Opswise Controller 5.2.0

Remote Interfaces

Remote Interfaces

Command Line Interface (CLI)

RESTful Web Service

Overview

Command Parameters

Return Codes

Using the Command Line Interface in z/OS

Categories of CLI Commands

Agents and Connectors Commands

Import and Export Commands

Passwords Commands

Resources Commands

Tasks Commands

Triggers Commands

Variables Commands

Overview

Categories of RESTful V

Agents Operations

Passwords Operations

Resources Operations

Tasks Operations

Triggers Operations

Variables Operations

The information on these pages also is located in the Opswise Controller 5.2.0 Remote Interfaces.pdf.

Command Line Interface (CLI)

Overview

The Opswise Command Line Interface (CLI) is implemented as a set of commands that perform specific actions in an Opswise Controller. The results of the actions are written to the CLI commands standard output.

For all platforms, the CLI is included in the Opswise Universal Agent 5.2.0 package.

UNIX	The CLI commands are installed in the following directory: • `/CLI_HOME/bin` Replace `CLI_HOME` with the directory in which the Agent components are installed.

Windows	The CLI commands are installed in the following directory:
	
	    • `DRIVE:\CLI_HOME\bin`
	
	Replace:
	
	    • `DRIVE` with the drive they are installed on. Add the appropriate directory for your O/S to your PATH environment variable in order to execute the commands without specifying their full path name.
	    • `CLI_HOME` with the directory in which the Agent components are installed.

z/OS	The CLI is implemented with a single program, **UAGCMDZ**. **UAGCMDZ** with alias **OPSCMDZ** is installed in the SUNVLOAD library as part of the Opswise Universal Agent for z/OS Installation package.

Modifying Variables

Currently, the CLI only supports modifying global variables, trigger variables, and task definition variables (see User-Defined Variables).

Modifying a task definition variable does not impact any existing task instances of that task. When the next task instance is created from the updated task definition, the new task instance will use the updated variable.

To modify a variable in a workflow (or task), you can create a Set Variable action.

Command Parameters

Each CLI command includes two types of parameters:

1. Global parameters are the same for every command. There are required and optional Global parameters.
2. Command-specific parameters are either unique to a single command or used by multiple, but not all, commands. There are required and optional command-specific parameters.

Global Parameters

The CLI provides the following Global parameters for use with any command:

- Configuration File
- Encrypted File
- Encrypted File Key
- Help
- Message Hub
- Network Provider
- OMS NFT
- OMS Servers
- Password
- Timeout
- SSL
- Transporter
- User ID
- Version

Commands can execute on any system that has TCP/IP connectivity to the Opswise Message Service (OMS) or Opswise Automation Center 5.1.0 Transporter. You must use the required Global parameters to connect to OMS (Network Provider and OMS Servers) or the Transporter and Message Hub (Message Hub and Transporter), and to log on to a Controller (User ID and Password).

Three methods are available for specifying global parameters:

1. Configuration file (**cmdtools.props**)
2. Command line
3. Environment variables

The following information is provided for each Global parameter:

- Description

- Syntax for specifying the parameter for each method (configuration file, command line, environment variable).
- Requirement to use the parameter.
- Default value (if any).
- Example

Configuration File

Description	Configuration file containing one or more Global parameters. You can specify a configuration file and/or individual global parameters on the command line.
Configuration File Syntax	n/a
Command Line Syntax	-c
Environment Variable Syntax	n/a
Required	No
Default	(none)
Example	cmdtools.props

Encrypted File

Description	Complete path to a file encrypted with the Universal Encrypt utility that contains encrypted user credentials. This file can be used to store encrypted Controller credentials that would otherwise be specified using the User ID or Password parameters. One or both of these parameters can be encrypted and stored in this file. Any other command line parameters stored in the encrypted file are ignored. Any values specified by User ID or Password on the command line and/or in the configuration file will override values stored in this file. ⚠ **Note** Universal Encrypt is included in every Agent installation package.
Configuration File Syntax	n/a
Command Line Syntax	-x
Environment Variable Syntax	n/a
Required	No
Default	(none)
Example	

Encrypted File Key

Description	Optional key that was passed to the Universal Encrypt utility when the encrypted file specified by -x was created.
Configuration File Syntax	n/a
Command Line Syntax	-k
Environment Variable Syntax	n/a
Required	No
Default	(none)

Example	

Help

Description	Provides help information for any command.
Configuration File Syntax	n/a
Command Line Syntax	-h | -?
Environment Variable Syntax	n/a
Required	No
Default	(none)
Example	-h

> ⚠ **Note**
> This parameter is exclusive; do not use it with any other parameter. If you do, only the Help information will be returned.

Message Hub

Description	Queue name for the Message Hub. You can access this information from the User interface: • From the navigation pane, select **Automation Center Resources > Connectors** to display the Connectors List screen. • Locate the **Queue** for the Message Hub.
Configuration File Syntax	network.core
Command Line Syntax	-n
Environment Variable Syntax	HUBNAME
Required	Yes (if Network Provider is **TRANSPORT**)
Default	HUB01
Example	HUB01

> ⚠ **Note:**
> If the Message Hub parameter is not contained in the configuration file or specified on the command line, the CLI will attempt to get its value from the HUBNAME environment variable.

Network Provider

Description	Network communication provider used by the command to communicate with Opswise Controller. There are two network providers: • **TRANSPORT** specifies that the legacy (prior to Opswise Controller 5.2.0) network provider is used: the Transporter. • **OMS** specifies that Opswise Message Service (OMS) is used. It is recommended that the OMS network provider is used whenever possible. If Network Provider is set to **TRANSPORT**, the Message Hub and Transporter parameters are used to configure the Transport network provider. If Network Provider is set to **OMS**, the OMS Servers parameter is used to configure the OMS network provider.

Configuration File Syntax	network.provider
Command Line Syntax	-r
Environment Variable Syntax	NETPROVIDER
Required	Yes (if you want to specify OMS Servers)
Default	TRANSPORT
Example	

OMS NFT

Description	Specification (yes or no) for whether or not to use NFT (network fault tolerance) for OMS. If NFT is not used: • A command execution will attempt to connect to each specified [OMS server|#OMS Servers]. If no connection can be made, the command will fail with a non-zero exit code (2). • If a network error occurs after a connection to an OMS server is made, the command will fail with a non-zero exit code (2).
Configuration File Syntax	network.omsnft
Command Line Syntax	-f
Environment Variable Syntax	OMSNFT
Required	No
Default	yes
Example	

OMS Servers

Description	Specifies one or more OMS server addresses used for network communication. The syntax of the value is PORT@HOST[,PORT@HOST...], where: • PORT is the TCP port on which the OMS server is listening. The OMS default port is 7878. • HOST is the host name or IP address of the OMS server. If multiple OMS servers are specified, they must be comma-separated.
Configuration File Syntax	network.omsservers
Command Line Syntax	-m
Environment Variable Syntax	OMSSERVERS
Required	Yes (if Network Provider is **OMS**)
Default	
Example	oms_servers 7878@dallas01.acme.com,7878@dallas02.acme.com

Password

Description	Password that the CLI will use to log into the Controller. Must be a valid password for this user.
Configuration File Syntax	security.password
Command Line Syntax	-p
Environment Variable Syntax	n/a
Required	Yes

Default	(none)
Example	o

SSL

Description	Specification (by its inclusion on the command line) for whether or not to enable SSL.
Configuration File Syntax	n/a
Command Line Syntax	-s
Environment Variable Syntax	n/a
Required	No
Default	(none)
Example	-s

Timeout

Description	Time (in seconds) that the CLI will wait before it reports an error.
Configuration File Syntax	config.timeout
Command Line Syntax	-o
Environment Variable Syntax	n/a
Required	No
Default	60
Example	10

Transporter

Description	Port and machine name (host name or IP address) where the Transporter is running. You can access this information from the Opswise Controller user interface: 1. From the navigation pane, select **Automation Center Resources > Connectors** to display the Connectors list screen. 2. Click the **Connector Name** for the Transporter that was specified in the network configuration to display its details, including the Port, Host Name, and IP Address. Provide this information in either of the following formats: • `Port@Host Name` • `Port@IP Address`
Configuration File Syntax	network.transports
Command Line Syntax	-t
Environment Variable Syntax	TSPNAME
Required	Yes (if Network Provider is **TRANSPORT**)
Default	(none)
Example	4803@localhost

⚠ **Note:**
If the Transporter parameter is not contained in the configuration file or specified on the command line, the CLI will attempt to get its value from the `TSPNAME` environment variable.

User ID

Description	User ID that the CLI will use to log into the Controller. Must be a valid user with the appropriate permissions defined in the Controller database using the Controller Security module.
Configuration File Syntax	security.userid
Command Line Syntax	-u
Environment Variable Syntax	n/a
Required	Yes
Default	(none)
Example	ops.admin

Version

Description	Displays version information for the CLI.
Configuration File Syntax	n/a
Command Line Syntax	-v
Environment Variable Syntax	n/a
Required	No
Default	(none)
Example	-v

⚠ **Note**
If this parameter is specified, all other command parameters (except Help) are ignored.

Using the Configuration File

The following is the command line syntax for using a configuration file to pass Global parameters into your command:

```
command -c filename command-specific parameter(s)
```

For example:

```
ops-task-status -c c.cfg task-name=mantask
```

> ⚠️ **Note**
> A sample CLI configuration file (`cmdtools.props`), created during the installation of the CLI (an optionally installed component of Opswise Universal Agent), is provided for your use. However, you can create a configuration file with any name; it must exist in the directory from where you are issuing the commands.

Using the Command Line

The following is the command line syntax for specifying individual Global parameters on the command line to pass them into your command:

```
command -r network provider -m OMS Server -u user id -p password [-o timeout] command-specific
parameter(s)
```

For example:

```
ops-task-status -r OMS -m 7878@dallas01.acme.com -u ops.admin -p o -o 10 task-name=mantask
```

Using Environment Variables

The following is the command line syntax for specifying Environmental Variables (and individual Global parameters) on the command line to pass them into your command:

```
command NETPROVIDER=network provider OMSSERVERS=OMS Server -u user id -p password [-o timeout]
command-specific parameter(s)
```

For example:

```
ops-task-status NETPROVIDER=OMS OMSSERVERS=7878@dallas01.acme.com -u ops.admin -p o -o 10
task-name=mantask
```

User Restriction

You can be restricted from logging in to the CLI either of two ways:

1. The system level default for CLI access, specified by the System Default Command Line Access Opswise Controller system property, has been set to **No**, and the **Command Line access** field on the User Definition screen for your user account is set to **-- System Default --**."
2. The **Command Line access** field is set to **No**, which overrides the **System Default Command Line Access** value.

If either restriction is in place, the following error message will display when you issue a CLI command:

```
User <your user name> not permitted to use the command line interface. Please check with your
administrator.
```

To remove the restriction, the system administrator must either:

- Set the System Default Command Line Access property to **Yes** and set the **Command Line access** field on the User Definition screen

for your user account to **-- System Default --**.
- Set the **Command Line access** field on the User Definition screen for your user account to **Yes**.

Password Expiration

While a password is expired, CLI access will be prohibited until the password has been changed.

> ⚠ **Note**
> Password expiration is not applicable to LDAP authenticated users.

Return Codes

The following table identifies all return codes that can be returned in response to a command:

Return Code	Description
0	Completed successfully.
1	Configuration error (incorrect command line or configuration file options).
2	Error processing the request.

Using the Command Line Interface in z/OS

All Opswise Controller commands are supported in the z/OS environment. They are managed and executed by the program UAGCMDZ with alias OPSCMDZ, which may execute as either a command processor or a standard z/OS batch job. OPSCMDZ is installed with the z/OS agent in library SUNVLOAD.

z/OS-Specific Syntax Requirements

The commands and syntax requirements for the z/OS environment are very similar to the other supported platforms with a few exceptions described below.

Command Line Options

When **OPSCMDZ** is executed as a batch program, command line parameters are specified with the step PARM keyword.

For example:

```
//STEP01   EXEC PGM=OPSCMDZ,
//  PARM='ops-agent-status -c dd:CMDOPTS agent-type=windows'
```

Configuration File Syntax

If you use the **-c** command line parameter for specifying a configuration file on the command line, you must use one of the following formats for the configuration file name:

Format	Syntax	Description
ddname	`-c dd:ddname`	*ddname* references a DD statement that exists in your batch JCL.
Data Set Name	`-c "//'dsname'"`	*dsname* is a fully-qualified data set name which may be a partitioned data set with a member name.

Line Numbers in Configuration File

Do not place line numbers in columns 73-80. The entire 80-byte record is read and processed as input.

Methods for Issuing Opswise Controller Commands in z/OS

Three methods are available for running OPSCMDZ and executing Opswise Controller commands:

1. Using a batch job.
2. Under a TSO session.
3. Using a batch TSO Terminal Monitor Program (TMP).

In each example where data sets are specified, we use:

UNV.SUNVLOAD	Library containing the CLI programs UAGCMDZ and OPSCMDZ alias.
USER.PARM	Data set containing command options.
USER.REXX	Data set containing user REXX EXECs.

Running Opswise Controller Commands in a Batch Job

The CLI program OPSCMDZ executes a single Opswise Controller command as a JCL batch job step. If multiple commands are to be executed, run each one as an individual job step.

The following example illustrates running OPSCMDZ as a batch job to request the status of all Windows Agents.

```
//jobname  JOB (acctg-info),'your name',MSGCLASS=X,MSGLEVEL=(1,1),
//             CLASS=A,NOTIFY=&SYSUID
//*
//STEP01   EXEC PGM=OPSCMDZ,
// PARM='ops-agent-status -c dd:CMDOPTS agent-type=windows'
//STEPLIB  DD DISP=SHR,DSN=UNV.SUNVLOAD
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CMDOPTS  DD DSN=USER.PARM(CMDOPTS),DISP=SHR
//CEEDUMP  DD SYSOUT=*
```

Here is a breakdown of this example:

- The step PARM value specifies the CLI command and it options.
- The job step can include only one Opswise Controller command.
- The STEPLIB points to the z/OS Agent load library.
- The command output will be in SYSPRINT.
- CMDOPTS is a DDNAME that references the location of the command parameters.

Issuing Commands under TSO

You can also issue Opswise commands under a TSO session.

The sample command string below shows an Opswise Controller command issued from an ISPF Command Shell prompt:

```
Enter TSO or Workstation commands below:
===> opscmdz ops-agent-status -c "//'USER.PARM(CMDOPTS)'" agent-type=windows
```

The above command will return data similar to the sample below:

```
Agentname                    AgentType     Status
agent-sys00101               Windows       Active
agent-sys00201               Windows       Offline
opscmd-complete
```

Issuing Commands as a Batch TSO

The following two examples show how to issue an Opswise Controller command in a batch TSO.

Example One

```
//CMDZBAT  JOB (acctg-info),'your name',MSGCLASS=X,MSGLEVEL=(1,1),
//            CLASS=A,NOTIFY=&SYSUID,TIME=5
//*
//STEP01   EXEC PGM=IKJEFT01,DYNAMNBR=200,REGION=40M
//STEPLIB  DD DISP=SHR,DSN=UNV.SUNVLOAD
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//CMDOPTS  DD DSN=USER.PARM(CMDOPTS),DISP=SHR
//SYSTSIN DD *
opscmdz ops-agent-status -c dd:CMDOPTS agent-type=windows
/*
```

Output will be in SYSTSPRT.

Example Two

```
//REXXAGNT JOB (acctg-info),'your name',MSGCLASS=X,MSGLEVEL=(1,1),
//             CLASS=A,NOTIFY=&SYSUID,TIME=5
//*
//STEP01   EXEC PGM=IKJEFT01,DYNAMNBR=200,REGION=40M
//STEPLIB  DD DISP=SHR,DSN=UNV.SUNVLOAD
//SYSEXEC  DD DSN=USER.REXX,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//CMDOPTS  DD DSN=USER.PARM(CMDOPTS),DISP=SHR
//SYSTSIN DD *
  %OUTTRAP1
/*
```

The following REXX EXEC must be located in USER.REXX(OUTTRAP1):

```
/*************************** REXX *******************************/
/* Using OUTTRAP to                                            */
/*        (1) Obtain z/OS agent status                         */
/*        (2) Test status from the command response            */
/*        (3) Launch a task if the agent status is Active       */
/***************************************************************/
x = OUTTRAP('OPS.')
opscmdz "ops-agent-status -c dd:CMDOPTS agent-type=z/OS"
SAY 'RC is:' RC
SAY OPS.0 'records were read.'
launch = 'NO'
DO i = 1 to OPS.0 WHILE launch = 'NO'
  IF SUBSTR(OPS.i,61,6) = 'Active' THEN
    DO
     launch = 'YES'
     opscmdz "ops-task-launch -c dd:CMDOPTS task-name=DUMPT"
    END
END
DO i = 1 to OPS.0
   SAY OPS.i
END
y = OUTTRAP('OFF')
```

Commands

The following table provides the command line syntax and description for all CLI commands. A description of the command-specific parameters available for each command is listed below its command line syntax.

The commands are listed alphabetically within the following categories:

- Agents and Connectors
- Import and Export
- Passwords
- Resources
- Tasks
- Triggers
- Variables

Agents and Connectors

List Agents Status

Description	Lists the status of one or more Agents.
Syntax	**ops-agent-status** [agent-name=*name*] agent-type=*type* [options=**v**]
Parameters	- agent-name= Optional; Name or partial name of one or more Agents. Wildcards are supported. - agent-type= Required; One of the following (not case sensitive): **windows**, **unix**, **linux**, **linux/unix**, **zos**, **z/os**. - options= Optional; **v** (Return verbose results.)
Example	```ops-agent-status -c config.txt agent-type=windows```

List Connectors Status

Description	Lists the status of one or more Connectors.
Syntax	**ops-connector-status** [connector-name=*name*] [options=**v**]
Parameters	- connector-name= Optional; Name or partial name of one or more Connectors. Wildcards are supported. - options= Optional; **v** (Return verbose results.)
Example	```ops-connector-status -c config.txt connector-name=*HUB01```

Resume Agent

Description	Resumes the ability of the identified Agent to run tasks.
Syntax	**ops-resume-agent** {agent-name=*name* | agent-id=*id*}
Parameters	- agent-name= Required if agent-id= is not specified; Agent name of the Agent whose ability to run tasks is to be resumed (agent-name= and agent-id= are mutually exclusive). - agent-id= Required if agent-name= is not specified; Agent ID of the Agent whose ability to run tasks is to be resumed (agent-id= and agent-name= are mutually exclusive).

Resume Agent Cluster

Description	Resumes the ability of the identified Agent Cluster to run tasks.
Syntax	**ops-resume-agent-cluster** agent-cluster-name=*name*

Parameter	• `agent-cluster-name=` Required; Agent cluster name of the Agent Cluster whose ability to run tasks is to be resumed.

Resume Agent Cluster Membership

Description	Resumes the membership of the identified Agent in the identified Agent Cluster.
Syntax	`ops-resume-agent-cluster-membership` {agent-name=*name* | agent-id=*ID*} agent-cluster-name=*name*
Parameters	• `agent-name=` Required if `agent-id=` is not specified; Agent name of the Agent whose membership in the Agent Cluster is to be resumed (`agent-name=` and `agent-id=` are mutually exclusive). • `agent-id=` Required if `agent-name=` is not specified; Agent ID of the Agent whose membership in the Agent Cluster is to be resumed (`agent-id=` and `agent-name=` are mutually exclusive). • `agent-cluster-name=` Required; Agent cluster name of the Agent Cluster in whose membership the identified Agent is to be resumed.

Set Agent Cluster Task Execution Limit

Description	Specifies whether or not an Agent Cluster has a task execution limit and, if so, sets the task execution limit.
Syntax	`ops-set-agent-cluster-task-execution-limit` agent-cluster-name=*name* limit-type=*type* limit-amount=*number*
Parameters	• `agent-cluster-name=` Required; Name of the Agent Cluster. • `limit-type=` Required; Specification for whether or not there is a limit to the number of tasks that can be run at the same time by the Agent Cluster. Allowable values (not case sensitive) are: **limited**, **unlimited**. • `limit-amount=` Required if `limit-type=`**limited**; Maximum number of tasks that can be run at the same time by the Agent Cluster.

Set Agent Task Execution Limit

Description	Specifies whether or not an Agent has a task execution limit and, if so, sets the task execution limit.
Syntax	`ops-set-agent-task-execution-limit` [agent-name=*name*] [agent-id=*ID*] limit-type=*type* limit-amount=*number*
Parameters	• `agent-name=` Required if `agent-id=` is not specified; Name of the Agent (`agent-name=` and `agent-id=` are mutually exclusive). • `agent-id=` Required if `agent-name=` is not specified; ID of the Agent (`agent-id=` and `agent-name=` are mutually exclusive). • `limit-type=` Required; Specification for whether or not there is a limit to the number of tasks that can be run concurrently by the Agent. Allowable values (not case sensitive) are **limited**, **unlimited**. • `limit-amount=` Required if limit-type=limited; Maximum number of tasks that can be run at the same time by the Agent.

Suspend Agent

Description	Suspends the ability of the identified Agent to run tasks.
Syntax	`ops-suspend-agent` [agent-name=*name*] [agent-id=*ID*]

Parameters	• `agent-name=` Required if `agent-id=` is not specified; Agent name of the Agent whose ability to run tasks is to be suspended (`agent-name=` and `agent-id=` are mutually exclusive). • `agent-id=` Required if `agent-name=` is not specified; Agent ID of the Agent whose ability to run tasks is to be suspended (`agent-id=` and `agent-name=` are mutually exclusive).

Suspend Agent Cluster

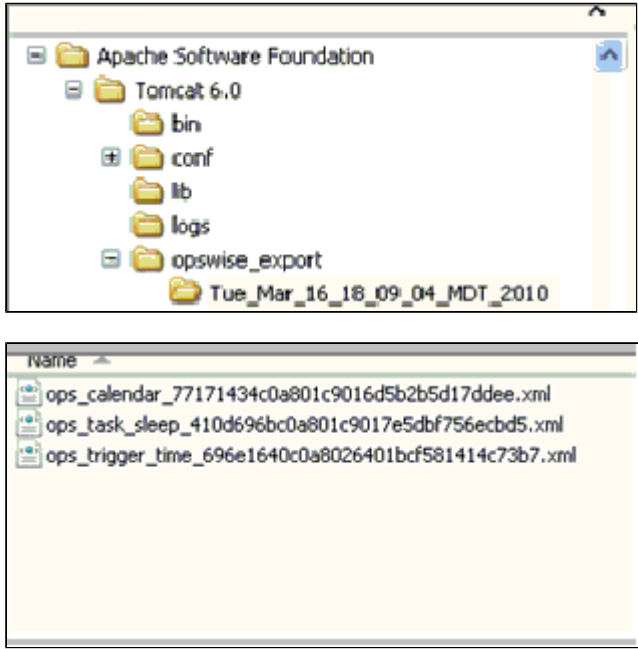| Description	Suspends the ability of the identified Agent Cluster to run tasks.
Syntax	`ops-suspend-agent-cluster` agent-cluster-name=*name*
Parameter	• `agent-cluster-name=` Required; Agent cluster name of the Agent Cluster whose ability to run tasks is to be suspended.

Suspend Agent Cluster Membership

Description	Suspends the membership of the identified Agent in the identified Agent Cluster.
Syntax	`ops-suspend-agent-cluster-membership` [agent-name=*name*] [agent-id=*ID*] agent-cluster-name=*name*
Parameters	• `agent-name=` Required if `agent-id=` is not specified; Agent name of the Agent whose membership in the Agent Cluster is to be suspended (`agent-name=` and `agent-id=` are mutually exclusive). • `agent-id=` Required if `agent-name=` is not specified; Agent ID of the Agent whose membership in the Agent Cluster is to be suspended (`agent-id=` and `agent-name=` are mutually exclusive). • `agent-cluster-name=` Required; Agent cluster name of the Agent Cluster in whose membership the identified Agent is to be suspended.

Import and Export

Export Bulk

Description	Performs a bulk export of all records in your Controller database. It creates a separate XML file for each record type in the following directory: `\$tomcat dir$\webapps\opswise\WEB-INF\plugins\com.opswise\backup\unload`
Syntax	`ops-export-bulk`
Example	``` ops-export-bulk -c config.txt ```

Export Trigger

Description	Performs an export of the specified trigger(s) and all the associated record(s). For example, if you export one trigger, the CLI also exports the calendar used in the trigger and all tasks launched by the trigger. The CLI creates a separate XML file for each record type and generates the output directory and file names using a date and timestamp, plus an internal record identifier generated by the Controller. All exported XML definition files will be saved to a timestamped directory under the path defined by the Export Path Opswise Controller system property. The following illustrates a sample output directory and exported data:
Syntax	**ops-export-trigger** trigger-name=*name* [trigger-type=*type*]
Parameters	`trigger-name=` Required; Name of one or more triggers. Wildcards are supported. `trigger-type=` Optional; If you used a wildcard in **trigger-name**, you can use **trigger-type** to narrow down the selection to a specific type of trigger. Allowable values are (not case sensitive): **cron**, **time**, **file_trigger**, **temporary**, **task_monitor**, **manual**, **application_monitor**.
Example	This example exports all triggers whose name begins with **SF**: `ops-export-trigger -c c.cfg trigger-name=SF*`

Import Bulk

Description	Imports into your Controller database the contents of the following directory: `\$tomcat dir$\webapps\opswise\WEB-INF\plugins\com.opswise\backup\unload` The data being imported must have been exported using the ops-export-bulk command.
Example	`ops-import-bulk -c config.txt`

Import Trigger

Description	Imports into your Opswise database the trigger records in the path specified in the `import-file=` parameter.

Syntax	**ops-import-trigger** import-file=*file*
Parameter	• import-file= Required; Path containing one or more XML files that were exported via ops-export-trigger or via the Opswise Controller user interface using the Export -> XML feature. ⚠️ **Note** This is the path on the Controller server where the XML files are stored, not a local path to where the command is run.

Passwords

Change User Password

Description	Changes the password for the specified Opswise User account. ⚠️ **Note** The authenticated user must have the ops_admin user role.
Syntax	**ops-change-user-password** user-id=*userid* new-password=*password*
Parameters	• user-id= Required; Opswise User ID whose password is to be changed. • new-password= Required; The new password that is to be assigned to the specified user.

Change Credentials Password

Description	Changes the runtime password for the specified Opswise credentials resource. ⚠️ **Note** The authenticated user must have the **Update** Credential permission.
Syntax	**ops-change-credentials-password** credentials-name=*name* new-runtime-password=*password*
Parameters	• credentials-name= Required; Name of the Opswise credentials resource that is to have its password changed. • new-runtime-password= Required; The new runtime password that is to be assigned to the specified credentials resource.

Resources

Update Resource Limit

Description	Updates the resource limit for a resource.
Syntax	**ops-update-resource-limit** resource-name=*name* resource-limit=*limit*

Parameters	
	• resource-name= Required; Name of resource to update the limit on. • resource-limit= Required; Resource limit (any integer >= 0).

Tasks

Cancel Task

Description	Cancels the specified task instance. See Cancelling a Task Instance for a description of behavior and restrictions.
Syntax	**ops-task-cancel** task-instance=*ID*
Parameter	• task-instance= Required; An internal Opswise Controller identifier.

Display All Tasks

Description	Displays all specified tasks.
Syntax	**ops-task-list** task-name=*name* task-type=*type*
Parameters	• task-name= Optional; name or partial name of one or more tasks. Wildcards are supported. • task-type= Required; One of the following (not case sensitive): **sleep**, **windows**, **Unix**, **zos**, **file_monitor**, **manual**, **email**, **ftp**, **sql**, **ftp_file_monitor**, **task_monitor**, **stored_procedure**, **workflow**, **sap**, **system_monitor**, **indesca**, **application_control**.

Display Task Instances Status

Description	Displays the status of all task instance(s) associated with the specified task.
Syntax	**ops-task-status** task-name=*name* [task-type=*type*] [task-status=*status*] [options=**v**]
Parameters	• task-name= Required; name or partial name of one or more tasks. Wildcards are supported. • task-type= Optional; One of the following (not case sensitive): **sleep**, **windows**, **unix**, **zos**, **file_monitor**, **manual**, **email**, **ftp**, **sql**, **ftp_file_monitor**, **task_monitor**, **stored_procedure**, **workflow**, **sap**, **system_monitor**, **indesca**, **application_control**. • task-status= Optional; not case sensitive. Any valid task status. • options= Optional; **v** (Return verbose results.)
| Example | ```
ops-task-status -c c.cfg task-name=mantask
``` |

Force Finish a Task

Description	Force finishes the specified task instance. See Force Finishing a Task Instance for a description of behavior and restrictions.
Syntax	**ops-task-forcefinish** task-instance=*ID*
Parameter	• task-instance= Required; An internal Opswise Controller identifier.

Launch a Task

Description	Launches the specified task(s). Wildcards are supported.
Syntax	**ops-task-launch** task-type=*type* task-name=*_name* [task-variables=*_variables*]
Parameters	• task-type= Required; One of the following (not case sensitive): **sleep**, **windows**, **unix**, **zos**, **file_monitor**, **manual**, **email**, **ftp**, **sql**, **ftp_file_monitor**, **task_monitor**, **stored_procedure**, **workflow**, **sap**, **system_monitor**, **indesca**, **application_control**. • task-name= Required; Valid task name. • task-variables= Optional; Any variables specified in the task that need a value to run properly.
| Example | (All **variable=value** pairs must be specified within one set of braces.)<br><br>```
task-variables={variable1=first value
variable2=second value}
``` |

Put Task on Hold

Description	Puts the specified task instance on hold. See Putting a Task Instance on Hold for a description of behavior and restrictions.
Syntax	**ops-task-hold** task-instance=*ID*
Parameter	• task-instance= Required; An internal Opswise Controller identifier.
| Example | ```
ops-task-hold -c c.cfg
task-instance=90079020d861e5e40128cbb3cdbe1cf3
``` |

Release a Task from Hold

Description	Releases the specified task instance from hold. See Releasing a Task Instance from Hold for a description of behavior and restrictions.
Syntax	**ops-task-release** task-instance=*ID*
Parameter	• task-instance= Required; An internal Opswise Controller identifier.

Example	
| | ```
ops-task-release -c c.cfg
task-instance=90079020d861e5e40128cbb3cdbe1cf3
``` |

Rerun a Task

Description	Reruns the specified Task instance. See Re-running a Task Instance for a description of behavior and restrictions.
Syntax	**ops-task-rerun** task-instance=*ID*
Parameter	• task-instance= Required; An internal Opswise Controller identifier.

Set Manual Task to Started

Description	For a Manual task, changes its status from Action Required to Started. This allows you to acknowledge the Manual task and indicate that the manual procedures have been started.
Syntax	**ops-manual-setstarted** task-instance=*ID*
Parameter	• task-instance= Required; An internal Opswise Controller identifier.
| Example | ```
ops-manual-setstarted -c c.cfg
task-instance=8fda93dcd861e5e4005bf48e2cad6206
``` |

Set Manual Task to Success

Description	For a Manual task, sets its status to Success.
Syntax	**ops-manual-setcompleted** task-instance=*ID*
Parameter	• task-instance= Required; An internal Opswise Controller identifier.
| Example | ```
ops-manual-setcompleted -c c.cfg
task-instance=8fda93dcd861e5e4005bf48e2cad6206
``` |

Set Task Prioirity

Description	For Agent-based tasks (Windows, Linux/Unix, or z/OS) in a status of Started, changes the priority on the specified task instance. See Changing the Priority of a Task Instance for a description of behavior and restrictions.
Syntax	**ops-task-setpriority** task-instance=*ID* priority=*priority*

Parameters	`task-instance=` Required; An internal Opswise Controller identifier. `priority=` Required; One of the following (not case sensitive): **high**, **medium**, **low**.

Skip a Task

Description	Skips the specified task instance. See Skipping a Task Instance for a description of behavior and restrictions.
Syntax	**ops-task-skip** `task-instance=ID`
Parameter	`task-instance=` Required; An internal Opswise Controller identifier.
Example	```ops-task-skip -c c.cfg task-instance=90079026d861e5e400bba81913a4fdd0```

Triggers

Disable a Trigger

Description	Disables the specified trigger(s).
Syntax	**ops-trigger-disable** `trigger-name=name` [`trigger-type=type`]
Parameters	`trigger-name=` Name or partial name of one or more triggers. Wildcards are supported. `trigger-type=` Optional; If you used a wildcard in `trigger-name=`, you can use `trigger-type=` to narrow down the selection to a specific type of trigger. One of the following (not case-sensitive): **cron**, **time**, **file_trigger**, **temporary**, **task_monitor**, **manual**, **application_monitor**.

Enable a Trigger

Description	Enables the specified trigger(s).
Syntax	**ops-trigger-enable** `trigger-name=name` [`trigger-type=type`]
Parameters	`trigger-name=` Name or partial name of one or more triggers. Wildcards are supported. `trigger-type=` Optional; If you used a wildcard in `trigger-name=`, you can use `trigger-type=` to narrow down the selection to a specific type of trigger. One of the following (not case-sensitive): **cron**, **time**, **file_trigger**, **temporary**, **task_monitor**, **manual**, **application_monitor**.

Launch Trigger Tasks Now

Description	Marks all conditions as satisfied in the specified trigger(s) and launches its associated tasks.
Syntax	**ops-trigger-now** `trigger-name=name` [`trigger-type=type`] [`trigger-variables=variables`]

Parameters	`trigger-name=` Required; Name or partial name of one or more triggers. Wildcards supported. `trigger-type=` Optional; If you used a wildcard in `trigger-name=`, you can use `trigger-type=` to narrow down the selection to a specific type of trigger. One of the following (not case-sensitive): **cron**, **time**, **file_trigger**, **temporary**, **task_monitor**, **manual**, **application_monitor**. `trigger-variables=` Optional; Any variables specified in the task(s) being triggered that need a value to run properly.
Example	All **variable=value** pairs must be specified within one set of braces: ``` trigger-variables={variable1=first value variable2=second value} ```

List Triggers Status

Description	Lists the status of the specified trigger(s). Possible statuses are **Enabled** or **Disabled**.
Syntax	**ops-trigger-status** trigger-name=*name* [trigger-type=*type*] [trigger-status=*status*] [options=**v**]
Parameters	`trigger-name=` Name or partial name of one or more triggers. Wildcards are supported. `trigger-type=` Optional; One of the following (not case-sensitive): **cron**, **time**, **file_trigger**, **temporary**, **task_monitor**, **manual**, **application_monitor**. `trigger-status=` Optional; One of the following (not case sensitive): **enabled**, **disabled**. `options=` Optional; **v** (Return verbose results.)

Variables

List Variables

Description	Lists the specified variable(s).
Syntax	**ops-variable-list** variable-name=*name* variable-scope=*scope* [task-name=*name*] [trigger-name=*name*] [options=**v**]
Parameters	`variable-name=` Optional; Name or partial name of one or more variables. Wildcards are supported. `variable-scope=` Required; One of the following (not case sensitive}: *global* (default), *local* (task & trigger), *task*, *trigger*. Defines the type of variable being listed. A global variable is created independently by selecting **Variables** from the Opswise Controller user interface navigation pane. A trigger variable is attached to a trigger; a task variable is attached to a task. Local can be either a task or trigger variable. `task-name=` Required if `variable-scope=`**task**; A single task name. `trigger-name=` Required if `variable-scope=`**trigger**; A single trigger name. `options=` Optional; **v** (Return verbose results.)
Example	``` ops-variable-list -c c.cfg variable-name=abc variable-scope=global ```

Set Variables

Description	Sets the specified variable.
Syntax	**ops-variable-set** variable-name=*name* variable-scope=*scope* variable-value=*string* [variable-description=*description*] [task-name=*name*] [trigger-name=*name*] create=*option*
Parameters	- variable-name= Required; Name of a specific variable. The name must begin with an alphabetic character and can consist of: alphabetic characters (a-z, A-Z), numbers (0-9), _ (underscore). White spaces are not permitted; names are not case-sensitive. - variable-scope= Required; One of the following (not case sensitive}: **global** (default), **task**, **trigger**. In cases where there may be more than one occurrence of the specified variable, variable-scope= defines which one should be set. A global variable is created independently by selecting **Variables** from the Opswise Controller user interface navigation pane. A trigger variable is attached to a trigger; a task variable is attached to a task. - variable-value= Required; String that will become the value of the specified variable. UTF-8 character set is supported. - variable-description= Optional; Description of the variable. - task-name= Required if variable-scope=**task**; A single task name. - trigger-name= Required if variable-scope=**trigger**; A single trigger name. - create= Required; Specification (**yes** or **no**) for whether or not a new variable may be created for this request.
Example	 ``` ops-variable-set -c c.cfg variable-name=myvar1 variable-value=mydata1 create=yes ``` ⊖ **Warning** CLI only supports modifying certain types of variables - see Modifying Variables, above.

RESTful Web Services API

Overview

Opswise Controller supports a RESTful-based web services API that allows you to perform the following operations:

- Request a list of agents
- Resume agent
- Resume agent cluster
- Resume agent cluster membership
- Set agent task execution limit
- Set agent cluster task execution limit
- Suspend agent
- Suspend agent cluster
- Suspend agent cluster membership
- Change an Opswise user password
- Change a runtime password on credentials
- Set a limit on a virtual resource
- Create a task (Windows, Linux/Unix, or z/OS)
- Launch a task
- Query a list of tasks
- Trigger a task
- Create a Temporary trigger
- Enable/Disable a trigger
- Query a list of triggers
- List variables
- Set variables

Formatting specifications for each of these web services, including details about field requirements, are provided below.

Access

Access to the Controller requires basic HTTP authentication over HTTPS.

The following methods are used:

- HTTP GET is used for data look up (no data is modified).
- HTTP POST is used for data modification such as create and update.

Returned Messages

The following table identifies the status codes (part of the HTTP/1.1 standard) that can be returned.

Status Code	Description
200	Success
401	User does not have authorization (password identification failure).
403	User does not have permission to access Web Services. Access options are set via: • System Default Web Service Access Opswise Controller system property. • User Definition screen (Web Service access field)
404	Resource not found.
500	All other errors. For List Variables, 500 is returned if the Web Service fails due to invalid parameters. (A status description also is returned.)

Any other failure status codes may be returned by the underlying RESTful services. Most resources (Controller records) are returned as XML.

You can check status codes in the Audit and the opswise.log file to determine the cause if their Web Service client is not displaying the error message that comes back with the response.

User Restriction

You can be restricted from logging in to the RESTful Web Services API either of two ways:

1. The system level default for RESTful Web Services API access, specified by the System Default Web Service Access Opswise Controller system property, has been set to **No**, and the **Web Service access** field on the User Definition screen for your user account is set to **-- System Default --**."
2. The **Web Service access** field is set to **No**, which overrides the **System Default Web Service Access** value.

If either restriction is in place, the following error message will be written to the Audit table and the opswise.log file when you attempt to access the RESTful Web Services API:

```
User <your user name> not permitted to use Web Services. Please check with your administrator.
```

To remove the restriction, the system administrator must either:

- Set the System Default Web Service Access property to **Yes** and set the **Web Service access** field on the User Definition screen for your user account to **-- System Default --**.
- Set the **Web Service access** field on the User Definition screen for your user account to **Yes**.

Password Expiration

While a password is expired, RESTful Web Services API access will be prohibited until the password has been changed.

> ⚠️ **Note**
> Password expiration is not applicable to LDAP authenticated users.

Operations

The RESTful Web Service API operations are listed alphabetically within the following categories:

- Agents
- Passwords
- Resources
- Tasks
- Triggers
- Variables

Agents

Request a List of Agents

Property	Description
URI	http:/*host_name*/opswise/resources/agents/list
HTTP Method	GET
Description	Returns the list of all agents.
Authentication	HTTP Basic
Consumes Content-Type	n/a
Produces Content-Type	application/xml
XML Request Example	Not applicable.

XML Response Example	```xml
	<agents>
	    <agent>
	        <hostName>testHost</hostName>
	            <ipAddress>10.253.1.111</ipAddress>
	            <name>super2 - 1</name>
	            <queueName>AGNT0001</queueName>
	            <sysID>91149e0dc0a8016b01ddd96883a1ad9a</sysID>
	            <type>Windows</type>
	            <version>1.7.0</version>
	        </agent>
	...
	</agents>
	``

Resume Agent

Property	Description
URI	http:/host_name/opswise/resources/agents/ops-resume-agent
HTTP Method	POST
Description	Resumes the specified agent.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml

XML Request Example	**By Agent Name**
	``` <resume-agent> <agentName>XPS-8300 - AGNT0001</agentName> </resume-agent> ```
	By Agent ID
	``` <resume-agent> <agentID>AGNT0001</agentID> </resume-agent> ```
XML Response Example	``` <command-response> <type>resume_agent</type> <success>true</success> <info>Successfully resumed agent "XPS-8300 - AGNT0001".</info> <errors /> </command-response> ```

Resume Agent Cluster

Property	Description
URI	http:/host_name/opswise/resources/agents/ops-resume-agent-cluster
HTTP Method	POST
Description	Resumes the specified agent cluster.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml
XML Request Example	``` <resume-agent-cluster> <agentClusterName>Opswise - Default Windows Cluster</agentClusterName> </resume-agent-cluster> ```
XML Response Example	``` <command-response> <type>resume_agent_cluster</type> <success>true</success> <info>Successfully resumed agent cluster "Opswise - Default Windows Cluster".</info> <errors /> </command-response> ```

Resume Agent Cluster Membership

Property	Description
URI	http:/host_name/opswise/resources/agents/ops-resume-agent-cluster-membership

HTTP Method	POST
Description	Resumes the specified agent cluster membership.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml
XML Request Example	**By Agent Name** ``` <resume-agent-cluster-membership> <agentClusterName>Opswise - Default Windows Cluster</agentClusterName> <agentName>XPS-8300 - AGNT0001</agentName> </resume-agent-cluster-membership> ``` **By Agent ID** ``` <resume-agent-cluster-membership> <agentClusterName>Opswise - Default Windows Cluster</agentClusterName> <agentID>AGNT0001</agentID> </resume-agent-cluster-membership> ```
XML Response Example	``` <command-response> <type>resume_agent_cluster_membership</type> <success>true</success> <info>Successfully resumed agent cluster membership.</info> <errors /> </command-response> ```

Set Agent Cluster Task Execution Limit

Property	Description
URI	http:/host_name/opswise/resources/agents/ops-set-agent-cluster-task-execution-limit
HTTP Method	POST
Description	Sets the task execution limit for the specified agent cluster.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml

XML Request Example	**Limit Type: Limited**
| | ```
<set-agent-cluster-task-execution-limit>
  <agentClusterName>Opswise - Default Windows Cluster</agentClusterName>
  <limitAmount>15</limitAmount>
  <limitType>Limited</limitType>
</set-agent-cluster-task-execution-limit>
``` |
| | **Limit Type: Unlimited** |
| | ```
<set-agent-cluster-task-execution-limit>
  <agentClusterName>Opswise - Default Windows Cluster</agentClusterName>
  <limitType>Unlimited</limitType>
</set-agent-cluster-task-execution-limit>
``` |
| XML Response Example | ```
<command-response>
  <type>set_agent_cluster_task_execution_limit</type>
  <success>true</success>
  <info>Successfully updated agent cluster"Opswise - Default Windows Cluster" with
task limit type: "Limited", amount: "15".</info>
  <errors />
</command-response>
``` |

Set Agent Task Execution Limit

Property	Description
URI	http:/host_name/opswise/resources/agents/ops-set-agent-task-execution-limit
HTTP Method	POST
Description	Sets the task execution limit for the specified agent.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml

XML Request Example	**By Agent Name** **Limit Type: Limited**
	``` <set-agent-task-execution-limit> <agentName>XPS-8300 - AGNT0001</agentName> <limitAmount>10</limitAmount> <limitType>Limited</limitType> </set-agent-task-execution-limit> ```
	By Agent ID **Limit Type: Limited**
	``` <set-agent-task-execution-limit> <agentID>AGNT0001</agentID> <limitAmount>10</limitAmount> <limitType>Limited</limitType> </set-agent-task-execution-limit> ```
	By Agent ID **Limit Type: Unlimited**
	``` <set-agent-task-execution-limit> <agentID>AGNT0001</agentID> <limitType>Unlimited</limitType> </set-agent-task-execution-limit> ```
XML Response Example	``` <command-response> <type>set_agent_task_execution_limit</type> <success>true</success> <info>Successfully updated agent "XPS-8300 - AGNT0001" with task limit type: "Limited", amount: "10".</info> <errors /> </command-response> ```

Suspend Agent

Property	Description
URI	http:/host_name/opswise/resources/agents/ops-suspend-agent
HTTP Method	POST
Description	Suspends the specified agent.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml

XML Request Example	**By Agent Name**
	``` <suspend-agent> <agentName>XPS-8300 - AGNT0001</agentName> </suspend-agent> ```
	By Agent ID
	``` <suspend-agent> <agentID>AGNT0001</agentID> </suspend-agent> ```
XML Response Example	``` <command-response> <type>suspend_agent</type> <success>true</success> <info>Successfully suspended agent "XPS-8300 - AGNT0001".</info> <errors /> </command-response> ```

Suspend Agent Cluster

Property	Description
URI	http:/host_name/opswise/resources/agents/ops-suspend-agent-cluster
HTTP Method	POST
Description	Suspends the specified agent cluster.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml
XML Request Example	``` <suspend-agent-cluster> <agentClusterName>Opswise - Default Windows Cluster</agentClusterName> </suspend-agent-cluster> ```
XML Response Example	``` <command-response> <type>suspend_agent_cluster</type> <success>true</success> <info>Successfully suspended agent cluster "Opswise - Default Windows Cluster".</info> <errors /> </command-response> ```

Suspend Agent Cluster Membership

Property	Description
URI	http:/host_name/opswise/resources/agents/ops-suspend-agent-cluster-membership

HTTP Method	POST
Description	Suspends the specified agent cluster membership.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml
XML Request Example	**By Agent Name** ``` <suspend-agent-cluster-membership> <agentClusterName>Opswise - Default Windows Cluster</agentClusterName> <agentName>XPS-8300 - AGNT0001</agentName> </suspend-agent-cluster-membership> ``` **By Agent ID** ``` <suspend-agent-cluster-membership> <agentClusterName>Opswise - Default Windows Cluster</agentClusterName> <agentID>AGNT0001</agentID> </suspend-agent-cluster-membership> ```
XML Response Example	``` <command-response> <type>suspend_agent_cluster_membership</type> <success>true</success> <info>Successfully suspended agent cluster membership.</info> <errors /> </command-response> ```

Passwords

Change an Opswise User Password

Property	Description
URI	http:/host_name/opswise/resources/opswise/ops-change-user-password
HTTP Method	POST
Description	Changes an Opswise user password.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml
XML Request Example	``` <change-user-password> <userId>test.password</userId> <newPassword>test</newPassword> </change-user-password> ```

XML Response Example	
| | ```
<command-response>
    <type>change_user_password</type>
    <success>true</success>
</command-response>
``` |

> ⚠ **Note**
> The Authentication used must have the ops_admin user role.

Change Runtime Password on Credentials

Property	Description
URI	http:/host_name/opswise/resources/credentials/ops-change-password
HTTP Method	POST
Description	Changes the runtime password of the credentials based on name.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml
| XML Request Example | ```
<change-credentials-password>
   <name>test</name>
   <newRuntimePassword>test2</newRuntimePassword>
</change-credentials-password>
``` |
| XML Response Example | ```
<command-response>
    <type>change_credentials_password</type>
    <success>true</success>
</command-response>
``` |

> ⚠ **Note**
> The user must have the **Update** Credential permission.

Resources

Set Limit on a Virtual Resource

Property	Description
URI	http:/host_name/opswise/resources/virtual/ops-update-resource-limit
HTTP Method	POST
Description	Updates resource limit for virtual resources.
Authentication	HTTP Basic

Consumes Content-Type	application/xml
Produces Content-Type	application/xml
XML Request Example	<pre><code><virtual-resource>
<name>Opswise - Solo</name>
<limit>1</limit>
</virtual-resource></code></pre>
XML Response Example	<pre><code><command-response>
<type>update_resource_limit</type>
<success>true</success>
<info>Successfully updated the virtual resource.</info>
<errors />
</command-response></code></pre>

Tasks

Create a Task (Windows, Linux/Unix, or z/OS)

Property	Description
URI	http:/*host_name*/opswise/resources/task/ops-task-create
HTTP Method	POST
Description	Creates a new task (Windows, Linux/Unix, or z/OS) with or without related lists of records (Variables, Notes, Virtual Resources).
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml
XML Request Example - Common Fields (See type-specific properties below)	<pre><code><task type="windows">
<properties>
<entry>
<key>summary</key>
<value>test summary</value>
</entry>
<entry>
<key>name</key>
<value>test task</value>
</entry>
<entry>
<key>command</key>
<value>echo 'Hello'</value>
</entry>
...
</properties>
</task></code></pre>

XML Request Example - Task Variables	
| | ```
<variables>
    <variable>
        <description>desc</description>
        <name>testVar</name>
        <value>testVar Val</value>
    </variable>
</variables>
``` |
| XML Request Example - Notes | ```
<notes>
    <note>
        <text>note text</text>
        <title>note title</title>
    </note>
</notes>
``` |
| XML Request Example - Virtual Resources | ```
<virtual-resources>
  <virtual-resource>
    <amount>2</amount>
    <resource>testresource</resource>
  </virtual-resource>
</virtual-resources>
``` |
| XML Request Example - Actions - Email Notifications | ```
<actions>
    <email-notification>
        <exitCodes>1</exitCodes>
        <notifyOnEarlyFinish>true</notifyOnEarlyFinish>
        <notifyOnLateFinish>true</notifyOnLateFinish>
        <notifyOnLateStart>true</notifyOnLateStart>
        <status>DEFINED,FAILED</status>
        <attachFile>true</attachFile>
        <attachStdError>true</attachStdError>
        <attachStdOut>true</attachStdOut>
        <bcc>bcc@ws.stonebranch.com</bcc>
        <body>Ws test body</body>
        <cc>cc@ws.stonebranch.com</cc>
        <emailConnection> Gmail Account</emailConnection>
        <fileName>test file</fileName>
        <fileNumLines>100</fileNumLines>
        <fileScanText>test file scan text</fileScanText>
        <fileStartLine>1</fileStartLine>
        <replyTo>replyto@ws.stonebranch.com</replyTo>
        <stderrNumLines>100</stderrNumLines>
        <stderrScanText>test stderr scan text</stderrScanText>
        <stderrStartLine>1</stderrStartLine>
        <stdoutNumLines>100</stdoutNumLines>
        <stdoutScanText>test stdout scan text</stdoutScanText>
        <stdoutStartLine>1</stdoutStartLine>
        <subject>WS Test Subject</subject>
        <to>to@ws.stonebranch.com</to>
    </email-notification>
</actions>
``` |

| XML Request Example - Actions - SNMP Notification | ```
<actions>
     <snmp-notification>
          <exitCodes>1</exitCodes>
          <notifyOnEarlyFinish>true</notifyOnEarlyFinish>
          <notifyOnLateFinish>true</notifyOnLateFinish>
          <notifyOnLateStart>false</notifyOnLateStart>
          <status>DEFINED,FAILED </status>
          <severity>3</severity>
          <snmpManager>test</snmpManager>
     </snmp-notification>
</actions>
``` |
|---|---|
| XML Request Example - Actions - Abort Action | ```
<actions>
     <abort-action>
          <exitCodes>1</exitCodes>
          <notifyOnEarlyFinish>true</notifyOnEarlyFinish>
          <notifyOnLateFinish>true</notifyOnLateFinish>
          <notifyOnLateStart>true</notifyOnLateStart>
          <status> WAITING,RESOURCE REQUESTED</status>
          <cancelProcess>true</cancelProcess>
          <overrideExitCode>4</overrideExitCode>
     </abort-action>
</actions>
``` |
| XML Request Example - Actions - Set Variable | ```
<actions>
     <set-variable-action>
          <exitCodes>1</exitCodes>
          <notifyOnEarlyFinish>false</notifyOnEarlyFinish>
          <notifyOnLateFinish>false</notifyOnLateFinish>
          <notifyOnLateStart>false</notifyOnLateStart>
          <status>finished</status>
          <variableDescription>testvarDescr</variableDescription>
          <variableName>testVar</variableName>
          <variableScope>1</variableScope>
          <variableValue>testVarVal</variableValue>
     </set-variable-action>
</actions>
``` |
| XML Request Example - Actions - System Operation | ```
<actions>
     <system-operation>
          <exitCodes>1</exitCodes>
          <status>WAITING,RESOURCE REQUESTED</status>
          <notifyOnEarlyFinish>true</notifyOnEarlyFinish>
          <notifyOnLateFinish>true</notifyOnLateFinish>
          <notifyOnLateStart>true</notifyOnLateStart>
          <description>Reset Task Action</description>
          <operation>Set Virtual Resource Limit</operation>
          <virtualResource>QA-Test-Virtual-Resource
Limit</virtualResource>
          <limit>27</limit>
          <notificationOption>Operation
Success/Failure</notificationOption>
     </system-operation>
</actions>
``` |

Windows Task Properties	
| | ```
name - required / unique
summary
credentials
credentials_var
retry_maximum
retry_indefinitely
retry_interval
opswise_groups - sys_id of Business Service
ls_enabled
ls_type
ls_time - format hh:mm
ls_duration - format hh:mm:ss
lf_enabled
lf_type
lf_time - format hh:mm
lf_duration - format hh:mm:ss
ef_enabled
ef_type
ef_time - format hh:mm
ef_duration - format hh:mm:ss
start_held
start_held_reason
agent - required
agent_var
agent_cluster
agent_cluster_var
broadcast_cluster
command_or_script
command
script
runtime_dir
parameters
environment
exit_code_processing
exit_codes
exit_code_text
exit_code_output
output_type
output_return_type
output_return_file
output_return_sline
output_return_nline
output_return_text
``` |

Linux/Unix Task Properties	
| | ```
name - required /unique
summary
credentials
credentials_var
retry_maximum
retry_indefinitely
retry_interval
opswise_groups - sys_id of Business Service
ls_enabled
ls_type
ls_time - format hh:mm
ls_duration - format hh:mm:ss
lf_enabled
lf_type
lf_time - format hh:mm
lf_duration - format hh:mm:ss
ef_enabled
ef_type
ef_time - format hh:mm
ef_duration - format hh:mm:ss
start_held
start_held_reason
agent_var
agent_cluster
agent_cluster_var
agent
broadcast_cluster
command_or_script
command
script
runtime_dir
parameters
environment
exit_code_processing
exit_codes
exit_code_text
exit_code_output
output_type
run_as_sudo
output_return_type
output_return_file
output_return_sline
output_return_nline
output_return_text
``` |

z/OS Task Properties	
| | ```
name
summary
credentials
credentials_var
retry_maximum
retry_indefinitely
retry_interval
opswise_groups - sys_id of Business Service
ls_enabled
ls_type
ls_time
ls_duration - format hh:mm:ss
lf_enabled
lf_type
lf_time - format hh:mm
lf_duration - format hh:mm:ss
ef_enabled
ef_type
ef_time - format hh:mm
ef_duration - format hh:mm:ss
start_held
start_held_reason
agent_var
agent_cluster
agent_cluster_var
agent
broadcast_cluster
jobname_new
jobclass_new
msgclass_new
jcl_location
jcl
parameters
exit_code_processing
exit_codes
exit_code_text
exit_code_output
output_type
schedule_id
proclib_name
output_return_type
output_return_file
output_return_sline
output_return_nline
output_return_text
``` |

Fixed Type Values	**Valid Type Values:** **exit_code_processing types:** Success Exitcode Range (1) Failure Exitcode Range (2) Success Output Contains (3) Failure Output Contains (4) **output_types:** STDOUT (1) STDERR (2) FILE (3) JOBLOG (4) SCRIPT (5) **output_return_types:** NONE (1) STDOUT (2) - Windows and Linux/Unix only STDERR (3) - Windows and Linux/Unix only FILE (4) JOBLOG (5) - z/OS only OUTERR (6) - Windows and Linux/Unix only **command_or_script types:** Command (1) Script (2) ⚠️ **Note** If command_or_script = 1, you must specify a command; if command_or_script = 2, you must specify a script.
| XML Response Example | ```xml
<command-response>
    <type>create_task</type>
    <success>true</success>
</command-response>
``` |

Launch a Task

Property	Description
URI	http://*host_name*/opswise/resources/task/ops-task-launch
HTTP Method	POST
Description	Launches the specified task. Variables optional.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml

XML Request Example	```xml
	<task-launch>
	    <name>Sleep 0</name>
	    <variables>
	        <variable>
	            <name>testVariable</name>
	            <value>Hello</value>
	        </variable>
	    </variables>
	</task-launch>
	``

XML Response Example	```xml
	<command-response>
	    <type>launch</type>
	    <success>true</success>
	    <info>Successfully launched the Sleep task Sleep 0 with Execid
	f505b243c0a801af0026913343d8dcfa.</info>
	    <errors />
	</command-response>
	``

Query a List of Tasks

Property	Description
URI	http:/host_name/opswise/resources/task/list
HTTP Method	POST
Description	Lists the specified task.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml

XML Request Example	name and type are optional.
	
	``xml
	<generic-query-filter>
	    <name>*0</name>
	    <type>Sleep</type>
	</generic-query-filter>
	``

XML Response Example	```xml
	<tasks>
	    <task>
	        <name>Sleep 0</name>
	        <sysID>410d6c0bc0a801c901838d8ac43b3279</sysID>
	        <type>Sleep</type>
	    </task>
	    <task>
	        <name>Sleep 30</name>
	        <sysID>410d6880c0a801c90196685fcc1ecb47</sysID>
	        <type>Sleep</type>
	    </task>
	</tasks>
	``

Trigger a Task

Property	Description

URI	http://*host_name*/opswise/resources/trigger/ops-trigger-now
HTTP Method	POST
Description	Launches trigger-associated tasks and optionally sets variables.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml

XML Request Example	<pre><trigger-now>
	    <name>Opswise - Every 15 Minutes MWF 9-5</name>
	    <variables>
	        <variable>
	            <name>testVariable</name>
	            <value>Hello</value>
	        </variable>
	    </variables>
	</trigger-now></pre>

XML Response Example	<pre><command-response>
	    <type>trigger_now</type>
	    <success>true</success>
	    <info>Successfully triggered Opswise - Every 15 Minutes MWF 9-5 with sys_id
	4c7d6b9cc0a801c900e7dac8b24ef142.</info>
	    <errors />
	</command-response></pre>

Triggers

Create a Temporary Trigger

Property	Description
URI	http:/host_name/opswise/resources/trigger/ops-create-temp-trigger
HTTP Method	POST
Description	Creates a temporary trigger. Variables are optional.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml

XML Request Example	``` <create-temp-trigger> <date>2015-01-15</date> <enable>true</enable> <name>Test Temp Trigger</name> <tasks> <task name="Sleep 30"/> </tasks> <time>13:42</time> <variables> <variable> <name>var56318389</name> <value>val</value> </variable> </variables> </create-temp-trigger> ```
XML Response Example	``` <command-response> <type>create_trigger</type> <success>true</success> <info>Successfully created the Temporary trigger., Successfully enabled trigger Test Temp Trigger</info> <errors /> </command-response> ```

Enable/Disable a Trigger

Property	Description
URI	http:/host_name/opswise/resources/trigger/ops-enable-disable-trigger
HTTP Method	POST
Description	Enables/Disables a list of triggers.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml
XML Request Example	``` <enable-disable-trigger> <trigger enable="true" name="Opswise - 1st and 15th"></trigger> <trigger enable="false" name="Opswise - 1st Sunday of Month"></trigger> </enable-disable-trigger> ```
XML Response Example	``` <command-response> <type>Enable Disable Triggers</type> <success>true</success> <info>Successfully enabled trigger Opswise - 1st and 15th, Successfully disabled trigger Opswise - 1st Sunday of Month</info> <errors /> </command-response> ```

Query a List of Triggers

Property	Description
URI	http:/host_name/opswise/resources/trigger/list
HTTP Method	POST
Description	Lists the specified trigger.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml

XML Request Example	`name`, `enabled`, and `type` are optional.
	
	``xml
	<generic-query-filter>
	    <name>*time*</name>
	    <enabled>true</enabled>
	    <type>Time</type>
	</generic-query-filter>
	``

XML Response Example	```xml
	<triggers>
	    <trigger>
	        <enabled>true</enabled>
	        <name>Example time trigger 1</name>
	        <sysID>07e69e9ac0a830016f4e065eb8e4ffe0</sysID>
	        <type>Time</type>
	    </trigger>
	    <trigger>
	        <enabled>true</enabled>
	        <name>Example time trigger 2</name>
	        <sysID>07e69051c0a8300135a540b2d5b07ade</sysID>
	        <type>Time</type>
	    </trigger>
	</triggers>
	``

Variables

List Variables

Property	Description
URI	http:/host_name/opswise/resources/variable/list
HTTP Method	POST
Description	Lists the specified variable.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml

XML Request Example	`variableName` is optional in all cases.

```
Global Variable
<variable-query-filter>
    <scope>global</scope>
</variable-query-filter>

<variable-query-filter>
    <scope>GLOBAL</scope>
    <variableName>demo*</variableName>
</variable-query-filter>

Local Variable
<variable-query-filter>
    <scope>LOCAL</scope>
</variable-query-filter>

<variable-query-filter>
    <scope>LOCAL</scope>
    <variableName>*2</variableName>
</variable-query-filter>

Task Variable
<variable-query-filter>
    <scope>TASK</scope>
    <taskName>UT-WS-Task</taskName>
</variable-query-filter>

<variable-query-filter>
    <scope>TASK</scope>
    <taskName>UT-WS-Task</taskName>
    <variableName>UT*2</variableName>
</variable-query-filter>

Trigger Variable
<variable-query-filter>
    <scope>TRIGGER</scope>
    <triggerName>UT-WS-Trigger</triggerName>
</variable-query-filter>

<variable-query-filter>
    <scope>TRIGGER</scope>
    <triggerName>UT-WS-Trigger</triggerName>
    <variableName>*2</variableName>
</variable-query-filter>
```

XML Response Example	`type` is either GLOBAL, Task, or Trigger.
| | ```
Example 1
<variables>
    <variable>
        <description></description>
        <name>demo_ops_download_dir</name>
        <sysId>db5c436ac0a8016501f1ab660b585961</sysId>
        <type>Global</type>
        <value>/home/opswise/download</value>
    </variable>
    <variable>
        <description></description>
        <name>demo_ops_linux_rc</name>
        <sysId>db5c7b72c0a80165004cda781dbf3115</sysId>
        <type>Global</type>
        <value>0</value>
    </variable>
</variables>

Example 2
<variables>
    <variable>
        <description>varDescription2</description>
        <name>UT_WS_var_2</name>
        <sysId>164a03300a0a0a6475b497e5d985c1aa</sysId>
        <taskId>164a02a80a0a0a644fc9a3d6dd33f72b</taskId>
        <taskName>UT-WS-Task - 9103c2e0-4395-4c75-84e8-a5da8dfb0244</taskName>
        <triggerId></triggerId>
        <type>Task</type>
        <value>varVal2</value>
    </variable>
    <variable>
        <description></description>
        <name>UT_WS_var_2</name>
        <sysId>164ea9ff0a0a0a6449f209da49f594c0</sysId>
        <taskId></taskId>
        <triggerId>164ea9a40a0a0a647df8372b88eb2394</triggerId>
        <triggerName>UT-WS-Trigger -
85777e3e-cd4b-4d29-9810-6974182de21e</triggerName>
        <type>Trigger</type>
        <value>val</value>
    </variable>
</variables>
``` |

Set Variables

Property	Description
URI	http://*host_name*/opswise/resources/variable/ops-variable-set
HTTP Method	POST
Description	Sets the specified variable.
Authentication	HTTP Basic
Consumes Content-Type	application/xml
Produces Content-Type	application/xml

XML Request Example	

	Global Variable
	```<set-variable>```
	``` <create>true</create>```
	``` <variable>```
	``` <name>testGlobalVar</name>```
	``` <description>testGlobalVar decription</description>```
	``` <value>testGlobalVar val</value>```
	``` </variable>```
	```</set-variable>```
	
	Task Variable
	```<set-variable>```
	``` <create>true</create>```
	``` <scope>task</scope>```
	``` <task>Sleep 60</task>```
	``` <variable>```
	``` <description>testTaskVar decription</description>```
	``` <name>testTaskVar</name>```
	``` <value>testTaskVar val</value>```
	``` </variable>```
	```</set-variable>```
	
	Trigger Variable
	```<set-variable>```
	``` <create>true</create>```
	``` <scope>trigger</scope>```
	``` <trigger>Opswise - Mon-Fri 9-5</trigger>```
	``` <variable>```
	``` <description>testTriggerVar decription</description>```
	``` <name>testTriggerVar</name>```
	``` <value>testTriggerVar val</value>```
	``` </variable>```
	```</set-variable>```

XML Response Example	

	```<command-response>```
	``` <type>set_variable</type>```
	``` <success>true</success>```
	``` <info>Variable testGlobalVar not found., Successfully created```
	```variable=testGlobalVar with value=testGlobalVar val.</info>```
	``` <errors />```
	```</command-response>```