BeyondTrust

Privileged Remote Access 20.2
API Programmer's Guide

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

TC:2/4/2021

Table of Contents

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

BeyondTrust Privileged Remote Access API Programmer's Guide

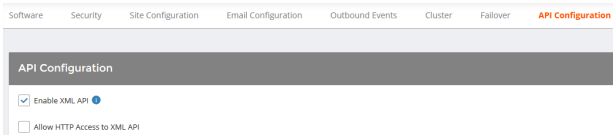
Version 1.19.2 (for BeyondTrust PRA 20.2.x)

Front-end integration of the BeyondTrust API enables customers to correlate BeyondTrust sessions with third-party or in-house developed applications to pull report data, issue commands, or automatically save a backup of the BeyondTrust Secure Remote Access Appliance's software configuration on a recurring basis.

One common example of API integration would be linking a customer relationship management ticketing system to BeyondTrust sessions.

You could also add a feature to an application to enable the user to start a session from directly within that program instead of the BeyondTrust access console.

To use the BeyondTrust API, ensure that the **Enable XML API** option is checked on the **Management > API Configuration** page of the **/login** administrative interface.



For the examples in the following pages, a sample URL of **access.example.com** is used. Please replace this URL with your BeyondTrust Secure Remote Access Appliance's public site URL.

The command and reporting APIs return XML responses that declare a namespace. If you are parsing these responses with a namespace-aware parser, you will need to set the namespace appropriately or ignore the namespace while parsing the XML.

- Reporting API: https://www.beyondtrust.com/namespaces/API/reporting
- Command API: https://www.beyondtrust.com/namespaces/API/command

Note: *The above namespaces are returned XML data and are not functional URLs.*

Note: *Prior to 16.1, a user account was used to authenticate to the API, with the username and password being passed in the request. Starting with 16.1, this method has been deprecated and is not available to new users. Instead, one or more API accounts must be created, with their client IDs and client secrets used to generate OAuth tokens.*

For users upgrading from a version prior to 16.1, the option to authenticate to the API with a user account is still available for backwards compatibility. However, it is highly recommended that you use the more secure OAuth method of authentication. If you are unable to switch to OAuth authentication, please follow the API request format described in our documentation archive at www.beyondtrust.com/docs/archive/privileged-remote-access.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Authenticate to the Privileged Remote Access API

API requests are executed by sending an HTTP request to the appliance. Send the request using any HTTPS-capable socket library or scripting language module, URL fetcher such as cURL, or an OAuth library specific to your platform. BeyondTrust's web APIs use OAuth as the authentication method.

To authenticate to the API, you must create an API account on the **/login > Management > API Configuration** page (see www.beyondtrust.com/docs/privileged-remote-access/getting-started/admin/api-configuration.htm). The account must have permission to access the necessary APIs. API requests require a token to be first created and then submitted with each API request. An example API request can be seen in the "Test Scenario" on page 64.

Create a Token

Create a token by POSTing to the URL of your BeyondTrust site followed by `/oauth2/token`:

```
https://access.example.com/oauth2/token
```

The OAuth client ID and client secret associated with the API account should be base64 encoded and included in an HTTP basic authorization header:

```
Authorization: Basic <base64-encoded "client_id:secret">
```

The request should include the following POST body:

```
grant_type=client_credentials
```

If the request is processed without error, you will get an access token JSON response:

```
{
      "access_token":"<token>"
      "token_type":"Bearer"
      "expires_in":3600
}
```

***Note:** This token expires after one hour. Any calls to the API past that point must have a new token. Each API account can have a maximum of 30 valid tokens. If an API account attempts to generate more than 30 tokens, then the oldest token is invalidated before a new one is generated.*

***Note:** The client secret cannot be modified, but it can be regenerated on the **/login > Management > API Configuration** page. Regenerating a client secret and then saving the account immediately invalidates any OAuth tokens associated with the account. Any API calls using those tokens will be unable to access the API. A new token must be generated using the new client secret.*

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Request an API Resource

Now that you have an access token, you can make GET/POST requests via HTTPS to the web API:

```
https://access.example.com/api/command
```

The obtained token is used for HTTP authentication and must be included in an HTTP authorization header with each request:

```
Authorization: Bearer <token>
```

If the token is valid, you gain access to the requested URL.

Authentication Errors

Requests made to the web API with expired or invalid tokens result in a JSON error response:

```
{
      "error":"access_denied"
      "message":"The resource owner or authorization server denied the request."
}
```

IMPORTANT!

When making consecutive API calls, you must close the connection after each API call.

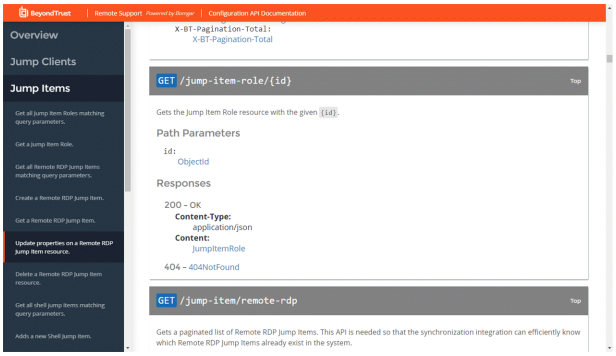
©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Configuration API

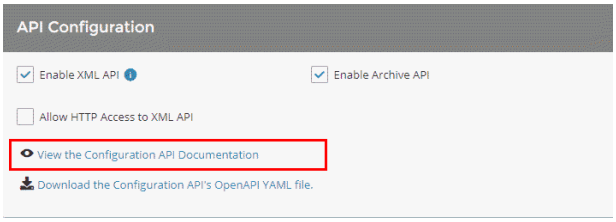
The Configuration API is written according to OpenAPI standards, and enables end users to view documentation for the API using their preferred OpenAPI tool, such as Swagger, Postman, or RediDoc. You can either view the Configuration API documentation directly in the product (/login), or download the YAML file and use a tool of your choice to view the documentation.

View the Configuration API Documentation in /login

Log into your site (for example, **https://example.com/login/apidocs.html**) and enter your credentials. You can find lists, descriptions, and examples for all available APIs.



You can click the link to view the in-product Config API documentation.



Access the YAML file via API

By following the steps below and referring to the documentation for the OpenAPI tool of your choice, you can view the API documentation and even *try out* features of the API using an intuitive browser user interface.

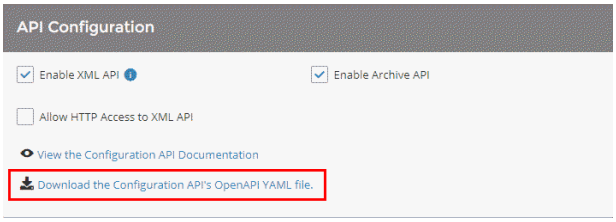
1. Go to **/login > Management > API Configuration**.
2. Under **API Accounts**, click **Add**.
3. Enter a name to identify your new API account.
4. Make sure the **Configuration API > Allow Access** box is checked.
5. Click **Save**.
6. Download and install your favorite software for running API calls. Please refer to the documentation for your selected software before proceeding, if needed.
7. In **/login > Management > API Configuration**, select the new API account you just created and click the edit icon.
8. Copy the **OAuth Client ID** and paste it into your selected software.
9. Back in **/login**, click **Generate New Client Secret**, copy it, and paste it into you selected software.
10. Click **Save** to save your API account.
11. Import the **OpenAPI.yaml** file from your site, using your preferred OpenAPI tool. The **OpenAPI.yaml** file can be accessed by creating a new **GET** request with the URL format https://example.com/api/config/v1/openapi.yaml. Once imported, the

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

documentation for the Configuration APIs will be automatically generated. Follow the instructions in your API call software in order to complete these steps.

Download the YAML file

Alternatively, you can download the YAML file by clicking the **Download the Configuration API's OpenAPI YAML file**



©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Command API

The command API is designed to send commands to your BeyondTrust site from an outside application. Commands can get or set session attributes, join an existing session, or terminate a session. You can also check the health of your appliance or get information about your BeyondTrust API version.

The command API is an authenticated API. For instructions on using authenticated APIs using OAuth, please see "Authenticate to the Privileged Remote Access API" on page 5.

Commands are executed by sending an HTTP request to the appliance. Send the request using any HTTPS-capable socket library, scripting language module, or URL fetcher such as **cURL** or **wget**. Use either **GET** or **POST** as the request method.

POST requests must include a "Content-Type: application/x-www-form-urlencoded" HTTP header when supplying parameters in the request body, and the parameters must be url-encoded. Multipart POST requests are not supported.

IMPORTANT!

When making consecutive API calls, you must close the connection after each API call.

The command API URL is **https://access.example.com/api/command**.

An XML schema describing the command API response format is available at **https://access.example.com/api/command.xsd**.

Required Parameter for Command API

action=[string]	The type of action to perform. Can be **join_session**, **set_session_attributes**, **get_session_attributes**, **import_jump_shortcut**, **terminate_session**, **check_health**, **set_failover_role**, or **get_api_info**.

The command API returns XML responses that declare a namespace. If you are parsing these responses with a namespace-aware parser, you need to set the namespace appropriately or ignore the namespace while parsing the XML.

- Command API: https://www.beyondtrust.com/namespaces/API/command

Note: *The above namespace is returned XML data and is not a functional URL.*

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Command: get_logged_in_reps

The **get_logged_in_reps** request returns XML data about all logged-in representatives. It requires no additional parameters.

> The command API is an authenticated API. For instructions on using authenticated APIs using OAuth, please see "Authenticate to the Privileged Remote Access API" on page 5. The API account must have read-only or full access to the command API.

XML Response for get_logged_in_reps Query

<logged_in_reps>	Returns a **<rep>** element for each logged-in representative. If no representatives are logged in, this element will contain no **<rep>** elements. If an error occurs, it will contain an **<error>** element describing the problem.

Element Names and Attributes

/logged_in_reps/rep

id (attribute)	Unique ID assigned to the representative.
<display_name>	This element is deprecated as of API version 1.10.0 but still exists for backwards compatibility. Its value is the same as that of **<public_display_name>**.
<public_display_name>	The public display name currently assigned to the representative.
<private_display_name>	The private display name currently assigned to the representative.
<type>	The type of rep logged in. Types include **Normal** and **Invited**.
<direct_link>	An HTML anchor tag containing the URL that customers can use to download the customer client to connect directly to the representative.
<logged_in_since>	The date and time at which the representative logged in.
<presentation_count>	The number of active presentations the representative is currently running.
<support_session_count>	The number of active sessions the representative is currently running.
<showing_on_rep_list>	Integer value (**1** or **0**) indicating if the rep has permission to show on the public site and has the **Showing On Representative List** option checked in the [[[Undefined variable ProductNames.Console]]].

Query Example: get_logged_in_reps

get_logged_in_reps	https://access.example.com/api/command?action=get_logged_in_reps

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

IMPORTANT!

*If you experience a high volume of support requests, repeatedly calling a command such as **get_logged_in_reps** might bottleneck your system. Therefore, a best practice is to not request a list of representatives or teams with each support request. Instead, if making the same API call in succession, consider caching the results for a period of time and reusing them. New sessions requests should reference the cached list instead of calling for the list each time.*

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Command: set_session_attributes

The **set_session_attributes** command sets the external key and other custom attributes for an active session.

The API account used to issue this command must have full access to the command API.

Required Parameter for set_session_attributes

lsid=[string]	The ID of the session whose attributes you wish to set. The session must currently be active.

Optional Parameters for set_session_attributes

session.custom.external_key=[string]	An arbitrary string that can link this session to an identifier on an external system, such as a customer relationship management ticket ID. This has a maximum length of 1024 characters.
session.custom.[custom field]=[string]	The code name and value of any custom fields. These fields must first be configured in **/login > Management > API Configuration**. Each attribute must be specified as a different parameter. Each custom field has a maximum length of 1024 characters. The maximum total size of all combined custom fields, including the external key, must be limited to 10KB.

***Note:** If an attribute is not listed in the URL, it will keep its existing value. To clear an attribute, you must set the attribute to an empty string.*

XML Response for set_session_attributes Query

<success>	Returns a message of **Session attributes were set** if the attributes were set successfully.
<error>	Returns an error message if the attributes were not set successfully.

Query Examples: set_session_attributes

Set external key for session c69a8e10bea9428f816cfababe9815fe	`https://access.example.com/api/command?action=set_session_attributes&lsid=c69a8e10bea9428f816cfababe9815fe&session.custom.external_key=ABC123`
Set a custom value for session c69a8e10bea9428f816cfababe9815fe	`https://access.example.com/api/command?action=set_session_attributes&lsid=c69a8e10bea9428f816cfababe9815fe&session.custom.custom_field1=Custom%20Value`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Command: get_session_attributes

The **get_session_attributes** command returns attributes set for an active session.

In order to issue the **get_session_attributes** command, you must supply the username and password for a BeyondTrust user account. That account must have the permission **Allowed to Use Command API** along with the permission **Administrator**.

The API account used to issue this command must have read-only or full access to the command API.

Required Parameter for get_session_attributes

lsid=[string]	The ID of the session whose attributes you wish to get. The session must currently be active.

XML Response for get_session_attributes Query

<custom_attributes>	Contains a **<custom_attribute>** element for each custom attribute set for the session.
<error>	Returns an error message if the attributes were not retrieved successfully.

Element Names and Attributes

	/custom_attributes/custom_attribute
display_name (attribute)	The display name assigned to the custom attribute.
code_name (attribute)	The code name assigned to the custom attribute.

Query Example: get_session_attributes

Get custom attributes for session c69a8e10bea9428f816cfababe9815fe	`https://access.example.com/api/command?action=get_session_attributes&lsid=c69a8e10bea9428f816cfababe9815fe`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Command: import_jump_shortcut

The **import_jump_shortcut** command creates a Jump shortcut. When dealing with a large number of Jump shortcuts, it may be easier to import them programmatically than to add them one by one in the access console.

The API account used to issue this command must have full access to the command API.

Required Parameters for import_jump_shortcut - Local Jump

name=[string]	The name of the endpoint to be accessed by this Jump Item. This name identifies the item in the session tabs. This string has a maximum of 128 characters.
local_jump_hostname=[string]	The hostname of the endpoint to be accessed by this Jump Item. This string has a maximum of 128 characters.
group=[string]	The code name of the Jump Group with which this Jump Item should be associated.  ***Note:*** *When using the import method, a Jump Item cannot be associated with a personal list of Jump Items.*

Optional Parameters for import_jump_shortcut - Local Jump

tag=[string]	You can organize your Jump Items into categories by adding a tag. This string has a maximum of 1024 characters.
comments=[string]	You can add comments to your Jump Items. This string has a maximum of 1024 characters.
jump_policy=[string]	The code name of a Jump Policy. You can specify a Jump Policy to manage access to this Jump Item.
session_policy=[string]	The code name of a session policy. You can specify a session policy to manage the permissions available on this Jump Item.

Required Parameters for import_jump_shortcut - Remote Jump

name=[string]	The name of the endpoint to be accessed by this Jump Item. This name identifies the item in the session tabs. This string has a maximum of 128 characters.
remote_jump_hostname=[string]	The hostname of the endpoint to be accessed by this Jump Item. This string has a maximum of 128 characters.
jumpoint=[string]	The code name of the Jumpoint through which the endpoint is accessed.
group=[string]	The code name of the Jump Group with which this Jump Item should be associated.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

	 ***Note:*** *When using the import method, a Jump Item cannot be associated with a personal list of Jump Items.*

Optional Parameters for import_jump_shortcut - Remote Jump

tag=[string]	You can organize your Jump Items into categories by adding a tag. This string has a maximum of 1024 characters.
comments=[string]	You can add comments to your Jump Items. This string has a maximum of 1024 characters.
jump_policy=[string]	The code name of a Jump Policy. You can specify a Jump Policy to manage access to this Jump Item.
session_policy=[string]	The code name of a session policy. You can specify a session policy to manage the permissions available on this Jump Item.

Required Parameters for import_jump_shortcut - VNC

remote_vnc_hostname=[string]	The hostname of the endpoint to be accessed by this Jump Item. This string has a maximum of 128 characters.
jumpoint=[string]	The code name of the Jumpoint through which the endpoint is accessed.
name=[string]	The name of the endpoint to be accessed by this Jump Item. This name identifies the item in the session tabs. This string has a maximum of 128 characters.
group=[string]	The code name of the Jump Group with which this Jump Item should be associated.  ***Note:*** *When using the import method, a Jump Item cannot be associated with a personal list of Jump Items.*

Optional Parameters for import_jump_shortcut - VNC

port=[integer]	A valid port number from **100** to **65535**. Defaults to **5900**.
tag=[string]	You can organize your Jump Items into categories by adding a tag. This string has a maximum of 1024 characters.
comments=[string]	You can add comments to your Jump Items. This string has a maximum of 1024 characters.
jump_policy=[string]	The code name of a Jump Policy. You can specify a Jump Policy to manage access to this Jump Item.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Required Parameters for import_jump_shortcut - Remote Desktop Protocol

name=[string]	The name of the endpoint to be accessed by this Jump Item. This name identifies the item in the session tabs. This string has a maximum of 128 characters.
remote_rdp_hostname=[string]	The hostname of the endpoint to be accessed by this Jump Item. This string has a maximum of 128 characters.
jumpoint=[string]	The code name of the Jumpoint through which the endpoint is accessed.
group=[string]	The code name of the Jump Group with which this Jump Item should be associated.  ***Note:*** *When using the import method, a Jump Item cannot be associated with a personal list of Jump Items.*

Optional Parameters for import_jump_shortcut - Remote Desktop Protocol

rdp_username=[string]	The username to sign in as.
domain=[string]	The domain the endpoint is on.
display_size=[string]	The resolution at which to view the remote system. Can be **primary** (default - the size of your primary monitor), **all** (the size of all of your monitors combined), or **XxY** (where **X** and **Y** are a supported width and height combination - e.g., **640x480**).
quality=[string]	The quality at which to view the remote system. Can be **low** (2-bit gray scale for the lowest bandwidth consumption), **best_perf** (default - 8-bit color for fast performance), **perf_and_qual** (16-bit for medium quality image and performance), **best_qual** (32-bit for the highest image resolution), or **video_opt** (VP9 codec for more fluid video). This cannot be changed during the remote desktop protocol (RDP) session.
console=[boolean]	**1**: Starts a console session. **0**: Starts a new session (default).
ignore_untrusted=[boolean]	**1**: Ignores certificate warnings. **0**: Shows a warning if the server's certificate cannot be verified.
tag=[string]	You can organize your Jump Items into categories by adding a tag. This string has a maximum of 1024 characters.
comments=[string]	You can add comments to your Jump Items. This string has a maximum of 1024 characters.
jump_policy=[string]	The code name of a Jump Policy. You can specify a Jump Policy to manage access to this Jump Item.
sql_server_hostname=[string]	The hostname of the SQL Server used to access SQL Server Management Studio. This string has a maximum of 64 characters.
sql_server_port=[integer]	The port used to access the SQL Server instance. The port value accepts only

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

	integers in the range of 1-65535, with **1433** as the default value.
sql_server_database=[string]	The database name of the SQL Server instance being accessed.. This string has a maximum of 520 characters.
custom_app_name=[string]	The name of the remote application being accessed. This string has a maximum of 520 characters.
custom_app_params=[string]	A space-separated list of parameters to pass to the remote application. Parameters with spaces can be delimited using double-quotes. This string has a maximum of 16,000 characters.

Required Parameters for import_jump_shortcut - Shell Jump Shortcut

name=[string]	The name of the endpoint to be accessed by this Jump Item. This name identifies the item in the session tabs. This string has a maximum of 128 characters.
shelljump_hostname=[string]	The hostname of the endpoint to be accessed by this Jump Item. This string has a maximum of 128 characters.
jumpoint=[string]	The code name of the Jumpoint through which the endpoint is accessed.
protocol=[string]	Can be either **ssh** or **telnet**.
group=[string]	The code name of the Jump Group with which this Jump Item should be associated. ***Note:*** *When using the import method, a Jump Item cannot be associated with a personal list of Jump Items.*

Optional Parameters for import_jump_shortcut - Shell Jump Shortcut

shelljump_username=[string]	The username to sign in as.
port=[integer]	A valid port number from **1** to **65535**. Defaults to **22** if the protocol is **ssh** or **23** if the protocol is **telnet**.
terminal=[string]	Can be either **xterm** (default) or **VT100**.
keep_alive=[integer]	The number of seconds between each packet sent to keep an idle session from ending. Can be any number from **0** to **300**. **0** disables keep-alive (default).
tag=[string]	You can organize your Jump Items into categories by adding a tag. This string has a maximum of 1024 characters.
comments=[string]	You can add comments to your Jump Items. This string has a maximum of 1024 characters.
jump_policy=[string]	The code name of a Jump Policy. You can specify a Jump Policy to manage access to this Jump Item.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

session_policy=[string]	The code name of a session policy. You can specify a session policy to manage the permissions available on this Jump Item.

Required Parameters for import_jump_shortcut - Protocol Tunnel Jump Shortcut

Field	Description
protocol_tunnel_hostname	The hostname of the endpoint to be accessed by this Jump Item. This string has a maximum of 128 characters.
jumpoint	The code name of the Jumpoint through which the endpoint is accessed.
tcp_tunnels	The list of one or more tunnel definitions. A tunnel definition is a mapping of a TCP port on the local user's system to a TCP port on the remote endpoint. Any connection made to the local port causes a connection to be made to the remote port, allowing data to be tunnelled between local and remote systems. Multiple mappings should be separated by a semicolon. Example: `auto->22;3306->3306` In the example above, a randomly assigned local port maps to remote port 22, and local port 3306 maps to remote port 3306.
name=[string]	The name of the endpoint to be accessed by this Jump Item. This name identifies the item in the session tabs. This string has a maximum of 128 characters.
group	The code name of the Jump Group with which this Jump Item should be associated. ***Note:*** *When using the import method, a Jump Item cannot be associated with a personal list of Jump Items.*

Optional Parameters for import_jump_shortcut - Protocol Tunnel Jump Shortcut

Field	Description
local_address	The address from which the connection should be made. This can be any address within the 127.x.x.x subrange. The default address is 127.0.0.1.
tag	You can organize your Jump Items into categories by adding a tag. This string has a maximum of 1024 characters.
comments	You can add comments to your Jump Items. This string has a maximum of 1024 characters.
jump_policy	The code name of a Jump Policy. You can specify a Jump Policy to manage access to this Jump Item.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Required Parameters for import_jump_shortcut - Web Jump Shortcut

Field	Description
web site_name	The name of the endpoint to be accessed by this Jump Item. This name identifies the item in the session tabs. This string has a maximum of 128 characters.
jumpoint	The code name of the Jumpoint through which the endpoint is accessed.
url	The URL of the web site. The URL must begin with either **http** or **https**.
group	The code name of the Jump Group with which this Jump Item should be associated.  ***Note:*** *When using the import method, a Jump Item cannot be associated with a personal list of Jump Items.*

Optional Parameters for import_jump_shortcut - Web Jump Shortcut

Field	Description
verify_certificate	**1**: The site certificate is validated before the session starts; if issues are found, the session will not start. **0**: The site certificate is not validated.
tag	You can organize your Jump Items into categories by adding a tag. This string has a maximum of 1024 characters.
comments	You can add comments to your Jump Items. This string has a maximum of 1024 characters.
jump_policy	The code name of a Jump Policy. You can specify a Jump Policy to manage access to this Jump Item.
session_policy	The code name of a session policy. You can specify a session policy to manage the permissions available on this Jump Item.

XML Response for import_jump_shortcut Query

<success>	Returns a message of **Successfully imported Jump Item shortcut** if the import succeeded.
<error>	Returns an error message if the import failed.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Query Examples: import_jump_shortcut

Import Local Jump shortcut "Endpoint" to the endpoint with hostname "ABCDEF02", pinning it to Jump Group "remote_access"	`https://access.example.com/api/command?action=import_jump_shortcut&name=Endpoint&local_jump_hostname=ABCDEF02&group=remote_access`
Import Local Jump shortcut "Endpoint" to the endpoint with hostname "ABCDEF02", pinning it to Jump Group "remote_access" and specifying its tag, comments, Jump Policy, and session policy	`https://access.example.com/api/command?action=import_jump_shortcut&name=Endpoint&local_jump_hostname=ABCDEF02&group=remote_access&tag=Frequent%20Access&comments=Web%20server&jump_policy=Notify&session_policy=Servers`
Import Remote Jump shortcut "Endpoint" to the endpoint with hostname "ABCDEF02", accessed through Jumpoint "London", pinning it to Jump Group "remote_access"	`https://access.example.com/api/command?action=import_jump_shortcut&name=Endpoint&remote_jump_hostname=ABCDEF02&jumpoint=London&group=remote_access`
Import VNC shortcut "Endpoint" to the endpoint with hostname "ABCDEF02", accessed through Jumpoint "London", pinning it to Jump Group "remote_access"	`https://access.example.com/api/command?action=import_jump_shortcut&name=Endpoint&remote_vnc_hostname=ABCDEF02&jumpoint=London&group=remote_access`
Import VNC shortcut "Endpoint" to the endpoint with hostname "ABCDEF02", accessed through Jumpoint "London", pinning it to Jump Group "remote_access" and specifying its port	`https://access.example.com/api/command?action=import_jump_shortcut&name=Endpoint&remote_vnc_hostname=ABCDEF02&jumpoint=London&group=remote_access&port=100`
Import RDP shortcut "Endpoint" to the endpoint with hostname "ABCDEF02", accessed through Jumpoint "London", pinning it to Jump Group "remote_access"	`https://access.example.com/api/command?action=import_jump_shortcut&name=Endpoint&remote_rdp_hostname=ABCDEF02&jumpoint=London&group=remote_access`
Import RDP shortcut "Endpoint" to the endpoint with hostname "ABCDEF02", accessed through Jumpoint "London", pinning it to Jump Group "remote_access" and specifying its username, domain, display size, quality, console session, untrusted certificate action, sql server name, sql server port, sql server database name, remote app name, and remote app parameters	`https://access.example.com/api/command?action=import_jump_shortcut&name=Endpoint&remote_rdp_hostname=ABCDEF02&jumpoint=London&group=remote_access&rdp_username=admin&domain=example&display_size=1280x720&quality=perf_and_qual&console=1&ignore_untrusted=1&sql_server_hostname=example.local&sql_server_port=1500&sql_server_database=example&custom_app_name=sql_server&custom_app_params=x,y,z`
Import Shell Jump shortcut "Endpoint" to the endpoint with hostname "ABCDEF02", accessed through Jumpoint "London" over SSH, pinning it to Jump Group "remote_access"	`https://access.example.com/api/command?action=import_jump_shortcut&name=Endpoint&shelljump_hostname=ABCDEF02&jumpoint=London&protocol=ssh&group=remote_access`
Import Shell Jump shortcut "Endpoint" to the endpoint with hostname "ABCDEF02",	`https://access.example.com/api/command?action=import_jump_shortcut&name=Endpoint&shelljump_hostname=`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

accessed through Jumpoint "London" over SSH, pinning it to Jump Group "remote_access", and specifying its username, port, terminal type, and keep-alive settings	`ABCDEF02&jumpoint=London&protocol=ssh&group=remote_access&shelljump_username=admin&port=25&terminal=vt100&keep_alive=120`
Import Protocol Tunnel Jump shortcut "Endpoint" to the endpoint with hostname "ABCDEF02", accessed through Jumpoint "London", pinning it to Jump Group "remote_access", with a randomly assigned local port mapping to remote port 22	`https://access.example.com/api/command?action=import_jump_shortcut&name=Endpoint&protocol_tunnel_hostname=ABCDEF02&jumpoint=London&group=remote_access&tcp_tunnels=auto->22`
Import Protocol Tunnel Jump shortcut "Endpoint" to the endpoint with hostname "ABCDEF02", accessed through Jumpoint "London", pinning it to Jump Group "remote_access", with a randomly assigned local port mapping to remote port 22, local port 3306 mapping to port 3306, and a local address of 127.0.0.5	`https://access.example.com/api/command?action=import_jump_shortcut&name=Endpoint&protocol_tunnel_hostname=ABCDEF02&jumpoint=London&group=remote_access&tcp_tunnels=auto->22;3306->3306&local_address=127.0.0.5`
Import Web Jump shortcut "Endpoint" to the endpoint with URL "example.com", accessed through Jumpoint "London", pinning it to Jump Group "remote_access"	`https://access.example.com/api/command?action=import_jump_shortcut&web site_name=Endpoint&url=example.com&jumpoint=London&group=remote_access`
Import Web Jump shortcut "Endpoint" to the endpoint with URL "example.com", accessed through Jumpoint "London", pinning it to Jump Group "remote_access" and not requiring certificate validation	`https://access.example.com/api/command?action=import_jump_shortcut&web site_name=Endpoint&url=example.com&jumpoint=London&group=remote_access&verify_certificate=0`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Command: terminate_session

The **terminate_session** command terminates a session that is in progress.

The API account used to issue this command must have full access to the command API.

Required Parameter for terminate_session

lsid=[string]	The unique ID representing the session you wish to terminate.

XML Response for terminate_session Query

<success>	Returns a message of **Successfully terminated** if the termination was successful.
<error>	Returns an error message if the termination was not successful.

Query Examples: terminate_session

Session da4b510978a541d49398e88c66e28475 terminated	`https://access.example.com/api/command?action=terminate_session&lsid=da4b510978a541d49398e88c66e28475`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Command: get_connected_client_list

The **get_connected_client_list** command returns XML data containing a summary or list of all connected BeyondTrust clients.

> The command API is an authenticated API. For instructions on using authenticated APIs using OAuth, please see "Authenticate to the Privileged Remote Access API" on page 5. The API account must have read-only or full access to the command API.

Optional Parameters for get_connected_client_list

type=[string]	The types of clients to return in the results. Can be a comma-separated list of values. Supported values are **all** (default), **representative**, **support_customer**, **presentation_attendee**, and **push_agent**. ***Note:** Currently, **pinned_client** is not a possible value. If the count of pinned Jump Clients is needed in the summary, then **all** must be specified.*
summary_only=[boolean]	To return only a summary, set this to **1**.

XML Response for get_connected_client_list

<connected_client_list>	Contains a **<connected_client_summary>** element with a summary of the data. Also contains a **<connected_client>** element for each client currently connected to the appliance. If an error occurs, it will contain an **<error>** element describing the problem.

Element Names and Attributes

/connected_client_list/connected_client_summary

<appliance_summary>	An **<appliance_summary>** element is created for each connected appliance.

/connected_client_list/connected_client_summary/appliance_summary

id (attribute)	The appliance's GUID.
<count>	A **<count>** element is created for each type of client connected to this appliance.

/connected_client_list/connected_client_summary/appliance_summary/count

type (attribute)	The type of client connected to the appliance. Can be one of **representative**, **support_customer**, **presentation_attendee**, **push_agent**, or **pinned_client**.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

/connected_client_list/connected_client

type (attribute)	The type of client connected to one of the clustered appliances. Can be one of **representative**, **support_customer**, **presentation_attendee**, or **push_agent**.
id (attribute)	A unique identifier which remains valid only while the client is connected.

Query Examples: get_connected_client_list

Get a list of all connected clients	https://access.example.com/api/command? action=get_connected_client_list
Get a list of all connected representatives	https://access.example.com/api/command? action=get_connected_client_list&type=representative
Get a list of all connected representatives and support customers	https://access.example.com/api/command? action=get_connected_client_list& type=representative,support_customer
Get a summary of all connected clients	https://access.example.com/api/command? action=get_connected_client_list&summary_only=1
Get a summary of all connected representatives	https://access.example.com/api/command? action=get_connected_client_list&summary_only=1& type=representative
Get a summary of all connected representatives and support customers	https://access.example.com/api/command? action=get_connected_client_list&summary_only=1& type=representative,support_customer

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Command: get_connected_clients

The **get_connected_clients** command returns XML data containing details of all connected BeyondTrust clients.

> The command API is an authenticated API. For instructions on using authenticated APIs using OAuth, please see "Authenticate to the Privileged Remote Access API" on page 5. The API account must have read-only or full access to the command API.

Required Parameters for get_connected_clients

type=[string]	The types of clients to return in the results. Can be a comma-separated list of values. Supported values are **all** (default), **representative**, **support_customer**, **presentation_attendee**, and **push_agent**.
id=[string]	The ID of the client. To get client IDs, see "API Command: get_connected_client_list" on page 23. Can be a comma-separated list of values. A maximum of 100 IDs is supported. This ID is a unique identifier which remains valid only while the client is connected.
include_connections=[boolean]	If this is set to **1**, then the client's list of connections to [[[Undefined variable RS.Appliance]]]s and an event log about those connections will be included in the results.

XML Response for get_connected_clients

<connected_clients>	Contains a child element for each connected client, including **<connected_representative>**, **<connected_support_customer>**, **<connected_presentation_attendee>**, and **<connected_push_agent>**.

Element Names and Attributes

/connected_clients/connected_representative

id (attribute)	A unique identifier which remains valid only while the client is connected.
<client_connections>	Contains a **<client_connections>** element and an **<event_log>** element. This element is returned only if the query specifies **include_connections**.
<hostname>	The hostname of the representative's computer.
<platform>	The operating system of the representative's computer. Also contains an **id** attribute that briefly notes the selected platform for the client.
<timezone_offset>	The number of seconds away from UTC.
<connected_since>	The date and time at which this connection was made. Data is returned in ISO 8601 format. Also contains a **ts** attribute which displays the connection start time as a

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

	UNIX timestamp (UTC). This element is returned only if the query specifies **include_connections**.
<user_id>	Unique ID assigned to the representative.
<type>	The type of account the representative is using. Can be one of **Normal** or **Invited**.
<username>	The username assigned to the representative.
<public_display_name>	The public display name assigned to the representative. Note that this field contains the public display name's value at the time of the conference, which may not match the current value if the **public_display_name** has subsequently been changed.
<private_display_name>	The private display name assigned to the representative. Note that this field contains the private display name's value at the time of the conference, which may not match the current value if the **private_display_name** has subsequently been changed.
<start_session_url>	A URL that can be sent to a customer to start a support session with the representative.
<presentation_count>	The number of presentations the representative is performing. Can be either **0** or **1**.
<support_session_count>	The number of sessions the representative is participating in.
<showing_on_rep_list>	Integer value (**1** or **0**) indicating if the representative appears in the representative list on the public site.
<routing_idle>	Integer value (**1** or **0**) indicating if the representative has a status of idle.
<routing_busy>	Integer value (**1** or **0**) indicating if the representative has a status of busy.
<routing_enabled>	Integer value (**1** or **0**) indicating if the representative has automatic session assignment enabled or disabled.
<routing_available>	Integer value (**1** or **0**) indicating if the representative is available to have sessions automatically assigned.
<support_license>	The type of license used by the representative.
<support_session_lsids>	Contains an **<lsid>** element for each session in which the representative is participating. This field corresponds with the **<lsid>** field of the **<connected_support_customer>** element.

/connected_clients/connected_support_customer

id (attribute)	A unique identifier which remains valid only while the client is connected.
<client_connections>	Contains a **<client_connections>** element and an **<event_log>** element. This element is returned only if the query specifies **include_connections**.
<hostname>	The hostname of the customer's computer.
<platform>	The operating system of the customer's computer. Also contains an id attribute that briefly notes the selected platform for the client.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

<timezone_offset>	The number of seconds away from UTC.
<connected_since>	The date and time at which this connection was made. Data is returned in ISO 8601 format. Also contains a **ts** attribute which displays the connection start time as a UNIX timestamp (UTC). This element is returned only if the query specifies **include_connections**.
<name>	The name which the customer entered in the **Your Name** field of the front-end survey or which was assigned programmatically.
<non_interactive>	Indicates if the session is a remote desktop protocol (RDP) session or a Shell Jump session. Can be either **rdp** or **shelljump**. If neither, this element is not returned.
<lsid>	A string which uniquely identifies this session. This field corresponds with the **<lsid>** field of the **<connected_representative>** element.

/connected_clients/connected_presentation_attendee

id (attribute)	A unique identifier which remains valid only while the client is connected.
<client_connections>	Contains a **<client_connections>** element and an **<event_log>** element. This element is returned only if the query specifies **include_connections**.
<hostname>	The hostname of the attendee's computer.
<platform>	The operating system of the attendee's computer. Also contains an id attribute that briefly notes the selected platform for the client.
<timezone_offset>	The number of seconds away from UTC.
<connected_since>	The date and time at which this connection was made. Data is returned in ISO 8601 format. Also contains a **ts** attribute which displays the connection start time as a UNIX timestamp (UTC). This element is returned only if the query specifies **include_connections**.
<name>	The name which the attendee entered when joining the presentation or which was assigned programmatically.

/connected_clients/connected_push_agent

id (attribute)	A unique identifier which remains valid only while the client is connected.
<client_connections>	Contains a **<client_connection>** element and an **<event_log>** element. This element is returned only if the query specifies **include_connections**.
<hostname>	The hostname of the Jumpoint's host computer.
<platform>	The operating system of the Jumpoint's host computer. Also contains an id attribute that briefly notes the selected platform for the client.
<timezone_offset>	The number of seconds away from UTC.
<connected_since>	The date and time at which this connection was made. Data is returned in ISO 8601 format. Also contains a **ts** attribute which displays the connection start time as a UNIX timestamp (UTC). This element is returned only if the query specifies **include_connections**.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

<name>	The Jumpoint's name.

/client_connection

<appliance_id>	The GUID of the appliance to which the client is connected.
<purpose>	The reason the representative is connected to this appliance. Can be either **master** or **traffic**. If not part of a cluster, this will always be **master**.
<receive_traffic_node>	Integer value (**1** or **0**) indicating whether this is the client's default traffic node or not. If not part of a cluster, this will always be **0**.
<connected_since>	The date and time at which the client connected. Data is returned in ISO 8601 format. Also contains a **ts** attribute which displays the connection start time as a UNIX timestamp (UTC).
<private_ip>	The client's private IP address that was used to connect to the appliance.

/event_log

<event>	An **<event>** element is created for each event that took place during this connection. Up to the last 20 events are returned. Events detail when and why a client connected to an appliance. Events also include failures to connect to nodes and normal disconnects. Includes a **ts** attribute which displays the timestamp of the event.

Query Examples: get_connected_clients

Get a detailed list of all connected clients	https://access.example.com/api/command? action=get_connected_clients
Get a detailed list of all connected representatives	https://access.example.com/api/command? action=get_connected_clients&type=representative
Get a detailed list of all connected representatives and support customers	https://access.example.com/api/command? action=get_connected_clients& type=representative,support_customer
Get a detailed list of all clients with IDs 101, 102, and 103	https://access.example.com/api/command? action=get_connected_clients&id=101,102,103
Get a detailed list of all clients with IDs 101, 102, and 103 AND whose type is representative or customer	https://access.example.com/api/command? action=get_connected_clients&id=101,102,103& type=representative,support_customer
Get a detailed list, with connection information, of all connected clients	https://access.example.com/api/command? action=get_connected_clients&include_connections=1
Get a detailed list, with connection information, of all connected representatives	https://access.example.com/api/command? action=get_connected_clients&type=representative& include_connections=1

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Get a detailed list, with connection information, of all connected representatives and support customers	https://access.example.com/api/command? action=get_connected_clients& type=representative,support_customer&include_connections=1
Get a detailed list, with connection information, of all clients with IDs 101, 102, and 103	https://access.example.com/api/command? action=get_connected_clients&id=101,102,103& include_connections=1
Get a detailed list, with connection information, of all clients with IDs 101, 102, and 103 AND whose type is representative or customer	https://access.example.com/api/command? action=get_connected_clients&id=101,102,103& type=representative,support_customer&include_connections=1

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Command: check_health

The **check_health** command returns XML data containing information about the Secure Remote Access Appliance.

The API account used to issue this command must have read-only or full access to the command API.

XML Response for check_health Query

<appliance>	The hostname of the appliance. Also contains an id attribute that contains the appliance's GUID.
<version>	The version number and build number of the BeyondTrust software running on the appliance.
<success>	Integer value (**1** or **0**) indicating if the health check of the appliance was successful.
<error_message>	Returns an error message if a problem is found. If no error is found, this element will not be returned.
<failover_role>	The role the appliance plays in the failover relationship. Can be one of **none** (if failover is not configured), **primary**, or **backup**.
<enabled_shared_ips>	Contains an **<ip>** element for each IP address which is shared between the primary and backup appliances. If no shared IP addresses are enabled or if failover is not configured, this element is not returned.
<last_data_sync_time>	The date and time at which the last data sync occurred between the primary and backup appliances. Data is returned in ISO 8601 format. Also contains a **ts** attribute which displays the data sync time as a UNIX timestamp (UTC). If failover is not configured, this element is not returned.
<last_data_sync_status>	Contains a string showing the status of the last data sync. If failover is not configured, this element is not returned.

Query Example: check_health

check_health	`https://access.example.com/api/command?action=check_health`

HTTP Status Check

In addition to using the API command above, you can use **https://access.example.com/check_health** to check the health of an appliance. This returns an HTTP status of 200 if the probe is successful and 500 (Server Error) if not. While you will see a simple human-readable message showing success or failure, no other data is exposed.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Command: get_api_info

The **get_api_info request** returns XML data containing the current API version information.

XML Response for get_api_info Query

<api_version>	The software version of the current BeyondTrust API.
<timestamp>	The server's current timestamp at the time this report was pulled.
<permissions>	The permissions of the API account used to issue this command. The permissions shown are detailed below.

Element Names and Attributes

	/get_api_info/permissions/permission
perm_backup	Integer value (**1** or **0**) indicating if the API account has permission to use the backup API.
perm_command	String indicating if the API account has **full access** to the command API, **read_only** access, or no access (**deny**).
perm_ecm	Integer value (**1** or **0**) indicating if the API account has permission to use the Endpoint Credential Manager (ECM) API.
perm_reporting	Integer value (**1** or **0**) indicating if the API account has permission to use the reporting API.
perm_scim	Integer value (**1** or **0**) indicating if the API account has permission to use the SCIM API.

Query Example: get_api_info

get_api_info	`https://access.example.com/api/command?action=get_api_info`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Command: set_failover_role

The **set_failover_role** command sets the failover role of an appliance to either primary or backup.

The API account used to issue this command must have full access to the command API.

Required Parameter for set_failover_role

role=[string]	The role to assign to this appliance. Can be either **primary** or **backup**.

Optional Parameters for set_failover_role

data_sync_first=[boolean]	To perform a data sync with the peer appliance before failing over, set this to **1**. All users on the existing primary appliance will be disconnected during the data sync, and no other operations will be available until the swap is complete. To fail over without a final data sync, set this to **0**.
force=[boolean]	This option is only applicable when contacting the primary appliance and attempting to set its role to backup. If this is set to **1**, then this appliance will become the backup even if the peer appliance cannot be contacted.

XML Response for set_failover_role Query

<success>	If a data sync is being performed first, returns a message of **Successfully started data sync. Role change will occur upon successful completion.** Otherwise, returns a message of **Successfully changed role.**
<error>	Returns an error message if the role was not set successfully.

Query Examples: set_failover_role

Set failover role to primary	`https://access.example.com/api/command?action=set_failover_role&role=primary`
Set failover role to backup	`https://access.example.com/api/command?action=set_failover_role&role=backup`
Set failover role to primary and perform a data sync	`https://access.example.com/api/command?action=set_failover_role&role=primary&data_sync_first=1`
Set failover role to backup and perform a data sync	`https://access.example.com/api/command?action=set_failover_role&role=backup&data_sync_first=1`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Set failover role to backup even if the primary appliance cannot be contacted	`https://access.example.com/api/command?action=set_failover_role&role=backup&force=1`
Set failover role to backup even if the primary appliance cannot be contacted, and perform a data sync	`https://access.example.com/api/command?action=set_failover_role&role=backup&data_sync_first=1&force=1`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Access Console Scripting and Client Scripting API

The BeyondTrust access console scripting feature is composed of three parts:

1. The BeyondTrust Access Console Script file format
2. Command line parameters for the access console
3. The BeyondTrust client scripting API

The BeyondTrust Access Console Script File

A BeyondTrust Console Script (BRCS) is a file that contains a sequence of commands to be executed by the BeyondTrust access console. The file extension is in the format "brcs-<companySiteName>." The Company Site Name is the name used to access your BeyondTrust site. During installation, the BeyondTrust access console uses the OS to associate the access console with the BRCS file type. Therefore, users can double-click a BRCS file and have it automatically executed by the BeyondTrust access console.

BRCS files have the following format:

```
BRCS1.0
<command>
<command>
…
```

This is more formally expressed as:

```
brcs_file = header , newline , commands ;
header = "BRCS" , version ;
version = digit , "." , digit ;
commands = command { newline , command } ;
digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;
newline = "\n" | "\r\n" ;
```

Note: *Script files can have a maximum of 10 commands.*

Each command consists of a set of key-value pairs separated by "&". The key in each pair is separated from the value by "=". Keys and values use the percent-encoding algorithm described in RFC3986 section 2.1. This is commonly referred to as url-encoding or url-escaping. It is commonly seen in the address bar of web browsers to represent the parameters passed to a web server. Commands have the following format:

```
action=<action>&parameter1=value1&parameter2=value2...
```

This is more formally expressed as:

```
command = "action=", value, [ parameters ] ;
parameters = "&", parameter, [ parameters ] ;
parameter = url_encoded_string, "=", url_encoded_string ;
url_encoded_string = {* see RFC 3986 *} ;
```


©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Command Line Parameters for the Access Console

Two command line parameters exist in the access console to support BRCS:

```
run-script <BRCS command>
run-script-file <path to BRCS file>
```

These command line parameters allow users to implement BRCS login via the command line.

Different behaviors can be seen when running a script from the command line, depending on the state of the access console:

- If the access console is not running, then attempting to run a script from the command line causes the access console to start the login dialog. After the user successfully logs in, the script is run.
- If the access console is already running but the user is not logged in, then the login dialog is shown. After the user logs in, the script is run.
- If the access console is already running and the user is already logged in, then attempting to run a script from the command line causes the existing instance of the access console to run the script.

Access console exit status:

- If an invalid script is given on the command line, then the access console terminates with an exit status > 0.
- If a valid script is given on the command line, then the access console terminates with an exit status of 0.

Examples:

```
bomgar-acc.exe --run-script "action=start_jump_item_
session&client.hostname=ABCEF02&session.custom.external_key=123456789"
bomgar-acc.exe --run-script-file my_script_file.brcs-beta60
```

The BeyondTrust Client Scripting API

The client scripting API enables you to generate a BeyondTrust Console Scripting (BRCS) file which allows you to send commands to the BeyondTrust access console from external applications.

Customers can use the client scripting API to generate BRCS files that can start a session with a specific Jump Item or simply to log into the access console.

The client scripting API URL is **https://access.example.com/api/client_script**.

This API accepts a client type (**rep**), an operation to perform (**generate**), a command to put in the script file, and a set of parameters to pass to the command. Here is an example of a valid Client Scripting API request:

```
https://access.example.com/api/client_script?type=rep&operation=generate&action=start_jump_item_
session&client.hostname=ABCDEFG02
```

The above request prompts the user to download a BeyondTrust access console script file. After downloading the script file, the user can run it using the access console. In this case, the script file contains commands to start a session with the Jump Item whose hostname, comments, public IP, or private IP matches the search string "ABCDEF02".

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Parameters for Client Scripting API

type=rep type=web_console	The BeyondTrust client to which the command applies. Currently the API only supports **rep** or **web_console** as the client type.
operation=generate operation=execute	The operation to perform. Currently the API only supports **generate** or **execute** as the operation. ***Note:*** *If the type is* ***rep****, the operation should be* ***generate****. If the type is* ***web_console****, the operation should be* ***execute****.*
action=<command>&parameter=[value]	The name of the command to run and the necessary parameters. Available actions include: • login • start_jump_item_session • push_and_start_local • push_and_start_remote • start_rdp_session • start_shell_jump_session Two actions are automatically added to the BRCS file: **login** and **delete_script_file**. The **delete_script_file** action has no parameters. ***Note:*** *The* ***web_console*** *type supports only the* ***start_jump_item_session*** *action.*

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Script Command: login

When generating any BeyondTrust Console Script, the **login** command is automatically added as the first command in the script file. It does not need to be specified in the URL used to generate the script file.

By default, this command opens the access console and attempts to log in using the credentials saved locally in the access console. If no credentials are saved, the command simply opens the access console login prompt. Once the user has correctly authenticated, the script continues running.

The **login** command has no effect if a user is already logged into the access console.

If you wish to specify the credentials to be used, you can create a separate script specifically to be used for logging in. The **login** command passes the login mechanism along with a username and password. Both username and password parameters are sent in plain text and is unencrypted.

IMPORTANT!

*You cannot specify multiple commands in the URL used to generate a script. For example, you cannot specify **login** and multiple **start_jump_item_session** commands in the same URL. Each command must be generated as a separate script.*

*However, a skilled developer may edit the **.brcs** script file once it has been generated in order to modify the login credentials and then run another command. BeyondTrust does not support scripts modified in this manner.*

Optional Parameters for login Command

mechanism=[string]	The mechanism to use for authentication. Currently, only **username_password** is supported. If this parameter is supplied, both other parameters must also be supplied.
username=[string]	The username of the account with which to log in. If this parameter is supplied, both other parameters must also be supplied.
password=[string]	The password of the account with which to log in. If this parameter is supplied, both other parameters must also be supplied.

Query Examples: login

Log into the access console, specifying the username and password	https://access.example.com/api/client_script?type=rep&operation=generate&action=login&mechanism=username_password&username=username&password=password

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Script Command: start_jump_item_session

The **start_jump_item_session** command attempts to start a session with a BeyondTrust Jump Item. Users may run this command for all Jump Items they are permitted to access via the Jump management interface in the access console.

Optional Parameters for the start_jump_item_session Command

jump.method	If specified, only Jump Items using the designated Jump method are included in the results. Acceptable values for this field are **push** (remote push), **local_push**, **pinned** (Jump Client), **rdp**, **vnc,** and **shelljump**.
credential_id	If specified, only a Jump Item with that specific credential ID associated is returned. This field has a maximum length of 255 characters.
search_string	Identifies the search criteria used to select and return specific Jump Items as results. ***Note:*** *This parameter is required only if no of the client fields below are specified.*
client.comments	If specified, only Jump Items with the given comments are included in the results. This field has a maximum length of 255 characters. Search is partial and case-insensitive.
client.hostname	If specified, only Jump Items with the given hostname are included in the results. This field has a maximum length of 255 characters. Search is partial and case-insensitive.
client.private_ip	If specified, only Jump Clients with the given private IP address are included in the results. This search field applies only to pinned clients. This field has a maximum length of 255 characters. Search is partial and case-insensitive.
client.public_ip	If specified, only Jump Clients with the given public IP address are included in the results. This search field applies only to pinned clients. This field has a maximum length of 255 characters. Search is partial and case-insensitive.
client.tag	If specified, only Jump Items with the given tag are included in the results. This field has a maximum length of 255 characters. Search is partial and case-insensitive.
session.custom.[custom field]=[string]	The code name and value of any custom fields. These fields must first be configured in **/login > Management > API Configuration**. Each attribute must be specified as a different parameter. Each custom field has a maximum length of 1024 characters. The maximum total size of all combined custom fields, including the external key, must be limited to 10KB.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

IMPORTANT!

*At least one **client.*** parameter must be specified. If multiple **client.*** parameters are specified, then only clients matching all criteria are returned.*

Query Examples: start_jump_item_session

Start a session with a Jump Item whose hostname contains "ABCDEF02"	https://access.example.com/api/client_script?type=rep&operation=generate&action=start_jump_item_session&client.hostname=ABCDEF02
Start a session with a Jump Item whose comments contain "maintenance" and whose tag contains "server"	https://access.example.com/api/client_script?type=rep&operation=generate&action=start_jump_item_session&client.comments=maintenance&client.tag=server
Start a session with a pinned Jump Client whose private IP address begins with "10.10.24" and associate custom attributes with the session	https://access.example.com/api/client_script?type=rep&operation=generate&action=start_jump_item_session&client.private_ip=10.10.24&jump.method=pinned&session.custom.custom_field1=Custom%20Value&session.custom.custom_field2=123

***Note:** If more than one Jump Item matches the search criteria, then a dialog opens, giving the user the option to select the appropriate Jump Item.*

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Script Command: push_and_start_local

The **push_and_start_local** command attempts to push the endpoint client client to a computer on the local network to start an access session. This can also be described as a local Jump.

Required Parameter for push_and_start_local Command

hostname=[string]	The hostname of the computer that is the target of the push and start operation. This field has a maximum length of 255 characters.

Optional Parameter for push_and_start_local Command

session.custom.[custom field]=[string]	The code name and value of any custom fields. These fields must first be configured in **/login > Management > API Configuration**. Each attribute must be specified as a different parameter. Each customer field has a maximum length of 1024 characters. The maximum total size of all combined custom fields, including the external key, must be limited to 10KB.

Query Examples: push_and_start_local

Jump to the local network computer "ABCDEF02"	https://access.example.com/api/client_script?type=rep&operation=generate&action=push_and_start_local&hostname=ABCDEF02
Jump to the local network computer "ABCDEF02" and associate custom attributes with the session	https://access.example.com/api/client_script?type=rep&operation=generate&action=push_and_start_local&hostname=ABCDEF02&session.custom.custom_field1=Custom%20Value&session.custom.custom_field2=123

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Script Command: push_and_start_remote

The **push_and_start_remote** command attempts to push the endpoint client client to a computer on a remote network through a Jumpoint in order to start an access session. This can also be described as a remote Jump.

Required Parameter for push_and_start_remote Command

target=[string]	The hostname or IP address of the target machine.

Optional Parameters for push_and_start_remote Command

jumpoint=[string]	The Jumpoint through which to start the session. This Jumpoint must be on the same subnet as the target computer. If not specified and the user has access to only one Jumpoint, then that Jumpoint is used automatically. If not specified and the user has access to more than one Jumpoint, then a dialog opens from which the user must select a Jumpoint.
session.custom.[custom field]=[string]	The code name and value of any custom fields. These fields must first be configured in **/login > Management > API Configuration**. Each attribute must be specified as a different parameter. Each customer field has a maximum length of 1024 characters. The maximum total size of all combined custom fields, including the external key, must be limited to 10KB.

Query Examples: push_and_start_remote

Jump to the remote computer "ABCDEF02" through the Jumpoint "Network01"	https://access.example.com/api/client_script?type=rep&operation=generate&action=push_and_start_remote&target=ABCDEF02&jumpoint=Network01
Jump to the remote computer "ABCDEF02" through the Jumpoint "Network01" and associate custom attributes with the session	https://access.example.com/api/client_script?type=rep&operation=generate&action=push_and_start_remote&target=ABCDEF02&jumpoint=Network01&session.custom.custom_field1=Custom%20Value&session.custom.custom_field2=123

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

API Script Command: start_shell_jump_session

The **start_shell_jump_session** command initiates a Shell Jump session, creating an SSH or Telnet connection to a remote network device.

Required Parameter for the start_shell_jump_session Command

target=[string]	The hostname or IP address of the machine targeted for a Shell Jump session.

Optional Parameters for the start_shell_jump_session Command

jumpoint=[string]	The Jumpoint through which to start the Shell Jump session. This Jumpoint must be on the same subnet as the target computer. If not specified and the user has access to only one Jumpoint, then that Jumpoint is used automatically. If not specified and the user has access to more than one Jumpoint, then a dialog opens from which the user must select a Jumpoint.
username=[string]	The username to use when authenticating. If not specified, the user must enter the username.
protocol=[string]	The network protocol to use. May be one of **ssh** (default) or **telnet**.
port=[integer]	The port number on which to connect. Defaults to 22.
terminal	The terminal type to use. May be one of **xterm** (default) or **vt100**.
session.custom.[custom field]=[string]	The code name and value of any custom fields. These fields must first be configured in **/login > Management > API Configuration**. Each attribute must be specified as a different parameter. Each customer field has a maximum length of 1024 characters. The maximum total size of all combined custom fields, including the external key, must be limited to 10KB.

Query Examples: start_shell_jump_session

Start a Shell Jump session with the computer "ABCDEF02"	https://access.example.com/api/client_script?type=rep&operation=generate&action=start_shell_jump_session&target=ABCDEF02
Start a Shell Jump session with the computer "ABCDEF02" through the Jumpoint "Network01"	https://access.example.com/api/client_script?type=rep&operation=generate&action=start_shell_jump_session&target=ABCDEF02&jumpoint=Network01
Start a Shell Jump session with the computer "ABCDEF02" through the Jumpoint "Network01". Authenticate with "jsmith", and use a Telnet protocol through port 40 with terminal type vt100	https://access.example.com/api/client_script?type=rep&operation=generate&action=start_shell_jump_session&target=ABCDEF02&jumpoint=Network01&username=jsmith&protocol=telnet&port=40&terminal=vt100
Start a Shell Jump session with the	https://access.example.com/api/client_script?type=rep&operation=generate&

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

computer "ABCDEF02" and associate custom attributes with the session	action=start_shell_jump_session&target=ABCDEF02&session.custom.custom_field1=Custom%20Value&session.custom.custom_field2=123

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Reporting API

The reporting API is designed to enable you to pull reporting data in XML format, suitable for importing into external databases and applications. The data presented is the same as in the session reports of the **/login** administrative interface.

The reporting API is an authenticated API. For instructions on using authenticated APIs using OAuth, please see "Authenticate to the Privileged Remote Access API" on page 5.

XML data is pulled by sending a simple HTTP request to the Secure Remote Access Appliance. The request can be sent using any HTTPS-capable socket library, scripting language module, or a URL fetcher such as **cURL** or **wget**. Either **GET** or **POST** may be used as the request method.

POST requests must include a "Content-Type: application/x-www-form-urlencoded" HTTP header when supplying parameters in the request body, and the parameters must be url-encoded. Multipart POST requests are not supported.

IMPORTANT!

When making consecutive API calls, you must close the connection after each API call.

The reporting API URL is **https://access.example.com/api/reporting**.

An XML schema which formally describes the format of the returned reporting data is available at **https://access.example.com/api/reporting.xsd**.

Required Parameter for Reporting API

generate_report=[string]	The type of report to be generated. Report types can be any of the following: AccessSession AccessSessionListing AccessSessionRecording Team AccessSessionSummary CommandShellRecording UserRecording

The reporting API returns XML responses that declare a namespace. If you are parsing these responses with a namespace-aware parser, you will need to set the namespace appropriately or ignore the namespace while parsing the XML.

- Reporting API: https://www.beyondtrust.com/namespaces/API/reporting

Note: *The above namespace is returned XML data and is not a functional URL.*

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Download Reports with AccessSession

The **AccessSession** query returns full information for all sessions which match given search parameters. You may use any of the following sets of parameters to generate reports:

- **start_date** and **duration**
- **start_time** and **duration**
- **end_date** and **duration**
- **end_time** and **duration**
- **lsid**
- **lsids**

The API account used to call this report must have access to the reporting API.

Parameters for AccessSession

start_date=[YYYY-MM-DD]	Specifies that the report should return all sessions, even those still in progress, that began on or after this date and that are within the duration specified below.
start_time=[timestamp]	Specifies that the report should return all sessions, even those still in progress, that began at or after this time and that are within the duration specified below. The time must be a UNIX timestamp (UTC).
end_date=[YYYY-MM-DD]	Specifies that the report should return only closed sessions that ended on or after this date and that are within the duration specified below.
end_time=[timestamp]	Specifies that the report should return only closed sessions that ended at or after this time and that are within the duration specified below. The time must be a UNIX timestamp (UTC).
duration=[integer]	Length of time from the specified date or time for which you wish to pull reports, or **0** to pull from the specified date to present. If **start_date** or **end_date** is specified, **duration** will represent days; if **start_time** or **end_time** is specified, **duration** will represent seconds.
lsid=[string]	The ID of the session for which you wish to see details.
lsids=[comma-separated strings]	A comma-delimited list of the IDs of sessions for which you wish to see details.

XML Response for AccessSession Query

<session_list>	Contains a **<session>** element for each session that matches the given criteria. If no sessions are returned, this element will contain no **<session>** elements. If an error occurs during the search, it will contain an **<error>** element describing the problem.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Element Names and Attributes

/session_list/session

lsid (attribute)	A string which uniquely identifies this session.
<session_type>	Indicates the type of session for which the report was run. The value will always be **support** in the current BeyondTrust API version.
<lseq>	An incrementing number used to represent sessions in a non-string format.  ***Note:*** *The LSEQ element is not guaranteed to be unique or strictly sequential.*
<start_time>	The date and time the session was begun. Data is returned in ISO 8601 format. Also contains a **timestamp** attribute which displays the start time as a UNIX timestamp (UTC).
<end_time>	The date and time the session was ended. Data is returned in ISO 8601 format. Also contains a **timestamp** attribute which displays the end time in UNIX timestamp (UTC). This element will be empty for sessions which are still in progress when the report was run or which closed abnormally.
<duration>	Session length in HH:MM:SS format.
<jump_group>	The element's content is the name of the Jump Group. For Personal Jump Groups, the name of the Jump Group is the Private Display Name of the representative who owns the Jump Group. The **<jump_group>** element has two attributes: **type**: This is the Jump Group's type, which can be "shared" or "personal". **id**: This is the Jump Group's unique ID for its type. Jump Groups of different types can have the same ID. For Personal Jump Groups, this is the unique ID of the user who owns the Jump Group. Each user can only have a single Personal Jump Group.
<jumpoint>	The name of the Jumpoint through which this session was initiated, if any. Also contains an **id** attribute, which displays the unique ID assigned to the Jumpoint.
<custom_attributes>	Contains a **<custom_attribute>** element for each custom field assigned to a session. This element displays only if custom fields have been defined. The format of each **<custom_attribute>** element is described below.
<session_chat_view_url>	The URL at which this session's chat transcript can be viewed in a web browser. This element is displayed only for sessions that have successfully ended.
<session_chat_download_url>	The URL at which this session's chat transcript can be downloaded. This element is displayed only for sessions that have successfully ended.
<session_recording_view_url>	The URL at which the video of the session may be viewed in a web browser. This element is displayed only if screen sharing recording was enabled at the time of the session and only if the user initiated screen sharing during the session. It is available only for sessions that have successfully ended.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

<session_recording_download_url>	The URL at which the video of the session may be downloaded. This element is displayed only if screen sharing recording was enabled at the time of the session and only if the user initiated screen sharing during the session. It is available only for sessions that have successfully ended.
<command_shell_recordings>	Contains a **<command_shell_recording>** element for each command shell that was initiated during the session. This element is displayed only if the user opened a remote command shell during the session, if command shell recording was enabled at the time of the session, and if the requesting user has permission to view session recordings. Each **<command_shell_recording>** element contains the child elements **<download_url>** and **<view_url>** as described below.
<file_transfer_count>	The number of file transfers which occurred during the session.
<file_move_count>	The number of files renamed via the **File Transfer** interface during the session.
<file_delete_count>	The number of files deleted via the **File Transfer** interface during the session.
<primary_customer>	Lists the **gsnumber** as an attribute and as an element, the **name** of the remote endpoint accessed by the user.
<primary_rep>	Lists the **gsnumber** and **id** as attributes and as an element, the **name** of the user who owned the session.
<customer_list>	A list of all endpoints accessed in the session. There should always be exactly one endpoint per session in the current BeyondTrust API version. The format of each **<customer>** element is described below.
<rep_list>	A list of all users who participated in the session, whether as the session owner or as conference members. The format of each **<representative>** element is described below.
<session_details>	Contains a chronological list of all events which occurred during the session. This element contains one or more child **<event>** elements, described below.

/session_list/session/custom_attributes/custom_attribute

display_name (attribute)	The display name assigned to the custom attribute.
code_name (attribute)	The code name assigned to the custom attribute.

/session_list/session/command_shell_recordings/command_shell_recording

instance (attribute)	The instance of the command shell session, starting with **0**.
<download_url>	The URL at which the video of the command shell session may be downloaded.
<view_url>	The URL at which the video of the command shell session may be viewed in a web browser.

/session_list/session/customer_list/customer

gsnumber (attribute)	Uniquely identifies the endpoint in regards to its current connection to the Secure Remote Access Appliance. A gsnumber may be recycled, so while two endpoints connected at the same time will never have the same gsnumber, one endpoint may

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

	have a gsnumber that was assigned to another endpoint in the past. Can be used to correlate a **<customer>** element with a **<primary_customer>** or with an event's **<performed_by>** or **<destination>** element.
<username>	The name used to identify the endpoint during the session.
<public_ip>	The endpoint's public IP address.
<private_ip>	The endpoint's private IP address.
<hostname>	The hostname of the endpoint.
<os>	The operating system of the endpoint.

/session_list/session/rep_list/representative

gsnumber (attribute)	Uniquely identifies the user in regards to their current connection to the Secure Remote Access Appliance. A gsnumber is assigned on a per-connection basis, so if a user leaves a session and then rejoins without logging out of the Secure Remote Access Appliance, their gsnumber will remain the same. However, if the user's connection is terminated for any reason, when that user logs back into the Secure Remote Access Appliance, they will be assigned a new gsnumber and will also appear multiple times in the <**rep_list**> element. A gsnumber may be recycled, so while two people connected at the same time will never have the same gsnumber, one person may have a gsnumber that was assigned to another person in the past. Can be used to correlate a <**representative**> element with a <**primary_rep**> or with an event's <**performed_by**> or <**destination**> element.
id (attribute)	Unique ID assigned to the user.
<username>	The username assigned to the user.
<display_name>	The display name assigned to the user. Note that this field contains the display name's value at the time of the conference, which may not match the current value if the **display_name** has subsequently been changed.
<public_ip>	The user's public IP address.
<private_ip>	The user's private IP address.
<hostname>	The hostname of the user's computer.
<os>	The operating system of the user's computer.
<session_owner>	Integer value (**1** or **0**) indicating whether the user was the owner of the session or was merely a conference member.
<seconds_involved>	Integer value indicating the number of seconds the user was involved in this session.
<invited>	Integer value (**1**) present only if the user is an invited user.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

/session_list/session/session_details/event

<table>
<tr><td>timestamp (attribute)</td><td>The system time at which the event occurred.</td></tr>
<tr><td>event_type (attribute)</td><td>The type of event which occurred. Event types include the following:<table><tr><td>Chat Message</td><td>Registry Imported</td></tr><tr><td>Command Shell Session Started*</td><td>Registry Key Added</td></tr><tr><td>Conference Member Added</td><td>Registry Key Deleted</td></tr><tr><td>Conference Member Departed</td><td>Registry Key Renamed</td></tr><tr><td>Conference Member State Changed</td><td>Registry Value Added</td></tr><tr><td>Conference Owner Changed</td><td>Registry Value Deleted</td></tr><tr><td>Credential Injection Attempt</td><td>Registry Value Modified</td></tr><tr><td>Credential Injection Attempt Failed</td><td>Registry Value Renamed</td></tr><tr><td>Directory Created</td><td>Screen Recording</td></tr><tr><td>File Deleted</td><td>Screenshot Captured</td></tr><tr><td>File Download</td><td>Service Access Allowed</td></tr><tr><td>File Download Failed</td><td>Session End</td></tr><tr><td>File Moved</td><td>Session Foreground Window Changed</td></tr><tr><td>File Upload</td><td>Session Start</td></tr><tr><td>File Upload Failed</td><td>System Information Retrieved</td></tr><tr><td>Registry Exported</td><td></td></tr></table>*Will only appear if recording is enabled for this session.</td></tr>
<tr><td><performed_by></td><td>The entity that performed the action. Indicates the entity's gsnumber and also its type, indicating whether this action was performed by the system, a endpoint, or a representative.</td></tr>
<tr><td><destination></td><td>The entity to which the event was directed. Indicates the entity's gsnumber and also its type, indicating whether this action was directed to the system, a customer, or a user.</td></tr>
<tr><td><body></td><td>The text of the message as displayed in the chat log area.</td></tr>
<tr><td><encoded_body></td><td>Can be shown in place of the <body> element above. Contains the base64 (RFC 2045 section 6.8) encoded value of what would have been shown in the <body> element, and is shown ONLY if the <body> text contains characters that are invalid according to XML specification. These characters are typically the result of binary data being sent through chat messages.</td></tr>
<tr><td><filename></td><td>The name of the transferred file.</td></tr>
<tr><td><files></td><td>If this event involved the transferring of files, then this element will contain a <file> element for every file transferred.</td></tr>
<tr><td><filesize></td><td>An integer indicating the size of the transferred file.</td></tr>
</table>

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

<system_information>	Applies only to **System Information Retrieved** events wherein the system information is pulled automatically upon session start. This element contains multiple **<category>** child elements as described below. ***Note:*** *System information is logged only when pulled automatically at the beginning of the session and not when specifically requested by the user. This is to prevent overload with the large amount of dynamic data that can be retrieved from the remote system.*
<data>	Contains an arbitrary number of **<value name="_" value="_" />** elements. The name and number of these elements varies based on **event_type**. For example, when a user joins the session, a **Conference Member Added** event would contain **<value>** elements for the user's **name**, **private_ip**, **public_ip**, **hostname**, and **os**.

/session_list/session/session_details/event/system_information/category

<description>	Contains multiple **<field>** elements, each of which contains a descriptor for the specific data field. For example, the **Drives** category would have **<field>** elements **Drive**, **Type**, **Percent Used**, etc. These **<field>** elements can be compared to table header cells.
<data>	Contains multiple **<row>** elements, each of which contains multiple **<field>** elements that correspond to the **<field>** elements above. For example, the **Drives** category would have a separate **<row>** for each drive on the endpoint computer. An example **<row>** might contain **<field>** elements **C:**, **Local Disk**, **60%**, etc. These **<row>** elements can be compared to table rows, with each **<field>** element a table cell.

Query Examples for AccessSession

Sessions started July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=AccessSession&start_date=2016-07-01&duration=0`
Sessions started the month of July 2016	`https://access.example.com/api/reporting?generate_report=AccessSession&start_date=2016-07-01&duration=31`
Sessions started 8:00 AM July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=AccessSession&start_time=1467360000&duration=0`
Sessions started 8:00 AM July 1 2016 to 6:00 PM July 1 2016	`https://access.example.com/api/reporting?generate_report=AccessSession&start_time=1467360000&duration=36000`
Sessions ended July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=AccessSession&end_date=2016-07-01&`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

	`duration=0`
Sessions ended the month of July 2016	`https://access.example.com/api/reporting?generate_report=AccessSession&end_date=2016-07-01&duration=31`
Sessions ended 8:00 AM July 1 2016 to 6:00 PM July 1 2016	`https://access.example.com/api/reporting?generate_report=AccessSession&end_time=1467360000&duration=36000`
Session c69a8e10bea9428f816cfababe9815fe	`https://access.example.com/api/reporting?generate_report=AccessSession&lsid=c69a8e10bea9428f816cfababe9815fe`
Sessions c69a8e10bea9428f816cfababe9815fe, a5eeaa58591047b88556f944804227b0, 5bf07601298b495b87310da9ce571e22	`https://access.example.com/api/reporting?generate_report=AccessSession&lsids=c69a8e10bea9428f816cfababe9815fe,a5eeaa58591047b88556f944804227b0,5bf07601298b495b87310da9ce571e22`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Download Reports with AccessSessionListing

The **AccessSessionListing** query returns a list of session IDs, external keys, and availability of a recording for sessions which match given search parameters. You may use any of the following sets of parameters to generate reports:

- **start_date** and **duration**
- **start_time** and **duration**
- **end_date** and **duration**
- **end_time** and **duration**

The API account used to call this report must have access to the reporting API.

Parameters for AccessSessionListing

start_date=[YYYY-MM-DD]	Specifies that the report should return all sessions, even those still in progress, that began on or after this date and that are within the duration specified below.
start_time=[timestamp]	Specifies that the report should return all sessions, even those still in progress, that began at or after this time and that are within the duration specified below. The time must be a UNIX timestamp (UTC).
end_date=[YYYY-MM-DD]	Specifies that the report should return only closed sessions that ended on or after this date and that are within the duration specified below.
end_time=[timestamp]	Specifies that the report should return only closed sessions that ended at or after this time and that are within the duration specified below. The time must be a UNIX timestamp (UTC).
duration=[integer]	Length of time from the specified date or time for which you wish to pull reports, or **0** to pull from the specified date to present. If **start_date** or **end_date** is specified, **duration** represents days; if **start_time** or **end_time** is specified, **duration** represents seconds.

XML Response for AccessSessionListing Query

<session_summary_list>	Contains a **<session_summary>** element for each session that matches the given criteria. If no sessions are returned, this element will contain no **<session_summary>** elements. If an error occurs during the search, it will contain an **<error>** element describing the problem.

Element Names and Attributes

/session_summary_list/session_summary

lsid (attribute)	The session ID for the given session.
has_recording (attribute)	Integer (**1** or **0**) indicating if the given session has a session recording.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

external_key (attribute)	An arbitrary string that can link this session to an identifier on an external system, such as a customer relationship management ticket ID. This can be input from within the access console or defined programmatically. This element is displayed only if an external key has been defined.

Query Examples for AccessSessionListing

Sessions started July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=AccessSessionListing&start_date=2016-07-01&duration=0`
Sessions started the month of July 2016	`https://access.example.com/api/reporting?generate_report=AccessSessionListing&start_date=2016-07-01&duration=31`
Sessions started 8:00 AM July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=AccessSessionListing&start_time=1467360000&duration=0`
Sessions started 8:00 AM July 1 2016 to 6:00 PM July 1 2016	`https://access.example.com/api/reporting?generate_report=AccessSessionListing&start_time=1467360000&duration=36000`
Sessions ended July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=AccessSessionListing&end_date=2016-07-01&duration=0`
Sessions ended the month of July 2016	`https://access.example.com/api/reporting?generate_report=AccessSessionListing&end_date=2016-07-01&duration=31`
Sessions ended 8:00 AM July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=AccessSessionListing&end_time=1467360000&duration=0`
Sessions ended 8:00 AM July 1 2016 to 6:00 PM July 1 2016	`https://access.example.com/api/reporting?generate_report=AccessSessionListing&end_time=1467360000&duration=36000`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Download Reports with AccessSessionSummary

The **AccessSessionSummary** query returns an overview of access session statistics by user. You may use any of the following sets of parameters to generate reports:

- **start_date**, **duration**, and **report_type**
- **start_time**, **duration**, and **report_type**
- **end_date**, **duration**, and **report_type**
- **end_time**, **duration**, and **report_type**

The API account used to call this report must have access to the reporting API.

Parameters for AccessSessionSummary

start_date=[YYYY-MM-DD]	Specifies that the report should return all sessions, even those still in progress, that began on or after this date and that are within the duration specified below.
start_time=[timestamp]	Specifies that the report should return all sessions, even those still in progress, that began at or after this time and that are within the duration specified below. The time must be a UNIX timestamp (UTC).
end_date=[YYYY-MM-DD]	Specifies that the report should return only closed sessions that ended on or after this date and that are within the duration specified below.
end_time=[timestamp]	Specifies that the report should return only closed sessions that ended at or after this time and that are within the duration specified below. The time must be a UNIX timestamp (UTC).
duration=[integer]	Length of time from the specified date or time for which you wish to pull reports, or 0 to pull from the specified date to present. If **start_date** or **end_date** is specified, **duration** represents days; if **start_time** or **end_time** is specified, **duration** represents seconds.
report_type=[string]	In the current BeyondTrust API version, **user** is the only accepted value.

XML Response for AccessSessionSummary Query

<summary_list>	Contains a **<summary>** element for each record that matches the given criteria. If no sessions are returned, this element will contain no **<summary>** elements. If an error occurs during the search, it will contain an **<error>** element describing the problem.

Element Names and Attributes

/summary_list/summary

id (attribute)	Returns the user's unique ID.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

type (attribute)	Specifies the report type generated. This value is always **user** in the current API version.
<display_name>	The display name of the user. Note that since summary reports represent an aggregation of sessions over a period of time, the display name used is the current value for the user, which may have been edited since the time of the first returned session.
<total_sessions>	The total number of sessions run by the user in the time specified.
<avg_sessions_per_weekday>	The average number of sessions conducted on Monday through Friday by the user, expressed as a decimal rounded to the nearest point.
<avg_duration>	The average length of each session, expressed as HH:MM:SS.

Query Examples

Sessions started July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=AccessSessionSummary&start_date=2016-07-01&duration=0&report_type=user`
Sessions started the month of July 2016, by user	`https://access.example.com/api/reporting?generate_report=AccessSessionSummary&start_date=2016-07-01&duration=31&report_type=user`
Sessions started 8:00 AM July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=AccessSessionSummary&start_time=1467360000&duration=0&report_type=user`
Sessions started 8:00 AM July 1 2016 to 6:00 PM July 1 2016	`https://access.example.com/api/reporting?generate_report=AccessSessionSummary&start_time=1467360000&duration=36000&report_type=user`
Sessions ended July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=AccessSessionSummary&end_date=2016-07-01&duration=0&report_type=user`
Sessions ended the month of July 2016	`https://access.example.com/api/reporting?generate_report=AccessSessionSummary&end_date=2016-07-01&duration=31&report_type=user`
Sessions ended 8:00 AM July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=AccessSessionSummary&end_time=1467360000&duration=0&report_type=user`
Sessions ended 8:00 AM July 1 2016 to 6:00 PM July 1 2016	`https://access.example.com/api/reporting?generate_report=AccessSessionSummary&end_time=1467360000&duration=36000&report_type=user`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Download Reports with AccessSessionRecording

The **AccessSessionRecording** query returns the requested access session recording file. Depending on your browser, this query will either immediately begin download or prompt you to open or save the file. Note that the requesting user must have permission to view session recordings.

The API account used to call this report must have access to the reporting API.

Parameter for AccessSessionRecording

lsid=[string]	The session ID for which you wish to download the video recording of the session.

Query Example for AccessSessionRecording

AccessSessionRecording: Session c69a8e10bea9428f816cfababe9815fe	`https://access.example.com/api/reporting?generate_report=AccessSessionRecording&lsid=c69a8e10bea9428f816cfababe9815fe`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Download Reports with CommandShellRecording

The **CommandShellRecording** query returns the requested command shell recording. Depending on your browser, this query will either immediately begin download or prompt you to open or save the file. Note that the requesting user must have permission to view session recordings.

The API account used to call this report must have access to the reporting API.

Parameters for CommandShellRecording

lsid=[string]	The session ID for which you wish to download the video recording of the command shell.
instance=[integer]	The instance number of the command shell recording you wish to download. Instances are enumerated starting with **0**. The instance number can be obtained from the **AccessSession** report.

Optional Parameter for CommandShellRecording

format=[string]	If this parameter has the value of **txt**, the command shell output will be in a text format instead of a recording.

Query Examples for CommandShellRecording

CommandShellRecording: First shell instance of session c69a8e10bea9428f816cfababe9815fe	`https://access.example.com/api/reporting?generate_report=CommandShellRecording&lsid=c69a8e10bea9428f816cfababe9815fe&instance=0`
CommandShellRecording: Third shell instance of session c69a8e10bea9428f816cfababe9815fe	`https://access.example.com/api/reporting?generate_report=CommandShellRecording&lsid=c69a8e10bea9428f816cfababe9815fe&instance=2`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Download Reports with Team

The **Team** query returns information about activity within a team. You may use any of the following sets of parameters to generate reports:

- **start_date** and **duration**
- **start_time** and **duration**
- **end_date** and **duration**
- **end_time** and **duration**

The API account used to call this report must have access to the reporting API.

Parameters for Team

start_date=[YYYY-MM-DD]	Specifies that the report should return team activity that began on or after this date and that is within the duration specified below.
start_time=[timestamp]	Specifies that the report should return team activity that began at or after this time and that is within the duration specified below. The time must be a UNIX timestamp (UTC).
end_date=[YYYY-MM-DD]	Specifies that the report should return team activity that ended on or after this date and that is within the duration specified below.
end_time=[timestamp]	Specifies that the report should return team activity that ended at or after this time and that is within the duration specified below. The time must be a UNIX timestamp (UTC).
duration=[integer]	Length of time from the specified date or time for which you wish to pull reports, or **0** to pull from the specified date to present. If **start_date** or **end_date** is specified, **duration** will represent days; if **start_time** or **end_time** is specified, **duration** will represent seconds.

Optional Parameter for Team

team_id=[integer]	The numeric ID of the team by which to filter results. Only the activity within the specified team will be returned. If this parameter is not specified, results from all teams will be returned.

XML Response for Team Query

<team_activity_list>	Contains a **<team_activity>** element for each team with any activity within the given parameters. If no teams are returned, this element will contain no **<team_activity>** elements. If an error occurs during the search, it will contain an **<error>** element describing the problem. Also contains **<start_time>** and **<end_time>** elements displaying the time parameters in the system time and with a **timestamp** attribute in UTC.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Element Names and Attributes

/team_activity_list/team_activity

id (attribute)	Integer representing the team's unique ID.
name (attribute)	The display name of the team. Note that this field contains the team name as it currently appears, which may not match the value at the time of the conference if the team name has been subsequently changed.
<logged_in_privileged users>	Contains a **<representative>** element for each user in that team who was logged into the access console before the first event in the report occurred. If no users were logged in at the start time, this element will be empty.
<events>	Contains an **<event>** element for each event that occurred within this team.

/team_activity_list/team_activity/logged_in_representatives/representative

gsnumber (attribute)	Uniquely identifies the user in regards to their current connection to the Secure Remote Access Appliance. A gsnumber is assigned on a per-connection basis, so if a user leaves a session and then rejoins without logging out of the Secure Remote Access Appliance, their gsnumber will remain the same. However, if the user's connection is terminated for any reason, when that user logs back into the Secure Remote Access Appliance, they will be assigned a new gsnumber. A gsnumber may be recycled, so while two people connected at the same time will never have the same gsnumber, one person may have a gsnumber that was assigned to another person in the past. Can be used to correlate a **<representative>** element with an event's **<performed_by>** or **<destination>** element.
id (attribute)	Unique ID assigned to the user.
<display_name>	The display name assigned to the user. Note that this field contains the display name's value at the time of the conference, which may not match the current value if the **display_name** has subsequently been changed.
<public_ip>	The user's public IP address.
<private_ip>	The user's private IP address.

/team_activity_list/team_activity/events/event

timestamp (attribute)	The system time at which the event occurred.
event_type (attribute)	The type of event which occurred. Event types include the following: Chat Message Conference Member Added Conference Member Departed Jump Item Authorization Request Jump Item Authorization Request Utilized Pinned Session Moved Away from Queue

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

	Conference Member State Changed / Pinned Session Moved to Queue File Download / Representative Monitoring Started File Download Failed / Representative Monitoring Stopped File Upload / Session Deployed to Queue File Upload Failed / Session Undeployed from Queue Files Shared
<performed_by>	The entity that performed the action. Indicates the entity's **gsnumber** and also its **type**, indicating whether this entity was the system or a user.
<destinations>	If this event was targeted to one or more specific users, it will contain one or more **<destination>** elements as described below.
<files>	If this event involved the transfer of files, then this element will contain a **<file>** element for every file transferred.
<data>	Contains an arbitrary number of **<value name="_" value="_" />** elements. The name and number of these elements varies based on the **event_type**. For example, when a user logs into the access console, a **Conference Member State Changed** event would contain **<value>** elements for the **hostname**, **os**, **private_ip**, **public_ip**, and **state**.
<body>	The text of the chat message as displayed in the chat log area.
<encoded_body>	Can be shown in place of the **<body>** element above. Contains the base64 (RFC 2045 section 6.8) encoded value of what would have been shown in the **<body>** element, and is shown ONLY if the **<body>** text contains characters that are invalid according to XML specification. These characters are typically the result of binary data being sent through chat messages.

/team_activity_list/team_activity/events/event/destinations/destination

gsnumber (attribute)	Indicates the **gsnumber** of the entity to which the event was destined.
type (attribute)	Indicates whether this entity was the system or a user.
[value]	The name of the entity to which the event was destined.

/team_activity_list/team_activity/events/event/files/file

name (attribute)	The name of the transferred file.
size (attribute)	An integer indicating the size of the transferred file.

Query Examples for Team

Activity started July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=Team&start_date=2016-07-01&duration=0`
Activity started the month of July 2016	`https://access.example.com/api/reporting?`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

	`generate_report=Team&start_date=2016-07-01&duration=31`
Activity started 8:00 AM July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=Team&start_time=1467360000&duration=0`
Activity started 8:00 AM July 1 2016 to 6:00 PM July 1 2016	`https://access.example.com/api/reporting?generate_report=Team&start_time=1467360000&duration=36000`
Activity started July 1 2016 to present for a specific team	`https://access.example.com/api/reporting?generate_report=Team&start_date=2016-07-01&duration=0&team_id=1`
Activity ended July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=Team&end_date=2016-07-01&duration=0`
Activity ended the month of July 2016	`https://access.example.com/api/reporting?generate_report=Team&end_date=2016-07-01&duration=31`
Activity ended 8:00 AM July 1 2016 to present	`https://access.example.com/api/reporting?generate_report=Team&end_time=1467360000&duration=0`
Activity ended 8:00 AM July 1 2016 to 6:00 PM July 1 2016	`https://access.example.com/api/reporting?generate_report=Team&end_time=1467360000&duration=36000`
Activity ended July 1 2016 to present for a specific team	`https://access.example.com/api/reporting?generate_report=Team&end_date=2016-07-01&duration=0&team_id=1`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

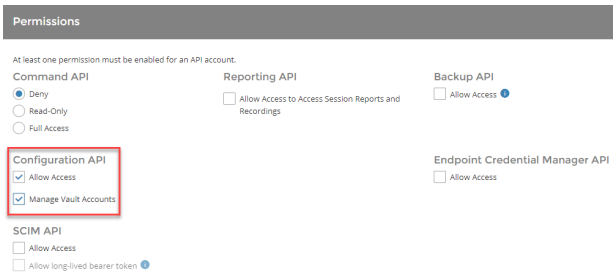
Vault Account Configuration APIs

You can list Vault accounts with the Vault Configuration API. Vault administrators can also create generic username/password and username/SSH key accounts using the API. This provides a programmatic way to onboard Vault accounts that can't automatically be discovered through Domain Discovery (Active Directory).

For more information on Vault account roles, please see Vault for Privileged Remote Access: New Member Role at https://www.beyondtrust.com/docs/privileged-remote-access/getting-started/admin/vault.htm.

API Account Permission for Vault Configuration APIs

Due to the sensitive information stored by Vault, there is a permission check box in **Management > API Configuration > Permissions** to manage which API Accounts are allowed to manage Vault Accounts. When checked, the API Account has permission to use all Vault APIs specified in this document. The permission can only be checked if the API Account already has permission to access the Configuration API. For new and existing API Accounts, the default value of the box is unchecked.



For more information, please see the section on Permissions in the API Configuration section of the Administrative Guide at https://www.beyondtrust.com/docs/privileged-remote-access/getting-started/admin/api-configuration.htm.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Backup API

The backup API is designed to enable you to automatically back up your BeyondTrust software configuration on a recurring basis. The backup file includes all your configuration settings and logged data except for recordings and some large files from the file store. The backup includes files from the file store only less than 200 KB in size and no more than 50 files total. In the event of a hardware failure, having a backup file helps to speed the disaster recovery process.

The backup API is an authenticated API. For instructions on using authenticated APIs using OAuth, see "Authenticate to the Privileged Remote Access API" on page 5. The API account used to issue this command must have access to the backup API.

Commands are executed by sending a simple HTTP request to the Secure Remote Access Appliance. The request can be sent using any HTTPS-capable socket library, scripting language module, or a URL fetcher such as **cURL** or **wget**. Either **GET** or **POST** may be used as the request method.

The backup API URL is **https://access.example.com/api/backup**.

Query Example

backup	`https://access.example.com/api/backup`

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Test Scenario

To get started with this basic API integration, follow the steps below.

1. Log into your BeyondTrust administrative interface and go to **Management > API Configuration**. Check the box to **Enable XML API**.
2. Create an API account and copy the client secret. This secret can be viewed only once and must be regenerated if lost.

   ```
   OAuth Client ID: e52a9aa6fc0508ddf3a40601a736b230a1bebcd1
   OAuth Client Secret: BU5u0fVEb1qEWuHdBK9AR6q9+O1CB26squ1susfJ0LsK
   ```

3. It is necessary to base64 encode these values ("Client ID:Client Secret") for use in the authorization header.

   ```
   Base64 Encoded:
   ZTUyYTlhYTZmYzA1MDhkZGYzYTQwNjAxYTczNmIyMzBhMWJlYmNkMTpCVTV1MGZWRWIxcUVXdUhkQks5QVI2cTkrTzF
   DQjI2c3F1MXN1c2ZKMExzSw==
   ```

4. We will use cURL to illustrate generating a token using a BeyondTrust API account and using that token to make requests to the BeyondTrust web API.
   a. First, we request a Bearer Token using the OAuth client ID and client secret.

      ```
      curl -H "authorization: Basic
      ZTUyYTlhYTZmYzA1MDhkZGYzYTQwNjAxYTczNmIyMzBhMWJlYmNkMTpCVTV1MGZWRWIxcUVXdUhkQks5QVI2c
      TkrTzFDQjI2c3F1MXN1c2ZKMExzSw==" --data "grant_type=client_credentials"
      https://access.example.com/oauth2/token
      ```

   b. This results in a JSON response containing the bearer token.

      ```
      {
            "access_token":"23MS6S2L42WCriESVzGbuwwsiQwdbxuAJ3Zj4DxO",
            "token_type":"Bearer",
            "expires_in":3600
      }
      ```

   c. We can now use that token to make a request to the API.

      ```
      curl -H "authorization: Bearer 23MS6S2L42WCriESVzGbuwwsiQwdbxuAJ3Zj4DxO"
      https://access.example.com/api/command?action=get_api_info
      ```

   d. This results in an XML response for the requested API.

***Note:** If you receive any errors such as **Document Not Found**, check that the API account has the necessary permissions. Also, make sure that a user is logged into the site while you are testing.*

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Privileged Remote Access API Change Log

API Version 1.19.2 for PRA 20.1.x

- Added the "Configuration API" on page 7.

API Version 1.19.0 for PRA 19.1.x

- Version update.

API Version 1.18.0 for PRA 18.2.x

- SCIM options have been added to the API Configuration.

API Version 1.16.0 for PRA 17.1.x

- Use OAuth 2.0 authentication for endpoint credential manager connections.
- When importing a Jump Item several changes have been made:
    - Specify a name for Jump Items.
    - Import VNC Jump Items.
    - Specify a SecureApp for RDP Jump Items.
    - Specify a local address for Protocol Tunnel Jump Items.
    - For Web Jump Items, set if the certificate should be verified.
    - API Command: import_jump_shortcut

API Version 1.15.1 for PRA 16.1.x

- Granularly define the accounts used for API access to the specific roles they serve. Additionally, OAuth 2.0 authentication is now used for authenticating API accounts.
    - Reporting API
    - Command API
    - Backup API

API Version 1.14.0 for PRA 15.3.x

- Import Jump Item shortcuts to minimize the time needed to create Jump Items.
    - API Command: import_jump_shortcut

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Privileged Remote Access API Version Reference

The following table shows the relationship between the API and BeyondTrust versions for BeyondTrust Privileged Remote Access.

API Version	BeyondTrust PRA Version
1.19.2	20.2.x
1.19.2	20.1.x
1.19.0	19.1.x
1.19.0	18.3.x
1.18.0	18.2.x
1.17.0	18.1.x
1.16.0	17.1.x
1.15.1	16.1.x
1.14.0	15.3.x
1.13.0	15.1.x, 15.2.x

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Appendix: Require a Ticket ID for Access to Jump Items

If your service requests use ticket IDs as part of the change management workflow, connect your ticket IDs to endpoint access in BeyondTrust. By leveraging BeyondTrust Jump Technology with your existing ticket ID process, your change management workflow integration lets you restrict a BeyondTrust access request by requiring a Ticket ID to be entered as part of the access request process before an access session begins.

What Users See

When users of the BeyondTrust access console attempt to access a Jump Item that uses a Jump Policy configured to require a ticket ID, a dialog opens. In the administrator-configured dialog, users enter the ticket ID needed, authorizing access this Jump Item.

To set up the connection to your existing ITSM or ticket ID system, create a Jump Policy you can apply to those Jump Items you want to only be used if a ticket ID from your external system is entered.

How It Works

After the user enters the required ID and clicks **OK**, the Secure Remote Access Appliance posts an HTTP outbound request to the ticket system URL configured in Jump Policies. The request contains information about both the ticket ID and the Jump Item, as well as user information. Your external system then replies asynchronously to either allow or deny access.

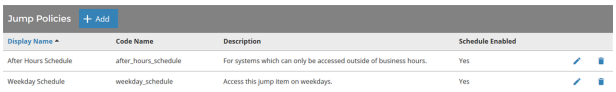
If the request is allowed, the external ticket ID system assigns the allowed session. Optionally, your external ITSM or ticket ID system may send a list of custom session attributes in its response to assign to the allowed session. For more information on using the BeyondTrust API see the Privileged Remote Access API Programmer's Guide at www.beyondtrust.com/docs/privileged-remote-access/how-to/integrations/api.

Follow the steps below to set up a ticket ID requirement for access.

Create a Jump Policy Requiring Ticket ID Approval

First, create a Jump Policy with the requirement of ticket ID approval enabled.

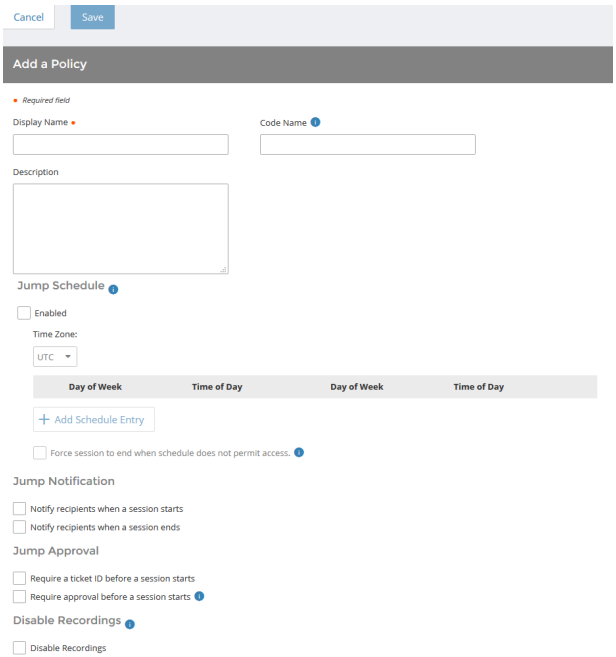
1. From your BeyondTrust /login administrative interface, go to **Jump > Jump Policies**.
2. In the **Jump Policies** section, click the **Add** button.



***Note:** A Jump Policy does not take effect until you have applied it to at least one Jump Client item.*

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

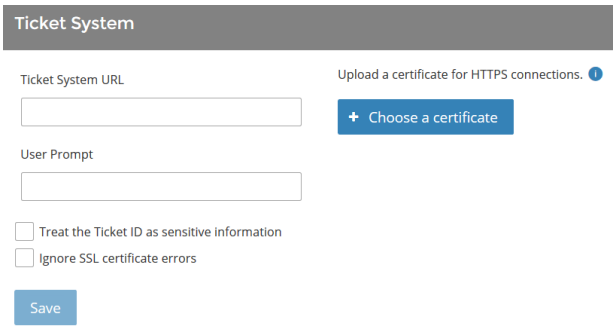
3. Enter a **Display Name**, **Code Name**, and **Description** in the corresponding locations to enable you to effectively apply this Jump Policy appropriate to your purposes after its creation.
4. Optionally, complete the configuration for **Jump Schedule** and **Jump Notification** if appropriate for the access control desired on this Jump Policy.
5. In the **Jump Approval** section, check **Require a ticket ID before a session starts**. To instantly disable ticket ID approval on this policy, simply uncheck this box. If ticket ID approval is enabled on a policy that does not have a ticket system URL configured, users attempting to access a Jump Item to which the policy is applied receive a message to contact the administrator.
6. Optionally, complete any additional approval configuration you wish this Jump Policy to enforce.
7. Click **Save**.



Connect External Ticket ID System to Jump Policies

Next, connect your existing ITSM or ticket ID system to the Secure Remote Access Appliance.

1. Remain in your BeyondTrust /login administrative interface on the **Jump > Jump Policies** page.
2. At the bottom of the **Jump Policies** page, locate the **Ticket System** section.



3. In **Ticket System URL**, enter the URL for your external ticket system. The Secure Remote Access Appliance sends an outbound request to your external ticketing system. The URL must be formatted for either HTTP or HTTPS. If an HTTPS URL is entered, the site certificate must be verified for a valid connection. If a Jump Policy requiring a ticket ID exists, a ticket system URL must be entered or you will receive a warning message.
4. The **Current Status** field is shown only when a valid status value exists to report the connection to the ticket system configured in **Ticket System URL**. Any ticket system configuration change resets the value.
5. Click **Choose a certificate** to upload the certificate for the HTTPS ticket system connection to the appliance. If your certificate is uploaded, the appliance uses it when it contacts the external system. If you do not upload a certificate and the **Ignore SSL certificate errors** box below this setting is checked, the Secure Remote Access Appliance optionally falls back to use the built-in certificate store when sending the request.

> ***Note:*** *When the **Ignore SSL certificate errors** box is checked, the Secure Remote Access Appliance will not include the certificate validation information when it contacts your external ticket system.*

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

6. In **User Prompt**, enter the dialog text you want access console users to see when they are requested to enter the ticket ID required for access.

7. If your company's security policies consider ticket ID information as sensitive material, check the **Treat the Ticket ID as sensitive information** box.

   If this box is checked, the ticket ID is considered sensitive information and asterisks are shown instead of text. You must use an HTTPS Ticket System URL. If an address with HTTP is entered, an error message appears to remind you HTTPS is required.

   When this feature is enabled you cannot bypass issues with SSL certificates by checking the **Ignore SSL certificate errors** box. This means you must have a valid SSL certificate in place. If you try to check the **Ignore SSL certificate errors** box, a message appears stating that you cannot ignore SSL certificate errors.

   When the Ticket ID is sensitive, the following rules apply:

   - Both the desktop and the web access consoles show asterisks instead of text.
   - The ticket is not logged anywhere by the access console or on the appliance.

8. Click **Save**.

API Approval Request

BeyondTrust PRA sends an HTTP Post request to the ticketing system URL. The POST request contains the following key-value pairs:

request_id	Unique ID that identifies the approval request.  ***Note:*** *The request ID must be sent from the external ticketing system to BeyondTrust PRA in the response. The maximum length is 255 characters, and the ticketing system must treat the request ID aas an opaque value.*
ticket_id	ticket ID entered by the user.
response_url	URL to which the integration should POST its reponse.
jump_item.computer_name	Hostname or IP address of the endpoint the user is requesting access for.
jump_item.type	Type of Jump Item being accessed: • client (for Jump Clients) • shell (for Shell Jump Shortcuts) • rdp • vnc • push_and_start (for Remote Jump and Local Jump) • vpro
jump_item.comments	Comments noted about the Jump Item.
jump_item.group	Group associated of the Jump Item.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

jump_item.tag	Tags associated with the Jump Item.
jump_item.jumpoint_name	Name of the Jumpoint.
jump_item.public_ip	Public IP address of the Jump Item. ***Note:*** *This is not provided for Jumpoints.*
jump_item.private_ip	Private IP address of the Jump Item. ***Note:*** *This is not provided for Jumpoints.*
jump_item.custom.<code>	Key-value pair designated for the Jump Item custom field. ***Note:*** *Only one key-value pair is permitted for each Jump Item custom field.*
user.id	The requesting user's unique ID.
user.username	Username used by the requesting user for authentication.
user.public_display_name	The requesting user's public display name.
user.private_display_name	The requesting user's private display name.
user.email_address	Email address listed for the requesting user.

API Approval Reponse

The external ticketing system sends an HTTP POST request to the Secure Remote Access Appliance URL at https://example.beyondtrust.com/api/endpoint_approval.

Note: *The API must be accessed over HTTPS.*

The POST request can contain the following key-value pairs in the POST body:

response_id	Request ID sent in the approval request. *Required
response	Response to the request; either allow or deny. *Required
message	Message displayed to the requesting user if the request is denied. *Optional ***Note:*** *The maximum length set for the message is 255 characters.*
session.custom.<code name>	One or more custom session attributes set for the access session. *Optional

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Error Messages

In certain circumstances, an error message displays in the **Ticket System** section:

- *Ticket System URL is required because one or more Jump Policies still require a ticket ID.* - A Jump Policy exists requiring the entry of a ticket ID for access.
- *Invalid ticket ID.* - The external ticket system explicitly denied the request. If the external ticket system sends the error message, that message is shown.
- *The Ticket System URL must start with "https://" when the Ticket ID is sensitive.* - You must enter an HTTPS URL when **Treat the Ticket ID as sensitive information** is checked.
- *Cannot ignore SSL errors when the Ticket ID is sensitive*. - When this option is checked, you cannot ignore SSL errors and must provide a valid SSL certificate.
- *The given host was not resolved.* - An invalid ticket system URL was attempted.
- *The ticket system failed to respond in time.* - The external ticket system failed to respond in a timely manner.

Users who are unable to connect due to misconfiguration or user error will see explanatory pop-up messages in the access console for the error state of the configuration.

- *No ticket system URL is configured. Please contact your administrator* - A ticket ID system URL is not configured in the /login administrative interface.
- *User Prompt Not Configured.* - The User Prompt is not configured in the /login administrative interface.
- *The ticket system returned an invalid response.* - An invalid ticket ID was entered.

The following errors can be returned by the Secure Remote Access Appliance:

404	Returned when no ticketing system URL is configured in /login
403	Returned when the request_id is not valid ***Note:*** *This error message is received when the request has timed out.*

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Disclaimers, Licensing Restrictions and Tech Support

Disclaimers

This document is provided for information purposes only. BeyondTrust Corporation may change the contents hereof without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. BeyondTrust Corporation specifically disclaims any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. The technologies, functionality, services, and processes described herein are subject to change without notice.

All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.

Licensing Restrictions

One BeyondTrust Privileged Remote Access license enables one support representative at a time to troubleshoot an unlimited number of remote computers, whether attended or unattended. Although multiple accounts may exist on the same license, two or more licenses (one per concurrent support representative) are required to enable multiple support representatives to troubleshoot simultaneously.

One BeyondTrust Privileged Remote Access license enables access to one endpoint system. Although this license may be transferred from one system to another if access is no longer required to the first system, two or more licenses (one per endpoint) are required to enable access to multiple endpoints simultaneously.

Tech Support

At BeyondTrust, we are committed to offering the highest quality service by ensuring that our customers have everything they need to operate with maximum productivity. Should you need any assistance, please contact BeyondTrust Technical Support at www.beyondtrust.com/support.

Technical support is provided with annual purchase of our maintenance plan.

©2003-2021 BeyondTrust Corporation. All Rights Reserved. Other trademarks identified on this page are owned by their respective owners. BeyondTrust is not a chartered bank or trust company, or depository institution. It is not authorized to accept deposits or trust accounts and is not licensed or regulated by any state or federal banking authority.