorized path	Path doesn't have policy assigned to it.
APPROVE_NOT_EXIST	Path Outage	There is an existing policy and/or violation for the path but path itself went down.
NO_SPF	Single Point of Failure	A path policy requires NO single point of failure between the host and volume, but in reality SPF exists.
REDUNDANCY_VIOLATION	Missing Redundancy	A path policy requires dual fabric redundancy between host and volume, but in reality there is only one fabric.
HOPS_VIOLATION	Switch Hop Count	The number of switch hops specified in the path policy is less than the actual number of switch hops between host and volume.
HOST_PORTS_VIOLATION	Missing Active Host Ports	The number of host ports specified in the path policy is greater than the actual number of host ports.
SHARE_VOLUME_SCOPE_VIOLATION	Unauthorized Sharing	The sharing model set in the path policy doesn't match the existing volume sharing conditions. For example, if the path policy has "Application" model of volume sharing (which means only hosts from the same application are allowed to share certain volume, but volume is configured to be available to any host) sharing violation will be created.
STORAGE_PORTS_VIOLATION	Missing Active Storage Ports	The number of storage ports specified in the path policy is greater than the actual number of storage ports.
BACKEND_OUTAGE_VIOLATION	Virtual Volume Missing Backend Support	A path from the host to the virtual volume doesn't have a corresponding backend link (backend LUN).

Violation Type	Corresponding UI Field	Description
ISCSI_NUMBER_OF_SESSIONS_VIOLATION	Session Count	The number of iSCSI sessions defined in the iSCSI path policy is greater than the actual number of iSCSI sessions.
ISCSI_NUMBER_OF_CONNECTIONS_VIOLATION	Connection Count	The number of iSCSI connections defined in the iSCSI path policy is greater than the actual number of iSCSI connections.
ISCSI_SECURITY_VIOLATION	Missing Security	In case security is required by the iSCSI path policy and if security settings for at least one iSCSI session corresponding to the path are not strong (both inbound and outbound keywords should be present), then the iSCSI security violation is generated.
HV_ACTIVE_PATH_CONFLICT_VIOLATION	Active Path Conflict	All storage ports that are part of the same path from ESX server to storage/ volume do NOT share the same storage processor, but the policy is turned on for the single storage processor for volume active paths.
HV_MISSING_VIRTUAL_CLUSTER_PATH_VIOLATION	Missing Virtual Cluster Paths	A volume is NOT accessed by all ESX hosts from the same virtual cluster, but the policy is turned on for identical volume access for hosts within a virtual cluster.
HV_INCONSISTENT_LUNS_VIOLATION	Inconsistent LUNS	Different LUN numbers exist for the same volume accessed by ESX hosts from the same virtual cluster, but the policy is turned on for identical volume LUN numbers for hosts within a virtual cluster.
HV_MISSING_VIRTUAL_CLUSTER_NAS_PATH_VIOLATION	Missing Virtual Cluster NAS Share	A NAS share is NOT accessed by all ESX hosts from the same virtual cluster, but the policy is turned on for identical NAS share access for hosts within a virtual cluster.

New Since Version

6.0. Addition of Violation Types to getType method documentation.

Related references

[HistoryObject](#) on page 141

ViolationResponse

A response to a request for Violation objects. This class contains an array of Violation objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

BaseResponse

Constructors

Initializes a volume response object.

```
public ViolationResponse()
```

Methods

The following methods are available from this class:

getViolations

```
public Violation[] getViolations()
```

setViolations

Should be used by the WSAPI.

```
public void setViolations(Violation[] violations)
```

Also see

Inherited methods from superclass `BaseResponse`.

Related references

[Violation](#) on page 198

[getViolations](#) on page 83

[BaseResponse](#) on page 131

VirtualMachine

Virtual machines in the system.

Superclass

HistoryObject

```
BaseObject
|__HistoryObject
```

Constructors

Initiates the virtual machine object.

```
public VirtualMachine()
```

Methods

The following methods are available from this class:

getDataStoreId

Gets the ID of the data source.

```
public String getDataStoreId()
```

setDataStoreId

Should be used by the WSAPI.

```
public void setDataStoreId(String dataStoreId)
```

getDnsName

Gets the DNS name.

```
public String getDnsName()
```

setDnsName

Should be used by the WSAPI.

```
public void setDnsName(String dnsName)
```

getGuestState

Gets the state of the guest machine.

```
public String getGuestState()
```

setGuestState

Should be used by the WSAPI.

```
public void setGuestState(String guestState)
```

getHostId

Gets the ID of the host associated with this virtual machine.

```
public String getHostId()
```

setHostId

Should be used by the WSAPI.

```
public void setHostId(String hostId)
```

getIps

Gets the IP address.

```
public String getIps()
```

setIps

Should be used by the WSAPI.

```
public void setIps(String ips)
```

getMemory

Gets the virtual machine's memory information.

```
public String getMemory()
```

setMemory

Should be used by the WSAPI.

```
public void setMemory(String memory)
```

getMoid

Gets the virtual machine ID.

```
public String getMoid()
```

setMoid

Should be used by the WSAPI.

```
public void setMoid(String moid)
```

getName

Gets the virtual machine name.

```
public String getName()
```

setName

Should be used by the WSAPI.

```
public void setName(String name)
```

getOs

Gets the virtual machine's operating system.

```
public String getOs()
```

setOs

Should be used by the WSAPI.

```
public void setOs(String os)
```

getPowerState

Gets the state of the virtual machine's power.

```
public String getPowerState() {
```

setPowerState

Should be used by the WSAPI.

```
public void setPowerState(String powerState)
```

getPowerStateChangeTime

Gets the time of a change to the virtual machine's power state.

```
public String getPowerStateChangeTime()
```

setPowerStateChangeTime

Should be used by the WSAPI.

```
public void setPowerStateChangeTime(String powerStateChangeTime)
```

getProcessors

Gets the virtual machine's processor information.

```
public String getProcessors()
```

setProcessors

Should be used by the WSAPI.

```
public void setProcessors(String processors)
```

Also see

Inherited methods from superclass HistoryObject.

New Since Version

6.0

VirtualMachineResponse

A response to a request for VirtualMachine objects. This class contains an array of VirtualMachine objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

BaseResponse

Constructors

Initiates the virtual machine response object.

```
public VirtualMachineResponse()
```

Methods

The following methods are available from this class:

getVirtualMachines

```
public VirtualMachine[] getVirtualMachines()
```

Returns

The virtual machines contained in the response.

setVirtualMachines

The virtual machines in the response. Should only be used by WSAPI.

```
public void setVirtualMachines(VirtualMachine[] virtualMachines)
```

Also see

Inherited methods from superclass BaseResponse.

New Since Version

6.0

Related references[VirtualMachine](#) on page 201[getVirtualMachine](#) on page 86[BaseResponse](#) on page 131

Volume

Represents a volume in the SAN.

Superclass

HistoryObject

```
BaseObject
|__HistoryObject
```

Constructors

Initializes a volume object.

```
public Volume()
```

Methods

The following methods are available from this class:

getCapacityGB

```
public double getCapacityGB()
```

Returns

The capacity of the volume.

setCapacityGB

The capacity of the volume. Should be used by the WSAPI.

```
public void setCapacityGB(double capacityGB)
```

getConsumedCapacityGB

```
public double getConsumedCapacityGB()
```

setConsumedCapacityGB

```
public void setConsumedCapacityGB(double consumedCapacityGB)
```

getDiskSize

```
public String getDiskSize()
```

setDiskSize

```
public void setDiskSize(String diskSize)
```

getDiskSpeed

```
public String getDiskSpeed()
```

setDiskSpeed

```
public void setDiskSpeed(String diskSpeed)
```

getDiskType

```
public String getDiskType()
```

Returns

The disk type.

setDiskType

Should be used by the WSAPI.

```
public void setDiskType(String diskType)
```

getId

Gets the unique identifier of this object.

```
public String getId()
```

Returns

The unique identifier of this object.

setId

The unique identifier for this object. Should only be used by WSAPI.

```
public void setId(String id)
```

getLabel

```
public String getLabel()
```

setLabel

```
public void setLabel(String label)
```

getName

```
public String getName()
```

Returns

The name of the volume.

setName

The name of the volume. Should be used by the WSAPI.

```
public void setName(String name)
```

getRawCapacityGB

```
public double getRawCapacityGB()
```

Returns

The raw capacity of the volume.

setRawCapacityGB

The raw capacity of the volume. Should be used by the WSAPI.

```
public void setRawCapacityGB(double rawCapacityGB)
```

getRedundancy

```
public String getRedundancy()
```

Returns

The volume redundancy.

setRedundancy

The volume redundancy. Should be used by the WSAPI.

```
public void setRedundancy(String redundancy)
```

getStorageID

```
public String getStorageID()
```

Returns

The storage ID.

setStorageID

The storage ID. Should be used by the WSAPI.

```
public void setStorageID(String storageID)
```

getType

```
public String getType()
```

Returns

The type of volume.

setType

The volume type. Should be used by the WSAPI.

```
public void setType(String type)
```

isVirtual

```
public boolean isVirtual()
```

setVirtual

```
public void setVirtual(Boolean virtual)
```

Also see

Inherited methods from superclass `HistoryObject`.

Related references

[HistoryObject](#) on page 141

VolumeMap

Represents mapping between a storage unit volume and a port on the same storage unit. In addition, you can use a volume mask to prevent certain ports from accessing the volume.

Superclass

`BaseObject`

Constructors

Initializes a volume map object.

```
public VolumeMap()
```

Methods

The following methods are available from this class:

getLun

```
public String getLun()
```

Returns

The logical unit number used by the host to access this volume.

setLun

The logical unit number used by the host to access this volume. Should only be used by WSAPI.

```
public void setLun(String lun)
```

getProtocolController

The protocol controller through which the volume is mapped to the storage port.

```
public String getProtocolController()
```

Returns

The protocol controller through which the volume is mapped to the storage port.

setProtocolController

```
public void String setProtocolController(String
protocolController)
```

getStoragePortId

```
public String getStoragePortId()
```

Returns

The port on the storage device to which the volume is mapped.

setStoragePortId

The port on the storage device to which the volume is mapped. Should only be used by WSAPI.

```
public void setStoragePortId(String storagePortId)
```

getStoragePortWwn

```
public String getStoragePortWwn()
```

Returns

The World Wide Name for the port to which the volume is mapped.

setStoragePortWwn

The World Wide Name for the port to which the volume is mapped. Should only be used by WSAPI.

```
public void setStoragePortWwn(String storagePortWwn)
```

getVolumeId

```
public String getVolumeId()
```

Returns

The ID of the volume being masked to a specified disk storage port.

setVolumeId

The ID of the volume being masked to a specified disk storage port. Should be used by the WSAPI.

```
public void setVolumeId(String volumeId)
```

isWwnsecurityvalidity

Validates the WWNs of the ports accessing a volume to ensure they are masked for that volume.

```
public boolean isWwnsecurityvalidity()
```

Returns

When this flag is true, the system validates the WWNs of the ports accessing a volume to ensure they are masked for that volume. The port WWN is stored in the

storagePortWwn member in the VolumeMask class. If set to false, the system ignores the WWNs of ports access a volume and therefor eliminates the need for masking.

setWwnsecurityvalidity

Validates the WWNs of the ports accessing a volume to ensure they are masked for that volume. Should be used by the WSAPI.

When this flag is true, the system validates the WWNs of the ports accessing a volume to ensure they are masked for that volume. The port WWN is stored in the storagePortWwn member in the VolumeMask class. If set to false, the system ignores the WWNs of ports access a volume and therefor eliminates the need for masking.

```
public void setWwnsecurityvalidity(boolean wwnsecurityvalidity)
```

Also see

Inherited methods from superclass BaseObject.

Related references

[VolumeMask](#) on page 211

[BaseObject](#) on page 130

VolumeMapResponse

A response to a request for VolumeMap objects. This class contains an array of VolumeMap objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed returned number of objects.

Superclass

BaseResponse

Constructors

Initializes a volume map response object

```
public VolumeMapResponse()
```

Methods

The following methods are available from this class:

getVolumeMaps

```
public VolumeMap[] getVolumeMaps()
```

Returns

An array of volume maps.

setVolumeMaps

Should only be used by WSAPI.

```
public void setVolumeMaps(VolumeMap[] volumeMaps)
```

Parameter

volumeMaps

An array of volume maps.

Also see

Inherited methods from superclass `BaseResponse`.

Related references

[VolumeMap](#) on page 208

[BaseResponse](#) on page 131

VolumeMask

Adds masking between a volume (disk storage port) and a host.

Superclass

`BaseObject`

Constructors

Initializes a volume mask object.

```
public VolumeMask()
```

Methods

The following methods are available from this class:

getInitiatorPortOrNodeWwn

```
public String getInitiatorPortOrNodeWwn()
```

Returns

The host port or adapter to which the storage volume is masked.

setInitiatorPortOrNodeWwn

The host port or adapter to which the storage volume is masked. Should only be used by WSAPI.

```
public void setInitiatorPortOrNodeWwn(String
initiatorPortOrNodeWwn)
```

getProtocolController

```
public String getProtocolController()
```

Returns

The protocol controller through which the volume is mapped to the storage port.

setProtocolController

The protocol controller through which the volume is mapped to the storage port. Should only be used by WSAPI.

```
public void setProtocolController(String protocolController)
```

getStoragePortId

```
public String getStoragePortId()
```

Returns

The ID of the storage port from which the volume is accessible to the host initiator.

setStoragePortId

The ID of the storage port from which the volume is accessible to the host initiator.
Should only be used by WSAPI.

```
public void setStoragePortId(String storagePortId)
```

getStoragePortWwn

The World Wide Name of the storage port.

```
public String getStoragePortWwn()
```

setStoragePortWwn

The World Wide Name of the storage port. Should only be used by WSAPI.

```
public void setStoragePortWwn(String storagePortWwn)
```

getVolumeId

```
public String getVolumeId()
```

Returns

The ID for the volume to be masked to the host adaptor or port.

setVolumeId

The identifier for the volume to be masked to the host adaptor or port. Should only be used by WSAPI.

```
public void setVolumeId(String volumeId)
```

Also see

Inherited methods from superclass `BaseObject`.

Related references

[BaseObject](#) on page 130

VolumeMaskResponse

A response to a request for VolumeMask objects. This class contains an array of VolumeMask objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed number of returned objects.

Superclass

`BaseResponse`

Constructors

Initializes a volume mask response object.

```
public VolumeMaskResponse()
```

Methods

The following methods are available from this class:

getMasks

```
public VolumeMask[] getMasks()
```

Returns

The masks known to OnCommand Insight.

setMasks

The masks known to OnCommand Insight. Should only be used by WSAPI.

```
public void setViolations(VolumeMask[] masks)
```

Also see

Inherited methods from superclass BaseResponse.

Related references

[VolumeMask](#) on page 211

[getVolumeMasksByStorageArray](#) on page 89

[BaseResponse](#) on page 131

VolumeResponse

A response to a request for Volume objects. This class contains an array of Volume objects and contains an iterator to retrieve remaining objects when the request involves more than the maximum allowed number of returned objects.

Superclass

BaseResponse

Constructors

Initializes a volume response object.

```
public VolumeResponse()
```

Methods

The following methods are available from this class:

getVolumes

```
public Volume[] getVolumes()
```

Returns

An array of volumes.

setVolumes

An array of volumes. Should only be used by WSAPI.

```
public void setVolumes(Volume[] volumes)
```

Also see

Inherited methods from superclass BaseResponse.

Related references

[Volume](#) on page 205

[getVolumes](#) on page 90

[getVolumesByStorageArray](#) on page 90

[BaseResponse](#) on page 131

VolumesSynchronizationState

Objects containing data for the synchronization state between two volumes that are part of replication process.

Superclass

BaseObject

Constructors

```
public VolumesSynchronizationState()
```

Methods

The following methods are available from this class:

getDetailedSynchronizationState

```
public int getDetailedSynchronizationState()
```

setDetailedSynchronizationState

Should be used by the WSAPI.

```
public void setDetailedSynchronizationState(int  
detailedSynchronizationState)
```

getSourceStorageId

```
public String getSourceStorageId()
```

setSourceStorageId

Should be used by the WSAPI.

```
public void setSourceStorageId(String sourceStorageId)
```

getSourceVolumeId

```
public String getSourceVolumeId()
```

setSourceVolumeId

Should be used by the WSAPI.

```
public void setSourceVolumeId(String sourceVolumeId)
```

getTargetStorageId

```
public String getTargetStorageId()
```

setTargetStorageId

Should be used by the WSAPI.

```
public void setTargetStorageId(String targetStorageId)
```

getTargetVolumeId

```
public String getTargetVolumeId()
```

setTargetVolumeId

Should be used by the WSAPI.

```
public void setTargetVolumeId(String targetVolumeId)
```

getTimeStamp

```
public long getTimeStamp()
```

setTimeStamp

Should be used by the WSAPI.

```
public void setTimeStamp(long timeStamp)
```

getVolumesSynchronizationState

```
public String getVolumesSynchronizationState()
```

setVolumesSynchronizationState

Should be used by the WSAPI.

```
public void setVolumesSynchronizationState(String  
volumesSynchronizationState)
```

Also see

Inherited methods from superclass `BaseObject`.

Related references

[BaseObject](#) on page 130

VolumesSynchronizationStateResponse

Container for synchronization states for replicated volumes.

Superclass

BaseResponse

Constructors

```
public VolumesSynchronizationStateResponse()
```

Methods

The following methods are available from this class:

getVolumesSynchronizationState

```
public VolumesSynchronizationState[]  
getVolumesSynchronizationState()
```

setVolumesSynchronizationState

Should only be used by WSAPI.

```
public void  
setVolumesSynchronizationState(VolumesSynchronizationState[]  
volumesSynchronizationState)
```

Also see

Inherited methods from superclass BaseResponse.

Related references

[VolumesSynchronizationState](#) on page 214

[BaseResponse](#) on page 131

Web services WSDL

WSDL is an XML-based service that communicates using web services. To implement WSDL in OnCommand Insight, see the WSDL procedures outlined in the WSDL specification and the WSDL file.

Accessing the WSDL specification

The WSDL specification describes an XML format used to communicate using web services. You can obtain the WSDL specification file from within the OnCommand Insight Administration web portal.

Steps

1. Log into the OnCommand Insight Administration web portal as an administrator.
2. From the Advanced menu, select **Insight Connect API**.
3. From the OnCommand Insight Connect API page, select **WSDL File**.

Connect API web services WSDL

Web Service Definition Language (WSDL) is an XML-based service description of how to communicate using web services. WSDL defines services as collections of network endpoints or ports. The WSDL specification provides an XML format for documents for this purpose. WSDL describes the public interface to the web service.

The abstract definition of ports and messages is separated from their concrete use or instance, which allows for the reuse of these definitions:

- A port is defined by associating a network address with a reusable binding, and a collection of ports defines a service.
Port types are abstract collections of supported operations.
- Messages are abstract descriptions of the data being exchanged.

The concrete protocol and data format specifications for a particular port type constitute a reusable binding, where the messages and operations are then bound to a concrete network protocol and message format.

OnCommand Insight Connect API code sample scripts and packages

You might find it beneficial to review some Connect API Java scripts and Perl script package examples.

Java is the only API supported by NetApp. Perl API and other examples are provided for reference only with no support.

To see the examples provided in the OnCommand Insight portal, click **Insight Connect API > Examples**.

Related tasks

[Accessing OnCommand Insight Connect API examples](#) on page 16

Java examples

The following examples are included with OnCommand Insight.

AnnotationGetSet

An example of how to retrieve and/or set annotations on devices in OnCommand Insight.

AnnotationImportExample

An example of how to set annotations on objects in OnCommand Insight using information defined in annotation file.

APIExamples

Examples of how to extract inventory data from OnCommand Insight. Includes extraction of information like storage arrays, paths, violations, and policies.

ApplicationAndBusinessEntity

An example of how to create business entity information, application information and how to associate business entities and applications to hosts in OnCommand Insight.

CapacityRequestExample

An example of how to manage requests/requirements as part of provisioning manager product.

CSVExample

An example of how to extract inventory information from the system and generate CSV files with this information. Includes information on connected ports, replica information, volume information, device information.

IdentificationImportExample

An example of how to perform identification of objects in OnCommand Insight using information defined in an identification file.

ImportHostApplicationsExample

An example of how to perform associations between host and applications in OnCommand Insight using information defined in an host application mapping file.

InventoryDumpMain

An example of how to extract inventory data and generate CSV files. Includes generation of host port, host volume, storage, storage pool, internal volume, volume, switch, switch port information.

PlanningExample

An example of extraction, analysis and validation of planning tasks within OnCommand Insight.

RpoExample

An example of extraction of replication state for volumes within OnCommand Insight.

Perl examples

Sample scripts and corresponding packages are provided as examples. Perl scripts cannot be executed without the appropriate packages or libraries.

The following examples are included with OnCommand Insight.

computeTotalStoragePerArray.pl

An example of how to iterate through all the volumes of all storage device and sum up their sizes to determine the capacity of each storage.

findDisconnectedHosts.pl

An example of determining the list of hosts that are not part of a path.

getPaths.pl

An example of how to iterate through all the paths in the system. Queries the local host for all paths using iterations.

getStorage.pl

An example of how to retrieve all storage devices in the system without making use of iteration.

To make the examples work, download the following libraries and their dependencies from a location, such as the Comprehensive Perl Archive Network (www.cpan.org):

- Ssleay: SSL library for Windows
- SOAP::LITE: Basic implementation of SOAP

The following packages are provided and necessary to support the scripts listed above.

APISessionContext.pm

The Perl module representing the API session context for interaction with the server.

APIVersion.pm

The Perl module representing an API version to interact with server.

Host.pm

The Perl module representing hosts in the system.

Iterator.pm

The Perl module for the iterator allowing iteration over results objects being returned by server calls.

Path.pm

The Perl module representing a path in the system.

SOAPParam.pm

The Perl module for SOAP parameter interaction.

StorageArray.pm

The Perl module representing a storage array in the system.

Timeout.pm

The Perl module for time out definition for interactions with system.

Volume.pm

The Perl module representing a volume in the system.

Perl examples are provided for reference purposes only and are not actively supported.

Copyright information

Copyright © 1994–2016 NetApp, Inc. All rights reserved. Printed in the U.S.

No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark information

NetApp, the NetApp logo, Go Further, Faster, AltaVault, ASUP, AutoSupport, Campaign Express, Cloud ONTAP, Clustered Data ONTAP, Customer Fitness, Data ONTAP, DataMotion, Fitness, Flash Accel, Flash Cache, Flash Pool, FlashRay, FlexArray, FlexCache, FlexClone, FlexPod, FlexScale, FlexShare, FlexVol, FPolicy, GetSuccessful, LockVault, Manage ONTAP, Mars, MetroCluster, MultiStore, NetApp Insight, OnCommand, ONTAP, ONTAPI, RAID DP, RAID-TEC, SANtricity, SecureShare, Simplicity, Simulate ONTAP, Snap Creator, SnapCenter, SnapCopy, SnapDrive, SnapIntegrator, SnapLock, SnapManager, SnapMirror, SnapMover, SnapProtect, SnapRestore, Snapshot, SnapValidator, SnapVault, StorageGRID, Tech OnTap, Unbound Cloud, and WAFL and other names are trademarks or registered trademarks of NetApp, Inc., in the United States, and/or other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. A current list of NetApp trademarks is available on the web at <http://www.netapp.com/us/legal/netapptmlist.aspx>.

How to send comments about documentation and receive update notifications

You can help us to improve the quality of our documentation by sending us your feedback. You can receive automatic notification when production-level (GA/FCS) documentation is initially released or important changes are made to existing production-level documents.

If you have suggestions for improving this document, send us your comments by email to doccomments@netapp.com. To help us direct your comments to the correct division, include in the subject line the product name, version, and operating system.

If you want to be notified automatically when production-level documentation is released or important changes are made to existing production-level documents, follow Twitter account @NetAppDoc.

You can also contact us in the following ways:

- NetApp, Inc., 495 East Java Drive, Sunnyvale, CA 94089 U.S.
- Telephone: +1 (408) 822-6000
- Fax: +1 (408) 822-4501
- Support telephone: +1 (888) 463-8277

Index

A

- Action class [118](#)
- ActionResponse class [119](#)
- addHostToApplication method [25](#)
- addHostToApplicationById method [26](#)
- addReservationRequests method [26](#)
- addReservationRequirements method [27](#)
- addVmToApplication method [28](#)
- AnnotationEnumValue class [120](#)
- annotations
 - enumeration values [120](#)
 - importing [17](#)
 - setting [101](#)
 - types [121](#)
 - values [123](#)
- AnnotationType class [121](#)
- AnnotationValue class [123](#)
- API servers
 - failure [124](#)
 - query requirements [125](#)
- API sessions [178](#)
- APIBadRequestException [117](#)
- APIException class [124](#)
- APISessionContext class [125](#)
- Application class [126](#)
- ApplicationResponse class [129](#)
- applications
 - associated hosts [25](#)
 - business entities [106](#)
 - data about [126](#)
 - hosts [26](#)
 - hosts, removing [95](#)
 - hosts/servers [52](#)
 - internal volumes [33, 97, 103](#)
 - qtrees [34, 98, 104](#)
 - response [129](#)
 - retrieving [31, 33, 36](#)
 - setting [102](#)
 - shares [35, 99, 104](#)
 - VMs [28, 37, 100](#)
 - volumes [36, 101, 105](#)
- Attribute class [129](#)
- authentication
 - Connect API [14](#)

B

- BaseObject class [130](#)
- BaseResponse class [131](#)
- best practices
 - changing default passwords [16](#)
- business entities
 - data about [132](#)
 - ports [39, 107](#)
 - storage [38–40, 107, 108](#)
 - switches [41, 109](#)
- BusinessEntity class [132](#)

C

- capacity management
 - request requirement [75](#)
 - request requirements [27, 75](#)
 - requests [26, 73](#)
 - retrieving requests [74](#)
- classes
 - added [12](#)
 - changes to [13](#)
 - list of [117](#)
 - new [10](#)
- closeSession method [28](#)
- comments
 - how to send feedback about documentation [223](#)
- Connect API
 - classes [117](#)
 - examples [16](#)
 - licensing [14](#)
 - main object types [14](#)
 - overview [9](#)
 - prerequisites [14](#)
 - sample project [19–24](#)
 - sessions [15](#)

D

- deprecated methods [12](#)
- device groups
 - response [136](#)
 - retrieving [43–46](#)
- DeviceBase class [133](#)
- DeviceGroup class [134](#)
- DeviceGroupResponse class [136](#)
- disaster recovery
 - path object [136](#)
- dismissPathOutageViolations method [29](#)
- documentation
 - how to receive automatic notification of changes to [223](#)
 - how to send feedback about [223](#)
 - URL [18](#)
- DR paths
 - retrieving by volumes [47](#)
- DRPath class [136](#)
- DRPathResponse class [138](#)

E

- examples of Connect API code [16](#)

F

- fabrics
 - by port IDs [48](#)
- feedback
 - how to send comments about documentation [223](#)
- find objects

- by IP [30](#)
- by WWN [30](#)
- findObjectByWwn method [30](#)
- findObjectsByIp method [30](#)

G

- Generic class [139](#)
- GenericResponse class [140](#)
- generics
 - device retrieval [48, 49](#)
 - response [140](#)
- getAllApplications method [31](#)
- getAPIVersion method [32](#)
- getApplication method [33](#)
- getApplicationOfInternalVolume method [33](#)
- getApplicationOfQtree method [34](#)
- getApplicationOfShare method [35](#)
- getApplicationOfVolume method [36](#)
- getApplicationsOfHost method [36](#)
- getApplicationsOfVm method [37](#)
- getBusinessEntities method [38](#)
- getBusinessEntity method [39](#)
- getBusinessEntityOfPort method [39](#)
- getBusinessEntityOfStorage method [40](#)
- getClosedViolations method [41](#)
- getConnectedPorts method [42](#)
- getDeviceGroup method [43](#)
- getDeviceGroups method [44](#)
- getDeviceGroupsByStorage method [44](#)
- getDeviceGroupsByVolume method [45](#)
- getDeviceGroupVolume method [46](#)
- getDRPathsByVolume method [47](#)
- getFabricsByPorts method [48](#)
- getGeneric method [48](#)
- getGenerics method [49](#)
- getHost method [50](#)
- getHostPolicy method [50](#)
- getHosts methods [51](#)
- getHostsOfApplication method [52](#)
- getInternalVolume methods [53](#)
- getInternalVolumes method [53](#)
- getInternalVolumesByStorageArray method [54](#)
- getInternalVolumesByStoragePool method [115](#)
- getLogicalPortsByLogicalSwitch methods [55](#)
- getLogicalPortsByPhysicalPort method [56](#)
- getLogicalPortsByPhysicalSwitch methods [57](#)
- getLogicalSwitchesByPhysicalSwitch method [58](#)
- getNodesOfDevice method [58](#)
- getPathPolicy method [59](#)
- getPaths method [60](#)
- getPathsByHost method [61](#)
- getPendingPathReservations method [62](#)
- getPhysicalPortByLogicalPort method [62](#)
- getPolicies method [63](#)
- getPort method [64](#)
- getPorts method [65](#)
- getPortsOfDevice method [66](#)
- getPortsOfDeviceType method [66](#)
- getPortsOfNode method [67](#)
- getQtrees method [68](#)
- getRecentDevice method [69](#)
- getRecentHost method [70](#)

- getRecentStorageArray method [70](#)
- getRecentSwitch method [71](#)
- getRecentTape method [72](#)
- getRecentVolume method [73](#)
- getReservationRequests method [73](#)
- getReservationRequestsByStates method [74](#)
- getReservationRequirement method [75](#)
- getReservationRequirements method [75](#)
- getReservationViolationTypeNames method [76](#)
- getSANscreenVersion method [77](#)
- getShares method [77](#)
- getStorageArray method [78](#)
- getStorageArrays method [79](#)
- getStoragePools method [112, 114](#)
- getStoragePoolsByStorageArray method [113](#)
- getSwitch method [79](#)
- getSwitches method [80](#)
- getTape method [81](#)
- getTapes method [82](#)
- getViolations method [83](#)
- getViolationsByHost method [83](#)
- getViolationsByRequests method [84](#)
- getViolationsByReservations method [85](#)
- getVirtualMachine method [86](#)
- getVirtualMachines method [86](#)
- getVolume method [87](#)
- getVolumeMapsByStorageArray method [88](#)
- getVolumeMasksByStorageArray method [89](#)
- getVolumes method [90](#)
- getVolumesByStorageArray method [90](#)
- getVolumesByStoragePool method [114](#)
- getVolumesSynchStatesByDRPath method [91](#)
- getVolumesSynchStatesBySourceVolume method [92](#)
- getZonesByPorts method [93](#)

H

- HistoryObject class [141](#)
- HostResponse class [142](#)
- hosts
 - applications [26](#)
 - class [142](#)
 - logical paths [61](#)
 - removing from applications [94, 95](#)
 - removing policies on [96](#)
 - response [142](#)
 - retrieving [50, 51](#)
 - retrieving policies for [50](#)
 - setting policies [110](#)

I

- information
 - how to send feedback about improving documentation [223](#)
- internal volumes
 - applications [33, 97, 103](#)
 - data about [143](#)
 - response [148](#)
 - retrieving [53, 54, 115](#)
- InternalVolume class [143](#)
- InternalVolumeResponse class [148](#)

iterators
 request [167](#)
 use of [15](#)

L

licenses [14](#)
 logging in [16](#)
 logical paths
 response [156](#)
 retrieving [60](#), [61](#)
 logical ports
 response [149](#)
 retrieving [149](#)
 retrieving by physical ports [56](#)
 retrieving by session context [42](#)
 retrieving by switch [55](#), [57](#)
 logical switches
 retrieving by physical switch [58](#)
 LogicalPort class [149](#)
 LogicalPortResponse class [149](#)
 lpath reservations
 retrieving pending [62](#)

M

masks
 getting [89](#), [99](#)
 methods
 added [10](#)
 deprecated [10](#), [12](#)
 new [10](#)

N

Node class [150](#)
 NodeResponse class [152](#)
 nodes
 response [152](#)
 retrieving [58](#), [150](#)

O

OnCommand Insight
 logging in [16](#)
 opening
 OnCommand Insight Administration portal [16](#)
 openSession method [94](#)

P

passwords
 default [16](#)
 Path class [153](#)
 path outages
 violations [29](#)
 path reservations
 response [155](#)
 PathReservation class [154](#)
 PathReservationResponse class [155](#)
 PathResponse class [156](#)

paths
 policies [59](#)
 reservation [154](#)
 response [156](#)
 retrieval [153](#)
 policies
 data about [157](#)
 paths [59](#)
 removing [96](#), [97](#)
 response [159](#)
 retrieving [63](#)
 setting for hosts [110](#)
 Policy class [157](#)
 PolicyResponse class [159](#)
 Port class [159](#)
 PortResponse Class [162](#)
 ports
 business entities [39](#), [107](#)
 data about [159](#)
 retrieving [64–67](#), [162](#), [163](#), [165](#)
 PortsByFabric class [163](#)
 PortsByFabricResponse Class [163](#)
 PortsByZone class [164](#)
 PortsByZoneResponse Class [165](#)
 prerequisites
 Connect API [14](#)
 product documentation
 location [18](#)

Q

Qtree class [165](#)
 qtrees
 applications [34](#), [98](#), [104](#)
 class for [165](#)
 get [68](#)

R

removeHostFromApplication method [94](#)
 removeHostFromApplicationById method [95](#)
 removeHostPolicies method [96](#)
 removeInternalVolumeFromApplication method [97](#)
 removePolicies method [97](#)
 removeQtreeFromApplication method [98](#)
 removeReservationRequirements method [99](#)
 removeShareFromApplication method [99](#)
 removeVmFromApplication method [100](#)
 removeVolumeFromApplication method [101](#)
 RequestIterator class [167](#)
 requests
 capacity management [26](#), [27](#), [74](#), [75](#)
 iterator [167](#)
 reservation violations
 response [177](#)
 ReservationRequest class [169](#)
 ReservationRequestTarget class [173](#)
 ReservationRequirement class [174](#)
 reservations
 capacity management [73](#)
 path [62](#)
 removing requirements [99](#)

- request target [129, 173](#)
 - requests [169](#)
 - requirements [174](#)
 - updating requests for [111](#)
 - updating requirements for [111](#)
 - violations [76](#)
 - violations of [175, 176](#)
 - ReservationViolation class [175](#)
 - ReservationViolationsByRequest class [176](#)
 - ReservationViolationsByRequestResponse class [177](#)
 - retrieval methods
 - API version [32](#)
 - applications [31, 33, 36](#)
 - closed violations [41](#)
 - device groups by volume [45, 46](#)
 - DR paths by volume [47](#)
 - generic devices [48, 49](#)
 - getFabricsByPorts [48](#)
 - host policies [50](#)
 - hosts/servers [50, 51, 70](#)
 - hosts/servers by application [52](#)
 - internal volumes [53, 54, 115](#)
 - logical paths [60, 61](#)
 - logical ports [55–57](#)
 - logical ports by session context [42](#)
 - logical switches by physical switch [58](#)
 - lzones [93](#)
 - nodes [58](#)
 - pending path reservations [62](#)
 - physical ports by logical ports [62](#)
 - ports [64, 65](#)
 - ports by device node [66](#)
 - ports by node [67](#)
 - ports by type [66](#)
 - storage arrays [70, 78, 79](#)
 - storage pools [112–114](#)
 - switches [71, 79, 80](#)
 - Symmetrix device group by id [43](#)
 - Symmetrix device groups [44](#)
 - tape devices [81, 82](#)
 - tapes [72](#)
 - violations [84](#)
 - virtual machines [86](#)
 - volumes [73, 87, 90, 114](#)
 - opening [15, 94](#)
 - setAnnotationValues method [101](#)
 - setApplication method [102](#)
 - setApplicationOfInternalVolume method [103](#)
 - setApplicationOfQtree method [104](#)
 - setApplicationOfShare method [104](#)
 - setApplicationOfVolume method [105](#)
 - setApplicationWithNoBusinessEntity method [106](#)
 - setBusinessEntity method [107](#)
 - setBusinessEntityOfPort method [107](#)
 - setBusinessEntityOfStorage method [108](#)
 - setBusinessEntityOfSwitch method [41, 109](#)
 - SetHostsPolicy method [110](#)
 - Share class [179](#)
 - shares
 - applications [35, 99, 104](#)
 - data about [179](#)
 - get [77](#)
 - storage
 - business entities [38–40, 107, 108](#)
 - storage arrays
 - response [182](#)
 - retrieving [70, 78, 79, 180](#)
 - storage pools
 - response [188](#)
 - retrieving [112–114, 183](#)
 - StorageArray class [180](#)
 - StorageArrayResponse class [182](#)
 - StoragePool class [183](#)
 - StoragePoolResponse class [188](#)
 - suggestions
 - how to send feedback about documentation [223](#)
 - switches
 - business entities [41, 109](#)
 - information about [189](#)
 - response [190](#)
 - retrieving [71, 79, 80](#)
 - Switches class [189](#)
 - SwitchResponse class [190](#)
 - synchronization
 - volumes [214, 216](#)
 - SynchronizationResponse class [191](#)
- ## T
- TapeResponse class [193](#)
 - tapes
 - data about [192](#)
 - response [193](#)
 - retrieving [72, 81, 82](#)
 - Task class [194](#)
 - TaskResponse class [196](#)
 - tasks
 - data about [194](#)
 - future violations on [197](#)
 - TaskViolationResponse class [197](#)
 - Twitter
 - how to receive automatic notification of documentation changes [223](#)
- ## S
- sample code
 - Java [218](#)
 - sample scripts
 - Perl [219](#)
 - samples [218](#)
 - SAN devices
 - discovery by WWN [30](#)
 - retrieving by IP/DNS [30](#)
 - SANscreenAPIFactory class [178](#)
 - scripts
 - Perl [219](#)
 - security
 - Connect API authentication [14](#)
 - default passwords [16](#)
 - sessions
 - closing [16, 28](#)

U

updateReservationRequests method [111](#)
 updateReservationRequirements method [111](#)

V

Violation class [198](#)
 ViolationResponse class [200](#)
 violations
 by requests [84](#)
 by reservations [85](#)
 closed [41](#)
 data about [198](#)
 from reservations [175, 176](#)
 future per task [197](#)
 path outages [29](#)
 reservations [76](#)
 response [177, 200](#)
 retrieving [83](#)
 virtual machines
 data about [201](#)
 response [204](#)
 retrieving [86](#)
 VirtualMachine class [201](#)
 VirtualMachineResponse class [204](#)
 VMs
 applications [37](#)
 Volume class [205](#)
 volume response [212](#)
 VolumeMap class [208](#)
 VolumeMapResponse class [210](#)
 VolumeMask class [211](#)
 VolumeMaskResponse class

 volume masks [212](#)
 VolumeResponse class [213](#)
 volumes
 applications [36, 101, 105](#)
 data about [205](#)
 DR paths [47](#)
 getting masks [89](#)
 mapping between storage volume and port [208](#)
 mapping response [210](#)
 maps [88](#)
 mask response [212](#)
 masks [211](#)
 response [213](#)
 retrieving [73, 87, 90, 114](#)
 synchronization [214](#)
 synchronization state response [216](#)
 synchronization states for [91, 92](#)
 VolumesSynchronizationState class [214](#)
 VolumesSynchronizationStateResponse class [216](#)

W

web services WSDL file [217](#)
 web services WSDL specification [217](#)
 WSDL web services file [217](#)
 WSDL web services specification [217](#)

Z

zones
 by Ports [93](#)