Use of Kaspersky Security Center OpenAPI

This article describes the options of applying Kaspersky Security Center OpenAPI methods for automation of deployment and use of Kaspersky Security for Virtualization 5.1 Light Agent in the multitenancy mode.

The detailed description of Kaspersky Security Center OpenAPI features can be found in the **kscopenapi.chm** file located in the installation folder of Kaspersky Security Center. The default path to the installation folder is C:\Program Files (x86)\Kaspersky Lab\Kaspersky Security Center.

Kaspersky Security Center OpenAPI use scenarios

Moving a virtual machine to the virtual Kaspersky Security Center Administration Server

The procedure allows moving virtual machines to the Administration Server under the following conditions:

- Kaspersky Security Center Network Agent must be installed on a virtual machine. The Network Agent must have the address of the main Administration Server specified in its settings.

- A virtual machine must have a unique ID.

- A virtual machine is present on the main Kaspersky Security Center Administration Server (for example, in the Unassigned devices group).

To move a virtual machine to the virtual Kaspersky Security Center Administration Server:

1. Log in to the main Kaspersky Security Center Administration Server, using the Login option.

2. Get the list of virtual machines. To do that, call the HostGroup.FindHosts method with the following fields:

    - **KLHST_WKS_DN** – virtual machine display name.
    - **KLHST_WKS_HOSTNAME** – virtual machine's unique ID.
    - **KLHST_WKS_IP_LONG** – IP address of the virtual machine.

3. In the returned list, find the virtual machine by its IP address in the **KLHST_WKS_IP_LONG** field and remember the values in the **KLHST_WKS_DN** and **KLHST_WKS_HOSTNAME** fields.

4. Log in to the virtual Kaspersky Security Center Administration Server, using the Login-VirtualServer option.

5. Create a Change Administration Server task on a virtual Kaspersky Security Center Administration Server and remember its **taskId**:

    a. In the **NewKscServerAddress** field, add the line of the following type: **MainKscServerAddress/VirtualServerName**.

    b. Assign the values you have saved at step 3: **HostDispName** = **KLHST_WKS_DN** and **HostName = KLHST_WKS_HOSTNAME**.

    c. Specify the displayed task name in the **DisplayName** field.

6. Run the task and specify its **taskId** that you have saved at the previous step.

7. Wait for the task to complete.

Distributing an installation package to the virtual Administration Server

To distribute an installation package to the virtual Administration Server:

1. Use the PackagesApi.GetPackages method to find the **KLPKG_NPI_PKGID** package ID for the **KLPKG_NPI_PRODUCT_NAME** application of version **KLPKG_NPI_PRODUCT_VERSION**.

2. Use the VServers.GetVServers method to find the virtual Administration Server ID, **KLVSRV_ID**, which corresponds to the name **KLVSRV_DN**.

3. Run the distribution of a package to the virtual Administration Server by using the PackagesApi.RetranslateToVServerAsync method. The ID of an asynchronous operation is returned.

4. To check the state of the asynchronous operation using its ID received at the previous step, call the AsyncActionStateChecker.CheckActionState method.

Installing an application to a virtual machine

The procedure allows performing remote installation on a virtual machine under the following conditions:

- Kaspersky Security Center Network Agent must be installed on a virtual machine.
- A virtual machine is present on the virtual Administration Server (for example, in the Unassigned devices group).
- Installation packages for Network Agent and Kaspersky Security for Virtualization 5.1 Light Agent are present on the virtual server. If there are no packages, create them or send them to the corresponding virtual Server.

To install Kaspersky Security for Virtualization 5.1 Light Agent:

1. Log in to the Administration Server from which the remote installation task will be run. To log in to the main Administration Server, use the Login method, and to log in to the virtual Administration Server, use the Login-VirtualServer method.

2. Get the list of virtual machines. To do that, call the HostGroup.FindHosts method with the following fields:

- **KLHST_WKS_DN** – virtual machine display name.
- **KLHST_WKS_HOSTNAME** – virtual machine's unique ID.
- **KLHST_WKS_IP_LONG** – IP address of a virtual machine.

3. In the returned list, find the virtual machine by its IP address in the **KLHST_WKS_IP_LONG** field and remember the values in the **KLHST_WKS_DN** and **KLHST_WKS_HOSTNAME** fields.

4. Using the PackagesApi.GetPackages method, find the ID of the installation package **KLPKG_NPI_PKGID**for the Network Agent **KLPKG_NPI_PRODUCT_NAME** = 1103 and version **KLPKG_NPI_PRODUCT_VERSION =** 1.0.0.0 (see the PackagesIds table).

5. Using the PackagesApi.GetPackages method, find the ID of the installation package **KLPKG_NPI_PKGID** for application Kaspersky Security for Virtualization 5.1 Light Agent **KLPKG_NPI_PRODUCT_NAME** = KSVLA and version **KLPKG_NPI_PRODUCT_VERSION =** 5.1.0.0 (see the PackagesIds table).

6. Use the HostGroup.FindGroups or HostGroup.GroupIdGroups method to specify the administration group to which the virtual machine should be moved after application installation.

7. Create a remote installation task and remember its **taskId**:

   a. Assign the value **KLPKG_NPI_PKGID** that you have saved at step 4 to **KlNagentPackageId**.

   b. Assign the value **KLPKG_NPI_PKGID** that you have saved at step 5 to **ProductPackageId**.

   c. Assign the values you have saved at step 3: **HostDispName** = **KLHST_WKS_DN** and **HostName = KLHST_WKS_HOSTNAME**.

   d. Specify the user name and password for the account under which the installation will be performed in the **HostOsUserLogin** and **HostOsUserPassword** fields.

   e. In the **GroupToMoveHostId**, specify the administration group to which the virtual machine from step 6 will be moved.

      f.   Specify the displayed task name in the **DisplayName** field.

8. [Run the task](#) and specify its **taskId** that you have saved at the previous step.

9. Wait for the task to complete.

Uninstalling an application from a virtual machine

To uninstall Kaspersky Security for Virtualization 5.1 Light Agent through the Network Agent:

1. Log in to the Administration Server from which the remote installation task will be run. To log in to the main Administration Server, use the [Login](#) method, and to log in to the virtual Administration Server, use the [Login-VirtualServer](#) method.

2. Get the list of virtual machines. To do that, call the [HostGroup.FindHosts](#) method with the following fields:

   - **KLHST_WKS_DN** – virtual machine display name.
   - **KLHST_WKS_HOSTNAME** – virtual machine's unique ID.
   - **KLHST_WKS_IP_LONG** – IP address of a virtual machine.

3. In the returned list, find the virtual machine by its IP address in the **KLHST_WKS_IP_LONG** field and remember the values in the **KLHST_WKS_DN** and **KLHST_WKS_HOSTNAME** fields.

4.

   a. Assign the values you have saved at step 3: **HostDispName** = **KLHST_WKS_DN** and **HostName = KLHST_WKS_HOSTNAME**.

   b. Specify the user name and password for the account under which the uninstallation will be performed in the **HostOsUserLogin** and **HostOsUserPassword** fields.

   c. Specify the information about the application to remove and its version in the fields **ProductName** (for example, KSVLA) and **ProductVersion** (for example, 5.1.0.0).

   d. Specify the displayed task name in the **DisplayName** field.

5. [Run the task](#) and specify its **taskId** that you have saved at the previous step.

6. Wait for the task to complete.

Removing a virtual machine from the protected infrastructure

Before removing the virtual machine, you will need to [uninstall the application installed on this virtual machine](#).

To remove the virtual machine from the protected infrastructure:

1. Log in to the virtual Kaspersky Security Center Administration Server, on which the virtual machine is located, using the [Login-VirtualServer](#) method.

2. Get the list of virtual machines. To do that, call the [HostGroup.FindHosts](#) method with the following fields:

   - **KLHST_WKS_DN** – virtual machine display name.
   - **KLHST_WKS_HOSTNAME** – virtual machine's unique ID.
   - **KLHST_WKS_IP_LONG** – IP address of a virtual machine.

3. In the returned list, find the virtual machine by its IP address in the **KLHST_WKS_IP_LONG** field and remember the values in the **KLHST_WKS_DN** and **KLHST_WKS_HOSTNAME** fields.

4. If you want to remove the virtual machine from the Managed devices group and move it to the Unassigned devices group:

a. To get the **id** of the Unassigned devices group, call the [HostGroup.GroupIdUnassigned](#) method.

b. call the [method of moving a virtual machine](#) to the administration group. Add the**KLHST_WKS_HOSTNAME** value received at step 3 to the **hostId** field, and add the **id** value received at step 4a to the **groupId** field.

5. If you want to delete the virtual machine completely, call the [virtual machine removal method](#). Use the value **KLHST_WKS_HOSTNAME** received at step 3 to the **hostId** field.

Used Kaspersky Security Center OpenAPI methods

Logging in to the main Kaspersky Security Center Administration Server

To use Kaspersky Security Center OpenAPI, you must log in to the Kaspersky Security Center Administration Server. To do that, call the **login** method:

```
POST https://MainKscServerIpAddress:Port/api/v1.0/login HTTP/1.1
Authorization: KSCBasic user="Base64UserKscLogin",
pass="Base64UserKscPassword", internal="0"
Content-Type: application/json
Content-Length: 2

{}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the Kaspersky Security Center OpenAPI port on the main Administration Server. Port **13299** is used by default.
- **Base64UserKscLogin** – user name of the main Administration Server in the Base64 format.
- **Base64UserKscPassword** – password of the main Administration Server in the Base64 format.

Logging in to the virtual Kaspersky Security Center Administration Server

```
POST https://MainKscServerIpAddress:Port/api/v1.0/login HTTP/1.1
Authorization: KSCBasic user="Base64UserVirtualKscLogin",
pass="Base64UserVirtualKscPassword", internal="0"
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: 2
{}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the OpenAPI Kaspersky Security Center port on the main Administration Server. Port **13299** is used by default.
- **Base64UserVirtualKscLogin** – user name of the main Administration Server in the Base64 format.
- **Base64UserVirtualKscPassword** – password of the main Administration Server in the Base64 format.

- **Base64VirtualKscName** – user name of the main Administration Server in the Base64 format.

Operations with virtual machines

Finding a virtual machine on the Kaspersky Security Center Administration Server

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/HostGroup.FindHosts
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength
{
   "wstrFilter" : "",
   "vecFieldsToReturn" : [

      "KLHST_WKS_DN",
      "KLHST_WKS_HOSTNAME"
      "KLHST_WKS_IP_LONG"

   ],
   "vecFieldsToOrder" : [],
   "pParams" : {"KLSRVH_SLAVE_REC_DEPTH" : 128,
   "KLGRP_FIND_FROM_CUR_VS_ONLY" : true },
   "lMaxLifeTime" : 600

}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the Kaspersky Security Center OpenAPI port on the main Administration Server. Port **13299** is used by default.
- **Base64VirtualKscName** – user name of the main Administration Server in the Base64 format. The `X-KSC-VServer` header is required for method queries for a virtual Administration Server.
- **BodyLength** – length of the JSON body of the request in bytes.
- **vecFieldsToReturn** – an array of virtual machine properties to return. The full list of fields is available in the method documentation.

  - **KLHST_WKS_DN** – virtual machine display name.
  - **KLHST_WKS_HOSTNAME** – virtual machine's unique ID.
  - **KLHST_WKS_IP_LONG** – IP address of a virtual machine.

The method returns the ChunkAccessor enumerator object ID on the Administration Server in the format:

```
{"strAccessor" : "iteratorId"}
```

This ID allows you to get the information about the virtual machines using methods for using the enumerator. Once the work with the enumerator is finished, release the enumerator by calling the ChunkAccessor.Release method.

Moving the virtual machine to the administration group

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/HostGroup.MoveHostsToGr
oup
X-KSC-VServer: Base64VirtualKscName
```

```
Content-Type: application/json
Content-Length: BodyLength
{"pHostNames":

  [
  "hostId"
  ],

"nGroup`"": groupId
}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the Kaspersky Security Center OpenAPI port on the main Administration Server. Port **13299** is used by default.
- **Base64VirtualKscName** – user name of the main Administration Server in the Base64 format. The `X-KSC-VServer` header is required for method queries for a virtual Administration Server.
- **BodyLength** – length of the JSON body of the request in bytes.
- **hostId** – id of a virtual machine to move. Get the **KLHST_WKS_HOSTNAME** value by calling HostGroup.FindHosts.
- **groupId** – id of the administration group to which the virtual machine will be moved. Get the **id** value by calling HostGroup.FindGroups.

Deleting a virtual machine

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/HostGroup.RemoveHost
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength
{"strHostName": "hostId"}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the Kaspersky Security Center OpenAPI port on the main Kaspersky Security Center Administration Server. Port **13299** is used by default.
- **Base64VirtualKscName** – user name of the main Administration Server in the Base64 format. The `X-KSC-VServer` header is required for method queries for a virtual Administration Server.
- **BodyLength** – length of the JSON body of the request in bytes.
- **hostId** – id of a virtual machine to delete. Get the **KLHST_WKS_HOSTNAME** value by calling HostGroup.FindHosts.

Operations with the enumerator

Getting the number of enumerator items

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/ChunkAccessor.GetItemsC
ount
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength
{"strAccessor" : "iteratorId"}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the OpenAPI Kaspersky Security Center port on the main Administration Server. Port **13299** is used by default.
- **Base64VirtualKscName** – user name of the main Administration Server in the Base64 format. The `X-KSC-VServer` header is required for method queries for a virtual Administration Server.
- **BodyLength** – length of the JSON body of the request in bytes.
- **iteratorId** – enumerator ID from the **strAccessor** field in the [HostGroup.FindHosts](#) method result.

The method returns the number of enumerator items in the format:

```
{"PxgRetVal" : itemsCount}
```

Getting enumerator items

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/ChunkAccessor.GetItemsC
hunk
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength
{
   "strAccessor" : "iteratorId",
   "nStart" : from,
   "nCount" : itemsCount

}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the Kaspersky Security Center OpenAPI port on the main Administration Server. Port **13299** is used by default.
- **Base64VirtualKscName** – user name of the main Administration Server in the Base64 format. The `X-KSC-VServer` header is required for method queries for a virtual Administration Server.
- **BodyLength** – length of the JSON body of the request in bytes.
- **iteratorId** – enumerator ID from the **strAccessor** field in the [HostGroup.FindHosts](#) method result.
- **from** – the index of the collection item, starting from which the items will be returned. The count starts from 0 index.
- **itemsCount** – number of items to return. If you need to return all items, use the **itemsCount** value returned by calling the [ChunkAccessor.GetItemsCount](#) method.

The method returns the enumerator items in the following format ([HostGroup.FindHosts](#) method used as an example):

```
{
"pChunk":{
    "KLCSP_ITERATOR_ARRAY":[
    {
      "type":"params",
      "value":{
          "KLHST_WKS_DN":"Host1",
          "KLHST_WKS_HOSTNAME":"7ad995e2-eb62-40e5-9c7e-
5abae19979a0",
          "KLHST_WKS_IP_LONG":{
              "type":"long",
              "value":2130706433
```

```
                }
            }
        },
        {
            "type":"params",
            "value":{
                "KLHST_WKS_DN":"Host2",
                "KLHST_WKS_HOSTNAME":"5fb6a90c-d054-4f9b-a342-
2a62949ad899",
                "KLHST_WKS_IP_LONG":{
                    "type":"long",
                    "value":172052763
                }
            }
        }
        ]
    },
    "PxgRetVal":2
}
```

In this example, two elements ("PxgRetVal":2) with the fields specified in the
vecFieldsToReturn parameter of the [HostGroup.FindHosts](#) query are returned.

Releasing the enumerator

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/ChunkAccessor.Release
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength
{"strAccessor" : "iteratorId"}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of
  the main Kaspersky Security Center Administration Server.
- **Port** – the Kaspersky Security Center OpenAPI port on the main Administration Server. Port
  13299 is used by default.
- **Base64VirtualKscName** – user name of the main Administration Server in the Base64
  format. The X-KSC-VServer header is required for method queries for a virtual
  Administration Server.
- **BodyLength** – length of the JSON body of the request in bytes.
- **iteratorId** – enumerator ID from the **strAccessor** field in the [HostGroup.FindHosts](#) method
  result.

Operations with administration groups

Finding administration groups

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/HostGroup.FindGroups
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength
{
    "wstrFilter" : "",
    "vecFieldsToReturn" : [
        "id",
```

```
        "name"      ],
    "vecFieldsToOrder" : [],
    "pParams" : {},
    "lMaxLifeTime" : 600
}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the OpenAPI Kaspersky Security Center port on the main Administration Server. Port **13299** is used by default.
- **Base64VirtualKscName** – user name of the main Administration Server in the Base64 format. The X-KSC-VServer header is required for method queries for a virtual Administration Server.
- **BodyLength** – length of the JSON body of the request in bytes.
- **vecFieldsToReturn** – an array of administration group properties to return. The full list of fields is available in the method documentation.

  - **id** – group id.
  - **name** – group name.

The method returns the ChunkAccessor enumerator object ID on the Administration Server in the format:

```
{"strAccessor" : "iteratorId"}
```

This ID allows you to get the information about the administration groups using methods for using the enumerator. Once the work with the enumerator is finished, release the enumerator by calling the ChunkAccessor.Release method.

Getting the Managed devices group ID

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/HostGroup.GroupIdGroups
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: 2
{}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the Kaspersky Security Center OpenAPI port on the main Administration Server. Port **13299** is used by default.
- **Base64VirtualKscName** – user name of the main Administration Server in the Base64 format. The X-KSC-VServer header is required for method queries for a virtual Administration Server.

The method returns a group ID in the following format:

```
{"PxgRetVal" : groupId}
```

Getting the Unassigned devices group ID

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/HostGroup.GroupIdUnassigned
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: 2
{}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the OpenAPI Kaspersky Security Center port on the main Administration Server. Port **13299** is used by default.
- **Base64VirtualKscName** – user name of the main Administration Server in the Base64 format. The `X-KSC-VServer` header is required for method queries for a virtual Administration Server.

The method returns a group ID in the following format:

```
{"PxgRetVal" : groupId}
```


Getting the list of virtual Administration Servers

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/VServers.GetVServers

Content-Type: application/json

Content-Length: 2

{}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the Kaspersky Security Center OpenAPI port on the main Administration Server. Port **13299** is used by default.

If successful, the method returns the array of virtual Administration Servers in the following format:

```
{
    "PxgRetVal": [
      {
        "type": "params",
        "value": {
            "KLVSRV_CREATED": {
                "type": "datetime",
                "value": "2020-06-15T09:41:18Z"
             },
            "KLVSRV_DN": "SomeVirtualServer",
            "KLVSRV_ENABLED": true,
            "KLVSRV_GROUPS": 8,
            "KLVSRV_GRP": 0,
            "KLVSRV_HST_UID": "VSRV08172e1f-4057-4579-89c4-
   d5e6256d8ad2",
            "KLVSRV_ID": 1,
            "KLVSRV_LIC_ENABLED": true,
            "KLVSRV_SUPER": 7,
            "KLVSRV_UID": "VSRV08172e1f-4057-4579-89c4-d5e6256d8ad2",
            "KLVSRV_UNASSIGNED": 11
        }
      }
    ]
    }
```

Main parameters:

- **KLVSRV_DN** – virtual Administration Server name.
- **KLVSRV_ID** – virtual Administration Server ID.

Operations with installation packages

Getting the list of installation packages

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/PackagesApi.GetPackages
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: 2
{}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the Kaspersky Security Center OpenAPI port on the main Administration Server. Port **13299** is used by default.
- **Base64VirtualKscName** – user name of the main Administration Server in the Base64 format. The `X-KSC-VServer` header is required for method queries for a virtual Administration Server.

If successful, the method returns the array of installation packages in the following format:

```
{
    "PxgRetVal": [
    {
        "type": "params",
        "value": {
           "KLPKG_NPI_CREATION_TIME": {
               "type": "datetime",
               "value": "2019-10-08T17:30:40Z"
             },
            "KLPKG_NPI_MODIF_TIME": {
               "type": "datetime",
               "value": "2019-10-08T17:30:40Z"
             },
           "KLPKG_NPI_NAME": "Kaspersky Security for Virtualization
5.1 Light Agent",
           "KLPKG_NPI_PACKAGE_PATH": "\\\\DESKTOP-
MI1CJOJ\\KLSHARE\\Packages\\KSVLA_5.1.44.278",
           "KLPKG_NPI_PKGID": 1,
           "KLPKG_NPI_PRODUCT_DISPL_NAME": "Kaspersky Security for
Virtualization 5.1 Light Agent",
           "KLPKG_NPI_PRODUCT_DISPL_VERSION": "5.1.44.278",
           "KLPKG_NPI_PRODUCT_NAME": "KSVLA",
           "KLPKG_NPI_PRODUCT_VERSION": "5.1.0.0",
           "KLPKG_NPI_SIZE": {
               "type": "long",
               "value": 217427607
             },
            "KLPKG_NPI_SS_DESCR":
"KSVLA_5.1.44.278\\exec\\ss_install.xml|1"
        }
    },
....
    ]
}
```

Main parameters:

- **KLPKG_NPI_PKGID** – installation package ID.
- **KLPKG_NPI_PRODUCT_NAME** – name of the application to be installed from the installation package.
- **KLPKG_NPI_PRODUCT_VERSION** – version of the application to be installed from the installation package.

Installation packages of Kaspersky Security Center Network Agent and Kaspersky Security application

To identify the packages of Network Agent and Kaspersky Security for Virtualization 5.1 Light Agent when calling Kaspersky Security Center OpenAPI methods, use the following table:

Package property name	Network Agent	KSVLA 5.1
KLPKG_NPI_PRODUCT_NAME	1103	KSVLA
KLPKG_NPI_PRODUCT_VERSION	1.0.0.0	5.1.0.0

Distribution of an installation package to the virtual Administration Server

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/PackagesApi.Retranslate
ToVServerAsync
Content-Type: application/json
Content-Length: BodyLength
{
    "nPackageId": packageId,
    "nVServerId": VirtualServerId,
    "pOptions":
    {
        "KLPKG_CREATE_STANDALONE_PRODS" : true,
        "KLPKG_CREATE_STANDALONE_NAGT" : true,
        "KLPKG_USE_LANGUAGE_TAG" : true, `
        "KLPKG_TYPE" : 1,
        "KLPKG_LAZY_RETRANSLATION" : false
    }
}
```

where:

MainKscServerIpAddress – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.

Port – the Kaspersky Security Center OpenAPI port on the main Administration Server. Port **13299** is used by default.

BodyLength – length of the JSON body of the request in bytes.

packageId – ID of the installation package to be distributed to the virtual Administration Server. **KLPKG_NPI_PKGID** value is received by calling the PackagesApi.GetPackages method.

VirtualServerId – ID of the virtual Administration Server to which the package should be distributed. **KLVSRV_ID** value is received by calling the VServers.GetVServers method.

If successful, the method returns the asynchronous operation ID in the format:

```
{"PxgRetVal" : "asyncActionId"}
```

To check if the asynchronous package distribution operation was successful, call the AsyncActionStateChecker.CheckActionState method.

Checking the state of the asynchronous operation

```
POST
https://MainKscServerIpAddress:Port/api/v1.0/AsyncActionStateChecker
.CheckActionState
Content-Type: application/json
Content-Length: BodyLength
{"wstrActionGuid": "asyncActionId"}
```

where:

MainKscServerIpAddress – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.

Port – the Kaspersky Security Center OpenAPI port on the main Administration Server. Port **13299** is used by default.

BodyLength – length of the JSON body of the request in bytes.

asyncActionId – asynchronous operation ID, received by calling the PackagesApi.RetranslateToVServerAsync method.

If successful, the method returns the asynchronous operation state in the format:

```
{
"bFinalized" : true
"bSuccededFinalized" : true
"lStateCode" : 0
"pStateData": {}
"lNextCheckDelay" : 0
}
```

where:

bFinalized – operation state. If the value is `true`, then the operation is completed.

bSuccededFinalized – attribute of the successful operation completion. If the value is `true`, then the operation is completed successfully.

Operations with tasks

Creating the Change Kaspersky Administration Server task

The task allows changing the Administration Server which controls the virtual machine.

```
POST https://MainKscServerIpAddress:Port/api/v1.0/Tasks.AddTask
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength
{"pData" : taskParams}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the Kaspersky Security Center OpenAPI port on the main Kaspersky Security Center Administration Server. Port **13299** is used by default.
- **Base64VirtualKscName** – user name of the main Administration Server in the Base64 format. The `X-KSC-VServer` header is required for method queries for a virtual Administration Server.
- **BodyLength** – length of the JSON body of the request in bytes.
- **taskParams** – Change Kaspersky Administration Server task parameters

```
{"pData" : {
    "PRTS_TASK_ENABLED": true,
```

```
        "TASKID_PRODUCT_NAME": "1103",
        "TASK_ADDITIONAL_PARAMS": {
            "type": "params",
            "value": {
            "ServerSslPorts": [ 13000 ],
            "Serverports": [ 14000 ],
            "ServerAddress": "NewKscServerAddress"
        }
    },
        "TASK_INFO_PARAMS": {
            "type": "params",
            "value": {
            "DisplayName": "Reconnect to vKSC",
            "HostList": [ {
                    "type": "params",
                    "value": {
                        "HostDispName": "Host1",
                        "HostName": "7ad995e2-eb62-40e5-9c7e-
5abae19979a0"
                    }
                }
            ]
        }
    },
    "TASKID_VERSION": "1.0.0.0",
    "TASKSCH_TYPE": 0,
    "TASK_NAME": "KLNAG_TASK_RECONNECTION",
    "TASKID_COMPONENT_NAME": "86"

} }
```

where:

- **ServerSslPorts** – an array of SSL ports of the new Administration Server. By default, it is just one port **13000**.
- **Serverports** – an array of ports of the new Administration Server. By default, it is just one port **14000**.
- **ServerAddress** – IP address in the IPv4 format or full domain name (FQDN) of the new Administration Server. If the virtual machine is controlled by a virtual Administration Server, the value must have the following format: **MainKscServerAddress/VirtualServerName**.
- **DisplayName** – task display name.
- **HostDispName** – name of the virtual machine to be moved to a new Administration Server. **KLHST_WKS_DN** value is received by calling the HostGroup.FindHosts method.
- **HostName** – unique ID of the virtual machine to be moved to a new Administration Server. **KLHST_WKS_HOSTNAME** value is received by calling the HostGroup.FindHosts method.

If successful, the method returns the ID of the created task in the format:

```
{"PxgRetVal" : "taskId"}
```

Creating the remote installation task

```
POST https://MainKscServerIpAddress:Port/api/v1.0/Tasks.AddTask
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength
{"pData" : taskParams}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.

- **Port** – the Kaspersky Security Center OpenAPI port on the main Administration Server. Port **13299** is used by default.
- **Base64VirtualKscName** – user name of the main Administration Server in the Base64 format. The `X-KSC-VServer` header is required for method queries for a virtual Administration Server.
- **BodyLength** – length of the JSON body of the request in bytes.
- **taskParams** – parameters of the remote application installation task:

```
{"pData" : {
    "PRTS_TASK_ENABLED": true,
    "TASKID_PRODUCT_NAME": "1093",
    "TASK_ADDITIONAL_PARAMS": {
        "type": "params",
        "value": {
            "KLTSK_RI_USE_SHARE": true,
            "KLTSK_RI_USE_SHARE_SRV": true,
            "KLTSK_RI_PACKAGES_IDS": [
                    KlNagentPackageId,
                    ProductPackageId
            ],
            "KLTSK_RI_USE_NAGENT": true,
            "KLTSK_RI_GROUP_TO_MOVE_HOST": GroupToMoveHostId,
            "klprts-TaskAccounts": [
            {
                "type": "params",
                "value": {
                    "klprts-TaskAccountAuthType": 1
                }
            },
            {
                "type": "params",
                "value": {
                    "klprts-TaskAccountUser": "HostOsUserLogin"
                }
            },
            {
                "type": "params",
                "value": {
                    "klprts-TaskAccountPassword": "HostOsUserPassword"
                }
            }
        }
        ]
    }
    },
    "TASK_INFO_PARAMS": {
            "type": "params",
            "value": {
            "DisplayName": "Install KSVLA 5.1.0.0 on host",
            "HostList": [
                    "type": "params",
                    "value": {
                        "HostDispName": "Host1",
                        "HostName": "7ad995e2-eb62-40e5-9c7e-
5abae19979a0"
                    }
        ]
        }
    },
"TASKID_VERSION": "1.0.0.0",
"TASKSCH_TYPE": 0,
```

```
"TASK_NAME": "Remote Installation",
"TASKID_COMPONENT_NAME": "87"
} }
```

where:

- **KlNagentPackageId** – ID of the Network Agent installation package found by calling the [PackagesApi.GetPackages](#) method. The package must be located on the Administration Server which controls the virtual machine. To send the packages to the Administration Server, use the [PackagesApi.RetranslateToVServerAsync](#) method.
- **ProductPackageId** – ID of the Kaspersky Security for Virtualization 5.1 Light Agent installation package found by calling the [PackagesApi.GetPackages](#) method. The package must be located on the Administration Server which controls the virtual machine. To send the packages, use the [PackagesApi.RetranslateToVServerAsync](#) method.
- **GroupToMoveHostId** – id of the administration group to which the virtual machine will be moved after application installation. To search for a group, use the [HostGroup.FindGroups](#) or [HostGroups.GroupIdGroups](#) method.
- **HostOsUserLogin** – user name for the account which the application will be installed.
- **HostOsUserPassword** – password for the user account under which the application will be installed.
- **DisplayName** – task display name.
- **HostDispName –** name of a virtual machine on which the application should be installed. **KLHST_WKS_DN** value is received by calling the [HostGroup.FindHosts](#) method.
- **HostName** – unique ID of the virtual machine to which the application should be installed. **KLHST_WKS_HOSTNAME** value is received by calling the [HostGroup.FindHosts](#) method.

If successful, the method returns the ID of the created task in the format:

```
{"PxgRetVal" : "taskId"}
```

Creating the remote application uninstallation task

```
POST https://MainKscServerIpAddress:Port/api/v1.0/Tasks.AddTask
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength
{"pData" : taskParams}
```

where:

- **MainKscServerIpAddress** – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.
- **Port** – the OpenAPI Kaspersky Security Center port on the main Administration Server. Port **13299** is used by default.
- **Base64VirtualKscName** – user name of the main Administration Server in the Base64 format. The `X-KSC-VServer` header is required for method queries for a virtual server.
- **BodyLength** – length of the JSON body of the request in bytes.
- **taskParams** – parameters of the remote application uninstallation task:

```
{"pData" : {
        "PRTS_TASK_ENABLED": true,
        "TASKID_PRODUCT_NAME": "1093",
        "TASK_ADDITIONAL_PARAMS": {
                "type": "params",
                "value": {
                        "klprts-TaskAccounts": [
                         {
                                "type": "params",
                                "value": {
                                        "klprts-TaskAccountAuthType": 1
                                }
```

```
                    },
                    {
                          "type": "params",
                          "value": {
                                "klprts-TaskAccountUser":
  "HostOsUserLogin"
                          }
                    },
                    {
                          "type": "params",
                          "value": {
                                "klprts-TaskAccountPassword":
  "HostOsUserPassword"
                          }
                    }
                ],
            "KLTSK_RI_USE_SHARE_SRV": true,
            "KLTSK_RI_USE_SHARE": true,
            "ProductVersion": "5.1.0.0",
            "KLTSK_RI_USE_NAGENT": true,
            "ProductName": "KSVLA",
            "UninstallType": 0
        }
    },
    "TASK_INFO_PARAMS": {
            "type": "params",
            "value": {
             "DisplayName": "Deinstall KSVLA 5.1.0.0 on host",
             "HostList": [

                    "HostDispName": "Host1",
                    "HostName": "7ad995e2-eb62-40e5-9c7e-
  5abae19979a0"

                ]
            }
    },
    "TASKID_VERSION": "1.0.0.0",
    "TASKSCH_TYPE": 0,
    "TASK_NAME": "Remote Deinstallation",
    "TASKID_COMPONENT_NAME": "87"
    } }
```

where:

- **HostOsUserLogin** – user name for the account which the application will be uninstalled.
- **HostOsUserPassword** – password for the user account under which the application will be uninstalled.
- **ProductVersion –** name of the application to uninstall.
- **ProductName –** version of the application to uninstall.
- **DisplayName** – task display name.
- **HostDispName –** name of a virtual machine on which the application should be removed. **KLHST_WKS_DN** value is received by calling the [HostGroup.FindHosts](#) method.
- **HostName** – unique ID of the virtual machine from which the application should be removed. **KLHST_WKS_HOSTNAME** value is received by calling the [HostGroup.FindHosts](#) method.

If successful, the method returns the ID of the created task in the format:

```
{"PxgRetVal" : "taskId"}
```

Running a task

```
POST https://MainKscServerIpAddress:Port/api/v1.0/Tasks.RunTask
X-KSC-VServer: Base64VirtualKscName
Content-Type: application/json
Content-Length: BodyLength
{"strTask" : "taskId"}
```

where:

MainKscServerIpAddress – IP address in the IPv4 format or full domain name (FQDN) of the main Kaspersky Security Center Administration Server.

Port – the Kaspersky Security Center OpenAPI port on the main Administration Server. Port **13299** is used by default.

Base64VirtualKscName – user name of the main Administration Server in the Base64 format. The X-KSC-VServer header is required for method queries for a virtual Administration Server.

BodyLength – length of the JSON body of the request in bytes.

taskId – task ID assigned once it is created.