
Carbon Black Cloud Python API Documentation

Release 1.5.1

Carbon Black Developer Network

Jan 30, 2024

USER GUIDE

1	Major Features	3
2	Audience for the SDK	5
3	API Credentials	7
4	Getting Started	9
4.1	Installation	9
4.2	Authentication	11
4.3	Getting Started with the Carbon Black Cloud Python SDK - “Hello CBC”	21
4.4	Resources	24
4.5	Guides	24
4.6	The CBCloudAPI Object	116
4.7	Audit and Remediation Package	127
4.8	Credential Providers Package	168
4.9	Endpoint Standard Package	172
4.10	Enterprise EDR Package	204
4.11	Platform Package	245
4.12	Workload Package	515
4.13	CBC SDK Package	553
4.14	Logging & Diagnostics	622
4.15	Testing	623
4.16	Changelog	625
4.17	Exceptions	635
5	Indices and tables	637
	Python Module Index	639
	Index	641

Release v1.5.1.

The Carbon Black Cloud Python SDK provides an easy interface to connect with Carbon Black Cloud products, including Endpoint Standard, Audit and Remediation, and Enterprise EDR. Use this SDK to more easily query and manage your endpoints, manipulate data as Python objects, and harness the full power of Carbon Black Cloud APIs.

MAJOR FEATURES

- **Supports the following Carbon Black Cloud Products with extensions for new features and products planned**

- Endpoint Standard
- Audit and Remediation
- Enterprise EDR
- Platform
- Workload

- **Reduced Complexity**

The SDK manages the differences among Carbon Black Cloud APIs behind a single, consistent Python interface. Spend less time learning specific API calls, and more time controlling your environment.

- **More Efficient Performance**

A built-in caching layer makes repeated access to the same resource more efficient. Instead of making identical API requests repeatedly, the SDK caches the results of the request the first time, and references the cache when you make future requests for the resource. This reduces the time required to access the resource later.

AUDIENCE FOR THE SDK

In general, the Carbon Black Cloud Python SDK is directed at those that:

- Have a working knowledge of Python.
- Have a basic understanding of what the Carbon Black Cloud does, and its basic terminology such as events, alerts, and watchlists.

API CREDENTIALS

To use the SDK and access data in Carbon Black Cloud, you must set up API keys with the correct permissions if you are using the X-Auth-Token authentication method, or create an access token if you are using Bearer or Personal API Token. Different APIs have different permission requirements for use, which is explained in the [Developer Network Authentication Guide](#).

The SDK manages your API credentials for you. There are multiple ways to supply the SDK with your API credentials, which is explained in [Authentication](#).

GETTING STARTED

Get started with Carbon Black Cloud Python SDK [here](#). For detailed information on the objects and methods exposed by Carbon Black Cloud Python SDK, see the full SDK Documentation below.

4.1 Installation

If you already have Python installed, skip to [Use Pip](#).

4.1.1 Install Python

Carbon Black Cloud Python SDK is compatible with Python 3.8+. UNIX systems usually have Python installed by default; it will have to be installed on Windows systems separately.

If you believe you have Python installed already, run the following two commands at a command prompt:

```
$ python --version
Python 3.8.16

$ pip --version
pip 20.2.3 from /usr/local/lib/python3.8/site-packages (python 3.8)
```

If `python --version` reports back a version of 3.8.x or higher, you're all set. If `pip` is not found, follow the instructions on this [guide](#).

Note: On many UNIX/Linux environments, the `python` and `pip` commands invoke Python version 2, for backwards compatibility. Python 2 is not compatible with the Carbon Black Cloud Python SDK. Python version 3 is invoked via the commands `python3` and `pip3`. Use these commands in this installation guide in place of `python` and `pip`.

If you're on Windows, and Python is not installed yet, download the [latest Python installer](#) from [python.org](#).



Ensure that the “Add Python to PATH” option is checked.

4.1.2 Use Pip

Once python and pip are installed, open a command prompt and type:

```
$ pip install carbon-black-cloud-sdk
```

This will download and install the latest version of the SDK from the Python PyPI packaging server.

Note: In Python environments that implement [PEP 668](#) and declare their global packages to be “externally managed,” the use of `pip` to install packages outside a virtual environment is no longer supported, unless overridden by a command-line option to `pip` (such as `--break-system-packages`). For the use of virtual environments, see the next section and the [Python virtual environment guide](#).

4.1.3 Virtual Environments (optional)

If you are installing the SDK with the intent to contribute to it’s development, it is recommended that you use virtual environments to manage multiple installations.

A virtual environment is a Python environment such that the Python interpreter, libraries and scripts installed into it are isolated from those installed in other virtual environments, and (by default) any libraries installed in a “system” Python, i.e., one which is installed as part of your operating system¹.

See the [python.org virtual environment guide](https://docs.python.org/3/library/venv.html) for more information.

¹ <https://docs.python.org/3/library/venv.html>

4.1.4 Get Source Code

Carbon Black Cloud Python SDK is actively developed on GitHub and the code is available from the [Carbon Black GitHub repository](#). The version of the SDK on GitHub reflects the latest development version.

To clone the latest version of the SDK repository from GitHub:

```
$ git clone git@github.com:carbonblack/carbon-black-cloud-sdk-python.git
```

Once you have a copy of the source, you can install it in “development” mode into your Python site-packages directory:

```
$ cd carbon-black-cloud-sdk-python
$ python setup.py develop
```

This will link the version of carbon-black-cloud-sdk-python you cloned into your Python site-packages directory. Any changes you make to the cloned version of the SDK will be reflected in your local Python installation. This is a good choice if you are thinking of changing or further developing carbon-black-cloud-sdk-python.

4.2 Authentication

Carbon Black Cloud APIs require authentication to secure your data.

There are several methods for authentication listed below. Every method requires one of the following type of credentials X-Auth-Token, OAuth App with Bearer or Personal API Token. See the [Developer Network Authentication Guide](#) to learn how to generate the type of credentials your implementation uses.

The SDK only uses one Authentication method at a time. It is recommended to create Authentication Methods for specific actions, and use them as needed.

For example, if using the [Devices API](#) to search for mission critical devices, and the [Live Response API](#) to execute commands on those devices, generate one API credential with appropriate permissions and access level. Store the credential with a profile name, and reference the profile when creating CBCloudAPI objects.

Example contents of credentials.cbc file used for authentication with X-Auth-Token. Read more about the credentials.cbc below.

```
[platform]
url=https://defense-prod05.conferdeploy.net
token=ABCDEFGHIIJKLMNO123456789/ABCD123456
org_key=ABCD123456
ssl_verify=false
ssl_verify_hostname=no
```

Example code authentication with a profile named “platform”

```
# import relevant modules
>>> from cbc_sdk.platform import Device
>>> from cbc_sdk import CBCloudAPI

# create Platform API object
>>> platform_api = CBCloudAPI(profile='platform')

# search for specific devices with Platform Devices API
>>> important_devs = platform_api.select(Device).set_target_priorities(["MISSION_CRITICAL
```

(continues on next page)

(continued from previous page)

```

→"}])

# execute commands with Live Response API
>>> for device in important_devs:
...     lr_session = platform_api.live_response.request_session(device.id)
...     lr_session.create_process(r'cmd.exe /c "ping.exe 192.168.1.1"')
...     lr_session.close()

```

For more examples on Live Response, check [Live Response](#)

4.2.1 Authentication Methods

With a File:

Credentials may be stored in a `credentials.cbc` file. With support for multiple profiles, this method makes it easy to manage multiple API Keys for different products and permission levels.

```
>>> cbc_api = CBCloudAPI('~/.carbonblack/myfile.cbc', profile='default')
```

With Windows Registry:

Windows Registry is a secure option for storing API credentials on Windows systems.

```
>>> provider = RegistryCredentialProvider()
>>> cbc_api = CBCloudAPI(credential_provider=provider, profile='default')
```

With macOS's Keychain Access:

The Keychain Access which is built into macOS can also be used for authentication.

```
>>> provider = KeychainCredentialProvider('CBC API Credentials', 'default')
>>> cbc_api = CBCloudAPI(credential_provider=provider)
```

With Amazon Secrets Manger:

There is a support for the Amazon Secrets Manager, navigate to the section for further details of how to set it up.

```
>>> provider = AWSCredentialProvider(secret_arn='your-secret-arn-string')
>>> cbc_api = CBCloudAPI(credential_provider=provider)
```

With an External Credential Provider:

Credential Providers allow for custom methods of loading API credentials. This method requires you to write your own Credential Provider.

```
>>> provider = MyCredentialProvider()
>>> cbc_api = CBCloudAPI(credential_provider=provider, profile='default')
```

Not Recommended:

At Runtime:

Credentials may be passed into `CBCloudAPI()` via keyword parameters. This method should be used with caution, taking care to not share your API credentials when managing code with source control.


```
>>> cbc_api = CBCloudAPI(url='https://defense.conferdeploy.net', token='ABCD/
↪1234',
...     org_key='ABCDEFGH')
```

Not Recommended:

With Environmental Variables:

Environmental variables can be used for authentication, but pose a security risk. This method is not recommended unless absolutely necessary.

With a File

Credentials may be supplied in a file that resembles a Windows .INI file in structure, which allows for multiple “profiles” or sets of credentials to be supplied in a single file. The file format is backwards compatible with CBAPI, so older files can continue to be used.

Example of a credentials file containing two profiles

```
[default]
url=http://example.com
token=ABCDEFGH IJKLMNOPQRSTU VWX/12345678
org_key=A1B2C3D4
ssl_verify=false

[production]
url=http://example.com
token=QRSTUVWXYZABCDEFGHIJKLMN/76543210
org_key=A1B2C3D4
ssl_verify=false
ssl_verify_hostname=no
ssl_cert_file=foo.certs
ssl_force_tls_1_2=1
proxy=proxy.example
ignore_system_proxy=on
integration=MyApplication/1.3.1
```

Common fields between all types of credentials

Keyword	Default	Required
url		Yes
org_key		Yes
ssl_verify	1	No
ssl_verify_hostname	1	No
ignore_system_proxy	0	No
ssl_force_tls_1_2	0	No
ssl_cert_file		No
proxy		No
integration		No
default_timeout	300000	No

X-AUTH-TOKEN specific fields

Keyword	Default	Required
token		Yes

OAuth App with Bearer specific fields

Keyword	Default	Required
csp_oauth_app_id		Yes
csp_oauth_app_secret		Yes

Personal API Token specific fields

Keyword	Default	Required
csp_api_token		Yes

Individual profiles or sections are delimited in the file by placing their name within square brackets: [profile_name]. Within each section, individual credential values are supplied in a keyword=value format.

Unrecognized keywords are ignored.

By default, the CBC SDK looks for credentials files in the following locations:

- The .carbonblack subdirectory of the current directory of the running process.
- The .carbonblack subdirectory of the user's home directory.
- The /etc/carbonblack subdirectory on Unix, or the C:\Windows\carbonblack subdirectory on Windows.

Within each of these directories, the SDK first looks for the credentials.cbc file, then the credentials.psc file (the older name for the credentials file under CBAPI).

You can override the file search logic and specify the full pathname of the credentials file in the keyword parameter credential_file when creating the *CBCloudAPI* object.

In all cases, you will have to specify the name of the profile to be retrieved from the credentials file in the keyword parameter profile when creating the *CBCloudAPI* object.

Example:

```
>>> cbc_api = CBCloudAPI(credential_file='~/.carbonblack/myfile.cbc', profile='default')
```

Note on File Security: It is recommended that the credentials file be secured properly on Unix. It should be owned by the user running the process, as should the directory containing it, and neither one should specify any file permissions for “group” or “other.” In numeric terms, that means the file should have 400 or 600 permissions, and its containing directory should have 500 or 700 permissions. This is similar to securing configuration or key files for ssh. If these permissions are incorrect, a warning message will be logged; a future version of the CBC SDK will disallow access to files altogether if they do not have the correct permissions.

Credential files *cannot* be properly secured in this manner under Windows; if they are used in that environment, a warning message will be logged.

With Windows Registry

CBC SDK also provides the ability to use the Windows Registry to supply credentials, a method which is more secure on Windows than other methods.

N.B.: Presently, to use the Windows Registry, you must supply its credential provider as an “external” credential provider. A future version of the CBC SDK will move to using this as a default provider when running on Windows.

By default, registry entries are stored under the key `HKEY_CURRENT_USER\Software\VMware Carbon Black\Cloud Credentials`. Under this key, there may be multiple subkeys, each of which specifies a “profile” (as with credential files). Within these subkeys, the following named values may be specified:

Common fields between all types of credentials

Keyword	Value Type	Default	Required
<code>url</code>	<code>REG_SZ</code>		Yes
<code>org_key</code>	<code>REG_SZ</code>		Yes
<code>ssl_verify</code>	<code>REG_DWORD</code>	1	No
<code>ssl_verify_hostname</code>	<code>REG_DWORD</code>	1	No
<code>ignore_system_proxy</code>	<code>REG_DWORD</code>	0	No
<code>ssl_force_tls_1_2</code>	<code>REG_DWORD</code>	0	No
<code>ssl_cert_file</code>	<code>REG_SZ</code>		No
<code>proxy</code>	<code>REG_SZ</code>		No
<code>integration</code>	<code>REG_SZ</code>		No
<code>default_timeout</code>	<code>REG_DWORD</code>	300000	No

X-AUTH-TOKEN specific fields

Keyword	Value Type	Default	Required
<code>token</code>	<code>REG_SZ</code>		Yes

OAuth App with Bearer specific fields

Keyword	Value Type	Default	Required
<code>csp_oauth_app_id</code>	<code>REG_SZ</code>		Yes
<code>csp_oauth_app_secret</code>	<code>REG_SZ</code>		Yes

Personal API Token specific fields

Keyword	Value Type	Default	Required
<code>csp_api_token</code>	<code>REG_SZ</code>		Yes

Unrecognized named values are ignored.

To use the Registry credential provider, create an instance of it, then pass the reference to that instance in the `credential_provider` keyword parameter when creating [CBCloudAPI](#). As with credential files, the name of the profile to be retrieved from the Registry should be specified in the keyword parameter `profile`.

Example:

```
>>> provider = RegistryCredentialProvider()
>>> cbc_api = CBCloudAPI(credential_provider=provider, profile='default')
```

Advanced Usage: The parameters `keypath` and `userkey` to `RegistryCredentialProvider` may be used to control the exact location of the “base” registry key where the sections of credentials are located. The `keypath` parameter allows specification of the path from `HKEY_CURRENT_USER` where the base registry key is located. If `userkey`, which is `True` by default, is `False`, the path will be interpreted as being rooted at `HKEY_LOCAL_MACHINE` rather than `HKEY_CURRENT_USER`.

Example:

```
>>> provider = RegistryCredentialProvider('Software\\Contoso\\My CBC Application')
>>> cbc_api = CBCloudAPI(credential_provider=provider, profile='default')
```

Note the use of doubled backslashes to properly escape them under Python.

With an External Credential Provider

Credentials may also be supplied by writing a class that conforms to the `CredentialProvider` interface protocol. When creating `CBCloudAPI`, pass a reference to a `CredentialProvider` object in the `credential_provider` keyword parameter. Then pass the name of the profile you want to retrieve from the provider object using the keyword parameter `profile`.

Example:

```
>>> provider = MyCredentialProvider()
>>> cbc_api = CBCloudAPI(credential_provider=provider, profile='default')
```

Details of writing a credential provider may be found in the *Developing a Custom Credential Provider* document.

At Runtime

The credentials may be passed into the `CBCloudAPI` object when it is created via the keyword parameters `url`, `token`, `org_key`, and (optionally) `ssl_verify` and `integration_name`.

Example:

```
>>> api = CBCloudAPI(url='https://example.com', token='ABCDEFGHijklmnopqrstuvwx/12345678
↳ ',
...                 org_key='A1B2C3D4', ssl_verify=False, integration_name='MyScript/1.0
↳ ')
```

The `integration_name` may be specified even if using another credential provider. If specified as a parameter, this overrides any integration name specified by means of the credential provider.

With Environmental Variables

The credentials may be supplied to CBC SDK via the environment variables `CBC_URL`, `CBC_TOKEN`, `CBC_ORG_KEY`, and `CBC_SSL_VERIFY`. For backwards compatibility with `CBAPI`, the environment variables `CBAPI_URL`, `CBAPI_TOKEN`, `CBAPI_ORG_KEY`, and `CBAPI_SSL_VERIFY` may also be used; if both are specified, the newer `CBC_xxx` environment variables override their corresponding `CBAPI_xxx` equivalents. To use the environment variables, they must be set before the application is run (at least `CBC_URL` or `CBAPI_URL`, and `CBC_TOKEN` or `CBAPI_TOKEN`), and the `credential_file` keyword parameter to `CBCloudAPI` must be either `None` or left unspecified. (The `profile` keyword parameter will be ignored.)

N.B.: Passing credentials via the environment can be insecure, and, if this method is used, a warning message to that effect will be generated in the log.

With macOS's Keychain Access

The SDK also supports the usage of macOS's Keychain Access. It works in a similar manner as our other authentication methods. Keychain Access is a key-value based password storage and since we have more than one key-value based entry we are going to use JSON to store our other entries, the JSON is going to be stored under the password value.

Note: You can start first by creating the JSON object, you can do that by using our CLI tool(`<SDK_ROOT>/bin/set-macos-keychain.py`) or by manually creating it. The tool can:

- Automatically import all of your profiles set in the `credentials.cbc` file. Or by setting a custom path to a file.
- Manually input the values of your credentials via prompt or by using system arguments.

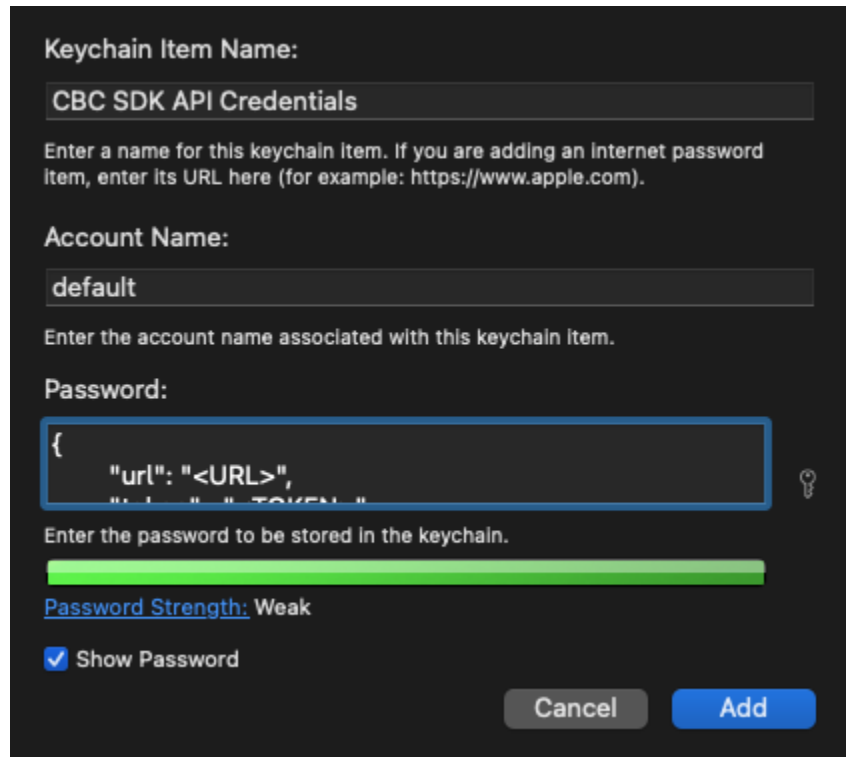
Find out how to use the script in its docstring or by using `--help`.

You can remove the keys that you won't be using or leave them empty. Reference our Explanation of API Credential Components.

```
{
  "url": "<URL>",
  "token" : "<TOKEN>",
  "org_key": "<ORG_KEY>",
  "ssl_verify": true,
  "ssl_verify_hostname": true,
  "ssl_cert_file": "<FILE_PATH>",
  "ssl_force_tls_1_2": true,
  "proxy": "<NAME_OF_THE_PROXY_HOST>",
  "ignore_system_proxy": true,
  "integration": "<INTEGRATION_NAME>",
  "default_timeout": 300000
}
```

Note: When you are storing a JSON object under the password's input in Keychain it is possible to see only the `{` in the input field, you can navigate with the arrows to check if the rest of the JSON is there.

Then we can move to storing that entry into the Keychain, create a new entry which looks like that:

A macOS Keychain Item dialog box with a dark background. It has three text input fields: 'Keychain Item Name:' with 'CBC SDK API Credentials', 'Account Name:' with 'default', and 'Password:' with a JSON string. Below the password field is a green strength indicator bar and the text 'Password Strength: Weak'. There is a checkbox for 'Show Password' which is checked. At the bottom right are 'Cancel' and 'Add' buttons.

Keychain Item Name:

CBC SDK API Credentials

Enter a name for this keychain item. If you are adding an Internet password item, enter its URL here (for example: https://www.apple.com).

Account Name:

default

Enter the account name associated with this keychain item.

Password:

{
 "url": "<URL>,"
 "auth": "TOKEN"

Enter the password to be stored in the keychain.

Password Strength: Weak

☒ Show Password

Cancel Add

After we've set the entry in the Keychain Access we can now authenticate our SDK using the `KeychainCredentialProvider`.

```
>>> from cbc_sdk.credential_providers import KeychainCredentialProvider
>>> provider = KeychainCredentialProvider('CBC API Credentials', 'default')
>>> cbc_api = CBCCloudAPI(credential_provider=provider)
```

You will be prompted to type your password so that python can access the keychain in order to obtain the credentials.

4.2.2 With Amazon Secrets Manger

Configure the AWS credentials

A full and comprehensive guide configuring the files and credentials regarding AWS can be found in their [official documentation](#).

Adding a secret to the AWS Secrets Manager

There is an [official guide](#) for creating a secret by AWS.

Note: Add your secrets as a key/value pairs. In the Explanation of API Credential Components you can find full information on required fields and their purpose.

Using our credential provider for the SDK

After the configuration of the AWS Credentials and storing your secret in the AWS Secret Manager, we can start using the credential provider.

```
>>> from cbc_sdk.credential_providers import AWSCredentialProvider
>>> from cbc_sdk import CBCloudAPI
>>> provider = AWSCredentialProvider(secret_arn='your-secret-arn-string')
>>> cbc_api = CBCloudAPI(credential_provider=provider)
```

AWS Single Sign-On Provider (SSO)

If you wish to set the SSO provider follow this [tutorial](#) for setting the config.

Then you can use the `profile_name` attribute in the `AWSCredentialProvider` like so:

```
>>> from cbc_sdk.credential_providers import AWSCredentialProvider
>>> from cbc_sdk import CBCloudAPI
>>> provider = AWSCredentialProvider(secret_arn='your-secret-arn-string', profile_name=
↳ "my-sso-profile")
>>> cbc_api = CBCloudAPI(credential_provider=provider)
```

4.2.3 Explanation of API Credential Components

When supplying API credentials to the SDK at runtime, with a file, or with Windows Registry, the credentials include these components:

Common fields between X-Auth-Token, OAuth App with Bearer and Personal API Token authentication methods

Key-word	Definition	De-fault	Re-quired
<code>url</code>	The URL used to access the Carbon Black Cloud.		Yes
<code>org_key</code>	The organization key specifying which organization to work with.		Yes
<code>ssl_veri:</code>	A Boolean value (see below) indicating whether or not to validate the SSL connection.	True	No
<code>ssl_veri:</code>	A Boolean value (see below) indicating whether or not to verify the host name of the server being connected to.	True	No
<code>ignore_s:</code>	A Boolean value (see below). If this is True, any system proxy settings will be ignored in making the connection to the server.	Fals	No
<code>ssl_forc:</code>	A Boolean value (see below). If this is True, the connection will be forced to use TLS 1.2 rather than any later version.	Fals	No
<code>ssl_cert:</code>	The name of an optional certificate file used to validate the certificates of the SSL connection. If not specified, the standard system certificate verification will be used.		No
<code>proxy</code>	If specified, this is the name of a proxy host to be used in making the connection.		No
<code>integrat:</code>	The name of the integration to use these credentials. The string may optionally end with a slash character, followed by the integration's version number. Passed as part of the <code>User-Agent: HTTP</code> header on all requests made by the SDK.		No
<code>default_t_:</code>	The default timeout for search queries, specified in milliseconds. This value may never be greater than the default of 300000 milliseconds.	30000	No

X-AUTH-TOKEN specific fields

Key-word	Definition	Re-quired
token	The access token to authenticate with. Same structure as X-Auth-Token defined in the Developer Network Authentication Guide . Derived from an API Key's Secret Key and API ID.	Yes

OAuth App with Bearer specific fields

Keyword	Definition	Re-quired
csp_oauth_app_id	Client ID, enter the Client ID that you set in Create OAuth 2.0 Client.	Yes
csp_oauth_app_secret	Client Secret, enter the secret that was generated in Create OAuth 2.0 Client.	Yes

Personal API Token specific fields

Keyword	Definition	Re-quired
csp_api_t	API tokens are issued by users in an organization and are associated with the user's account and the organization from which they generated the API token.	Yes

When supplying API credentials to the SDK with environmental variables, the credentials include these components:

Keyword	Legacy	Default
CBC_URL	CBAPI_URL	
CBC_TOKEN	CBAPI_TOKEN	
CBC_ORG_KEY	CBAPI_ORG_KEY	
CBC_SSL_VERIFY	CBAPI_SSL_VERIFY	True

Alternative keywords are available to maintain backwards compatibility with CBAPI.

Boolean Values

Boolean values are specified by using the strings `true`, `yes`, `on`, or `1` to represent a `True` value, or the strings `false`, `no`, `off`, or `0` to represent a `False` value. All of these are case-insensitive. Any other string value specified will result in an error.

For example, to disable SSL connection validation, any of the following would work:

```
ssl_verify=False
ssl_verify=false
ssl_verify=No
ssl_verify=no
ssl_verify=Off
ssl_verify=off
ssl_verify=0
```


4.3 Getting Started with the Carbon Black Cloud Python SDK - “Hello CBC”

This document will help you get started with the Carbon Black Cloud Python SDK by installing it, configuring authentication for it, and executing a simple example program that makes one API call.

4.3.1 Installation

Make sure you are using Python 3. Use the command `pip install carbon-black-cloud-sdk` to install the SDK and all its dependencies. (In some environments, the correct command will be `pip3 install carbon-black-cloud-sdk` to use Python 3.)

You can also access the SDK in development mode by cloning the GitHub repository, and then executing `python setup.py develop` (in some environments, `python3 setup.py develop`) from the top-level directory. Setting your `PYTHONPATH` environment variable to the directory `[sdk]/src`, where `[sdk]` is the top-level directory of the SDK, will also work for these purposes. (On Windows, use `[sdk]\src`.)

See also the [Installation](#) section of this documentation for more information.

4.3.2 Authentication

To make use of APIs, you will need an *API token*, in case you are using Carbon Black Cloud to manage your identity and authentication, or if you are using VMware Cloud Services Platform, an *OAuth App with Bearer* or a *Personal API Token*. For our example, we will use a custom CBC-managed key with the ability to list devices. To learn more about the different authentication methods, click [here](#).

Log into the Carbon Black Cloud UI and go to **Settings > API Access**. Start by selecting **Access Levels** at the top of the screen and press **Add Access Level**. Fill in a name and description for your sample access level, keep **Copy permissions** from set to **None**, and, under the permission category **Device** and permission name **General information**, check the **Read** check box. Press **Save** to save and create the new access level.

Now select **API Keys** at the top of the screen and press **Add API Key**. Enter a name for the key, and, optionally, a description. For **Access Level type**, select **Custom**, and for **Custom Access Level**, select the access level you created above. Press **Save** to save and create the new API key. An **API Credentials** dialog will be displayed with the new API ID and secret key; this dialog may also be re-displayed at any time by finding the API key in the list, clicking the drop-down arrow under the **Actions** column, and selecting **API Credentials**.

We will use a credentials file to store the credential information by default. Create a directory named `.carbonblack` under your user home directory. (On Windows, this directory is generally `C:\Users\[username]`, where `[username]` is your user name.) Within this directory create a file `credentials.cbc` to store your credentials. Copy the following template to this new file:

```
[default]
url=
token=
org_key=
ssl_verify=True
integrationName=CustomSDKScript/1.0
```

Following the `url=` keyword, add the top-level URL you use to access the Carbon Black Cloud, including the `https://` prefix and the domain name, but without any of the path information following it.

Following the `token=` keyword, add the API Secret Key from the API Credentials dialog, followed by a forward slash (/) character, followed by the API ID from the API Credentials dialog. (The secret key is always 24 characters in length, and the API ID is always 10 characters in length.)

Following the `org_key=` keyword, add the organization key from your organization, which may be seen under the Org Key: heading at the top of the API Keys display under Settings > API Access. It is always 8 characters in length.

Save the completed `credentials.cbc` file, which should look like this (*example text only*):

```
[default]
url=https://example.net
token=ABCDEFGHGIJKLMNOPQRSTUWVX/ABCDEFGHJI
org_key=A1B2C3D4
ssl_verify=True
```

On UNIX systems, you must make sure that the `credentials.cbc` file is properly secured. The simplest commands for doing so are:

```
$ chmod 600 ~/.carbonblack/credentials.cbc
$ chmod 700 ~/.carbonblack
```

For further information, please see the [Authentication](#) section of the documentation, as well as the [Authentication Guide](#) on the Carbon Black Cloud Developer Network.

4.3.3 Setting the User-Agent

The SDK supports custom User-Agent`s, which allow you to identify yourself when using the SDK to make API calls. The credential parameter `integration_name` is used for this. If you use a file to authenticate the SDK, this is how you could identify yourself:

```
[default]
url=http://example.com
token=ABCDEFGHGIJKLMNOPQRSTUWVX/12345678
org_key=A1B2C3D4
integration_name=MyScript/0.9.0
```

See the [Authentication](#) documentation for more information about credentials.

4.3.4 Running the Example

The example we will be running is `list_devices.py`, located in the `examples/platform` subdirectory of the GitHub repository. If you cloned the repository, change directory to `[sdk]/examples/platform`, where `[sdk]` is the top-level directory of the SDK. (On Windows, use `[sdk]\examples\platform`.) Alternately, you may view the current version of that script in “raw” mode in GitHub, and use your browser’s Save As function to save the script locally. In that case, change directory to whichever directory you saved the script to.

Execute the script by using the command `python list_devices.py` (in some environments, `python3 list_devices.py`). If all is well, you will see a list of devices (endpoints) registered in your organization, showing their numeric ID, host name, IP address, and last checkin time.

You can change what devices are shown by adding a query value with the `-q` parameter, and also by using additional parameters to modify the search criteria. Execute the command `python list_devices.py --help` (in some environments, `python3 list_devices.py --help`) for a list of all possible command line parameters.

4.3.5 Inside the Example Script

Once the command-line arguments are parsed, we create a Carbon Black Cloud API object with a call to the helper function `get_cb_cloud_object()`. The standard `select()` method is used to create a query object that queries for devices; the query string is passed to that object via the `where()` method, and other criteria are added using specific setters.

The query is an iterable object, and calling upon its iterator methods invokes the query, which, in this case, is the [Search Devices](#) API. The example script turns those results into an in-memory list, then iterates on that list, printing only certain properties of each retrieved Device object.

4.3.6 Calling the SDK Directly

Now we'll repeat this example, but using the Python command line directly without a script.

Access your Python interpreter with the `python` command (or `python3` if required) and type:

```
>>> from cbc_sdk.rest_api import CBCloudAPI
>>> from cbc_sdk.platform import Device
>>> cb = CBCloudAPI(profile='default')
```

This imports the necessary classes and creates an instance of the base `CBCloudAPI` object. By default, the file credentials provider is used. We set it to use the default profile in your `credentials.cbc` file, which you set up earlier.

Note: On Windows, a security warning message will be generated about file access to CBC SDK credentials being inherently insecure.

This creates a query object that searches for all devices:

```
>>> query = cb.select(Device)
```

For convenience, we load the entirety of the query results into an in-memory list:

```
>>> devices = list(query)
```

Using a simple `for` loop, we print out the ID, host name, internal IP address, and last contact time from each returned device. Note that the contents of the list are `Device` objects, not dictionaries, so we access individual properties with the `object.property_name` syntax, rather than `object['property_name']`:

```
>>> for device in devices:
...     print(device.id, device.name, device.last_internal_ip_address, device.last_
...           ↪contact_time)
... 
```

Searching is an important operation in the SDK, as that is how objects are generally retrieved for other operations. The [Guide to Searching](#) contains more information about searching.

4.3.7 Next Steps

- *Guides*: Information and Examples related to specific actions you want to take on your Carbon Black Cloud data

4.4 Resources

Here you can find examples, recorded demonstrations, and other resources we think will be useful to get the most out of the Carbon Black Cloud Python SDK.

4.4.1 Audience for These Resources

In general, and unless otherwise indicated, these guides are directed at those that:

- Have a working knowledge of Python.
- Have a basic understanding of what the Carbon Black Cloud does, and its basic terminology such as events, alerts, and watchlists.

Certain guides may be more geared towards audiences with more experience with the Carbon Black Cloud, such as administrators.

4.4.2 Examples

The [GitHub repository](#) also has some example scripts which will help you get started using the SDK.

4.4.3 Recordings

Demonstrations are found on our [YouTube channel](#).

A recent highlight shows how to schedule Audit and Remediation Tasks.

4.5 Guides

Here we've listed a collection of tutorials, recorded demonstrations and other resources we think will be useful to get the most out of the Carbon Black Cloud Python SDK.

4.5.1 Audience for These Guides

In general, and unless otherwise indicated, these guides are directed at those that:

- Have a working knowledge of Python.
- Have a basic understanding of what the Carbon Black Cloud does, and its basic terminology such as events, alerts, and watchlists.
- Need information to update to new versions of the SDK when enhanced features are released.

Certain guides may be more geared towards audiences with more experience with the Carbon Black Cloud, such as administrators.

Information about updating to new versions of the SDK to take advantage of new features in Carbon Black Cloud are in *Migration Guides*.

4.5.2 Feature Guides

Searching

Almost every interaction with the Carbon Black Cloud SDK will involve searching for some object on the server that your code can then inspect or operate on. Searching in the SDK involves three steps:

1. Create a *query object* with the `select()` method.
2. Refine the query by using the query object's methods to add a text *query* and/or *search criteria*.
3. Execute the query to see its results.

Creating a Query Object

A query object is created via the `CBCloudAPI.select()` operation, specifying the type of data to be retrieved.

In this example, we create a query to search for all devices with antivirus active:

```
# assume the CBCloudAPI object is in the variable "api"
>>> from cbc_sdk.platform import Device
>>> device_query = api.select(Device).where('status:ACTIVE')

# The device query has been created but not yet executed
>>> type(device_query)
<class 'cbc_sdk.platform.devices.DeviceSearchQuery'>
```

The `select()` method may take either a class reference or a string class name:

```
>>> query1 = api.select(Device)
>>> query2 = api.select("Device")

# prove that the query we get back in either case is the same
>>> type(query1) == type(query2)
True
```

Selecting an Object Directly

The `select()` method may also be used to retrieve an object directly if you know its ID value, by passing the ID as a second parameter:

```
>>> dev = api.select(Device, 1234567) # assume this device exists
>>> type(dev)
<class 'cbc_sdk.platform.devices.Device'>
```

Refining a Query

Queries may support one of two different methods for refining a query:

- Through the use of *text query*.
- Through adding *criteria*.

Text Query Support

Text queries may be added to a query object by using the query object's `where()`, `and_()`, and `or_()` methods. The following example sets up a query looking for events in which the program `googleupdate.exe` accesses the system registry on a device with a specific hostname, IP address, and owner:

```
# assume the CBCloudAPI object is in the variable "api"
>>> from cbc_sdk.platform import Observation
>>> obs_query = api.select(Observation).where(process_name='svchost.exe').and_
↳ (observation_type='CONTEXTUAL_ACTIVITY')

# further refine the query
>>> obs_query.and_(event_type='netconn')
>>> obs_query.and_(netconn_protocol='PROTO_TCP').and_(netconn_port=80)
```

The `where()` method supplies the initial query parameters, while `and_()` and `or_()` add additional query parameters. As with other languages, `and_()` gets grouped together before `or_()`.

Parameters may either be supplied as text strings or as keyword assignments:

```
>>> from cbc_sdk.platform import Device
# the following two queries are equivalent
>>> string_query = api.select(Device).where("status:ACTIVE")
>>> keyword_query = api.select(Device).where(status="ACTIVE")
```

However, mixing the two types in a single query is not allowed:

```
# this is not allowed
>>> from cbc_sdk.platform import Device
>>> bogus_query = api.select(Device).where(status="ACTIVE").and_("virtualMachine:true")
cbc_sdk.errors.ApiError: Cannot modify a structured query with a raw parameter
```

Criteria Support

Criteria are usually added to queries using methods specific to each query. For example, this query looks for alerts with severity 9 or 10 on a machine running macOS 10.14.6:

```
>>> from cbc_sdk.platform import Alert
>>> alert_query = api.select(Alert)

# Refine the query with parameters
>>> alert_query.where(alert_severity=9).or_(alert_severity=10)

# Refine the query with criteria
>>> alert_query.set_device_os(["MAC"]).set_device_os_versions(["10.14.6"])
```

This query produces the following JSON block to be passed to a POST request to the server:

```
{
  "query": "alert_severity:9 OR alert_severity:10",
  "criteria": {
    "device_os": ["MAC"],
    "device_os_version": ["10.14.6"]
  }
}
```

In newer queries, the various specific methods for setting each individual criterion will be replaced with a single method:

```
# Refine the query with criteria (new style)
>>> alert_query.add_criteria("device_os", ["MAC"]).add_criteria("device_os_version", [
    ↪ "10.14.6"])
```

Note: The `add_criteria()` method is explicitly supported with Alerts v7, as well as other query classes that make use of `CriteriaBuilderSupportMixin`. Over time, the existing “specific” methods for setting criteria will be deprecated.

Certain queries accept a *time range* criterion, set with the `set_time_range()` method. This allows a range of times to be specified which returned objects must fall within. Parameters for `set_time_range()` are as follows:

- `start`: Specifies the starting time of the range, in ISO 8601 format.
- `end`: Specifies the ending time of the range, in ISO 8601 format.
- `range`: Specifies the scope of the request in units of time.

A `range` parameter begins with a minus sign, marking an interval backwards from the current time. This is followed by an integer number of units, followed by a letter specifying whether the interval is years ('y'), weeks ('w'), days ('d'), hours ('h'), minutes ('m'), or seconds ('s').

Note: For Process search, the `range` parameter is called `window`.

When setting a time range, either `start` and `end` must *both* be specified, or `range` must be specified. `range` takes precedence if it is specified alongside `start` and/or `end`.

Executing a Query

To execute a query after it's been refined, simply evaluate the query in an *iterable context*. This may be done either by passing it to a function that takes iterable values, or by iterating over it in a `for` loop. This example shows how a device query may be executed:

```
# create and refine a device query
>>> from cbc_sdk.platform import Device
>>> device_query = api.select(Device).where('status:ACTIVE').set_os(["WINDOWS"])

# easiest way to execute it is to turn it into a list
>>> matching_devices = list(device_query)

# or you can iterate over it using a for loop
```

(continues on next page)

(continued from previous page)

```
>>> for matching_device in device_query:
...     print(f"Matching device ID: {matching_device.id}")
...
Matching device ID: 1234
Matching device ID: 5678

# using it in a list comprehension also works
>>> matching_device_ids = [device.id for device in device_query]
>>> print(matching_device_ids)
[1234, 5678]

# you can also use the standard Python len() function to return the number of results
>>> print(len(device_query))
2
```

The `first()` or `one()` methods on a query always return the first object matched by that query. The difference between those is that, if there is more than one result for that query, the `one()` method will raise an error.

Asynchronous Queries

Some queries may also be executed asynchronously by using the `execute_async()` method, which is useful if you have a query which will take a long time to execute and you want your script to do other things while waiting for the query to return. Here's how we execute the device query from the last example asynchronously:

```
# create and refine a device query
>>> from cbc_sdk.platform import Device
>>> device_query = api.select(Device).where('status:ACTIVE').set_os(["WINDOWS"])

# now execute it
future = device_query.execute_async()

# await the results
device_list = future.result()
```

The `execute_async()` method returns a standard `concurrent.futures.Future` object, and that Future's `result()` method will return a list with the results of the query.

Faceting

Facet search queries return statistical information indicating the relative weighting of the requested values as per the specified criteria. Only certain query types support faceting.

Simple Faceting

Simple faceting is built into certain queries, allowing you to generate a summary on certain fields of all objects that match the query. To perform this, create and refine a query object as you would normally, then call the `facets()` method on the query, passing it the names of the fields you want to facet on.

Here is an example for USB devices:

```
>>> from cbc_sdk.endpoint_standard import USBDevice
>>> usb_devices = api.select(USBDevice).set_statuses(['APPROVED'])
>>> facet_data = usb_devices.facets(['vendor_name', 'product_name'])
```

This facet query might produce data that looks like this:

```
[
  {
    "field": "vendor_name",
    "values": [
      {
        "id": "Generic",
        "name": "Generic",
        "total": 2
      },
      {
        "id": "Kingston",
        "name": "Kingston",
        "total": 2
      }
    ]
  },
  {
    "field": "product_name",
    "values": [
      {
        "id": "DataTraveler 3.0",
        "name": "DataTraveler 3.0",
        "total": 2
      },
      {
        "id": "Mass Storage",
        "name": "Mass Storage",
        "total": 2
      }
    ]
  }
]
```

Facet Queries

More complex facet queries are performed by creating a query *on* a facet type, then refining it as usual, then getting the results from the query:

```
>>> from cbc_sdk.platform import ObservationFacet
>>> query = api.select(ObservationFacet).where(process_pid=1000)
```

Facet queries have two types of special criteria that may be set. One is the range type which is used to specify discrete values (integers or timestamps - specified both as seconds since epoch and also as ISO 8601 strings). The results are then grouped by occurrence within the specified range:

```
>>> from cbc_sdk.platform import ObservationFacet
>>> range = {
...     "bucket_size": "+1DAY",
...     "start": "2020-10-16T00:00:00Z",
...     "end": "2020-11-16T00:00:00Z",
...     "field": "device_timestamp"
... }
>>> query = api.select(ObservationFacet).where(process_pid=1000).add_range(range)
```

The range settings are as follows:

- **field** - the field to return the range for, should be a discrete one (integer or ISO 8601 timestamp)
- **start** - the value to begin grouping at
- **end** - the value to end grouping at
- **bucket_size** - how large of a bucket to group results in. If grouping an ISO 8601 property, use a string like '-3DAYS'.

Multiple ranges can be configured per query by passing a list of range dictionaries.

The other special criterion that may be set is the **term** type, which allows for one or more fields to use as a criteria on which to return weighted results. Terms may be added using the `add_facet_field()` method, specifying the name of the field to be summarized:

```
>>> from cbc_sdk.platform import ObservationFacet
>>> query = api.select(ObservationFacet).where(process_pid=1000).add_facet_field(
    ↪ "process_name")
```

Once the facet query has been fully refined, it is executed by examining its `results` property:

```
>>> from cbc_sdk.platform import EventFacet
>>> event_facet_query = api.select(EventFacet).add_facet_field("event_type")
>>> event_facet_query.where(process_guid="WNEXFKQ7-00050603-0000066c-00000000-
    ↪ 1d6c9acb43e29bb")
>>> range = {
...     "bucket_size": "+1DAY",
...     "start": "2020-10-16T00:00:00Z",
...     "end": "2020-11-16T00:00:00Z",
...     "field": "device_timestamp"
... }
>>> event_facet_query.add_range(range)
>>> synchronous_results = event_facet_query.results
>>> print(synchronous_results)
```

(continues on next page)

(continued from previous page)

EventFacet object, bound to <https://defense-eap01.conferdeploy.net>.

```
-----
num_found: 16
processed_segments: 1
  ranges: [{'start': '2020-10-16T00:00:00Z', 'end': '2020...
  terms: [{'values': [{'total': 14, 'id': 'modload', 'na...
total_segments: 1
```

Facet queries may also be executed asynchronously, as with other asynchronous queries, by calling their `execute_async()` method and then calling the `result()` method on the returned Future object:

```
>>> from cbc_sdk.platform import EventFacet
>>> event_facet_query = api.select(EventFacet).add_facet_field("event_type")
>>> event_facet_query.where(process_guid="WNEXFKQ7-00050603-0000066c-00000000-
↳ 1d6c9acb43e29bb")
>>> range = {
...     "bucket_size": "+1DAY",
...     "start": "2020-10-16T00:00:00Z",
...     "end": "2020-11-16T00:00:00Z",
...     "field": "device_timestamp"
... }
>>> event_facet_query.add_range(range)
>>> asynchronous_future = event_facet_query.execute_async()
>>> asynchronous_result = asynchronous_future.result()
>>> print(asynchronous_result)
```

EventFacet object, bound to <https://defense-eap01.conferdeploy.net>.

```
-----
num_found: 16
processed_segments: 1
  ranges: [{'start': '2020-10-16T00:00:00Z', 'end': '2020...
  terms: [{'values': [{'total': 14, 'id': 'modload', 'na...
total_segments: 1
```

The result for facet queries is a single object with two properties, `terms` and `ranges`, that contain the facet search result weighted as per the criteria provided:

```
>>> print(synchronous_result.terms)
[{'values': [{'total': 14, 'id': 'modload', 'name': 'modload'}, {'total': 2, 'id': 'crossproc', 'name': 'crossproc'}], 'field': 'event_type'}]
>>> print(synchronous_result.ranges)
[{'start': '2020-10-16T00:00:00Z', 'end': '2020-11-16T00:00:00Z', 'bucket_size': '+1DAY', 'field': 'device_timestamp', 'values': None}]
```

Query Timeouts

Some search queries make use of a timeout value, specified in milliseconds, which may be specified wither through a `timeout` parameter to a method, or via a `timeout()` setter method on a query class. These timeouts follow a specific set of rules.

The *absolute maximum* timeout value is 300,000 milliseconds (5 minutes). No search may have a timeout longer than this.

An application may specify a *shorter* maximum timeout value for all searches by including it in the credentials, under the key name `default_timeout`. This default timeout value may not be greater than the absolute maximum timeout. If this value is specified, no search may have a timeout longer than this value.

This means that, for any given search, the timeout will be the *smallest* of these values:

- The value specified via a parameter to the search, if one was specified.
- The value configured in the credentials, if one is so configured.
- The absolute maximum timeout value, as defined above.

Search Suggestions

Some classes offer the ability to provide “suggestions” as to search terms that may be employed, via a static method on the class. Here is an example for `Observation`:

```
>>> from cbc_sdk.platform import Observation
>>> suggestions = Observation.search_suggestions(api, query="device_id", count=2)
>>> for suggestion in suggestions:
...     print(suggestion["term"], suggestion["required_skus_all"], suggestion["required_
↪skus_some"])
device_id [] ['threathunter', 'defense']
netconn_remote_device_id ['xdr'] []
```

And here is an example for `BaseAlert`:

```
>>> from cbc_sdk.platform import BaseAlert
>>> suggestions = BaseAlert.search_suggestions(api, query="device_id")
>>> for suggestion in suggestions:
...     print(suggestion["term"], suggestion["required_skus_some"])
device_id ['defense', 'threathunter', 'deviceControl']
device_os ['defense', 'threathunter', 'deviceControl']
[...additional entries elided...]
workload_name ['kubernetesSecurityRuntimeProtection']
```

Alerts

Use alerts to get notifications about monitored activities such as the appearance or spread of risky files on your end-points. The Carbon Black Cloud Python SDK provides an easy way to search, investigate and set the workflow of Alerts using python classes instead of raw requests.

You can use all the operations shown in the API, such as retrieving, filtering, closing, and adding notes to the alert or the associated threat. You can locate the full list of operations and attributes in the `Alert()` class.

Resources

- [API Documentation](#) on Developer Network
- [Alert Search Fields](#) on Developer Network
- Example script in [GitHub](#)
- If you are updating from SDK version 1.4.3 or earlier, see the ``alerts-migration`` guide.
- If you are updating from Notifications, see the ``notification-migration`` guide.

Note: In Alerts v7, and therefore SDK 1.5.0 onwards, Observed Alerts are not included; they are an Observation. The field `category` has been removed from Alert. In other APIs where this field remains it will always have a value of `THREAT`. More information is available [here](#).

Retrieve Alerts

By using the following the example, you can retrieve the first 5 `[:5]` alerts that have a minimum severity level of 7.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import Alert
>>> api = CBCloudAPI(profile='sample')
>>> alerts = api.select(Alert).set_minimum_severity(7)[:5]
>>> print(alerts[0].id, alerts[0].device_os, alerts[0].device_name, alerts[0].category)
d689e626-5d6a-<truncated> WINDOWS Alert-WinTest THREAT
```

Filter Alerts

Filter alerts by using the fields described in the [Alert Search Schema](#).

Set required values for specific fields by using the `add_criteria()` method to limit the number of returned alerts. Use this method for fields that are identified in the [Alert Search Fields](#) with “Searchable Array”.

The following snippet limits returns to specific devices, where the `device_id` is an integer and the `device_target_value` is a string.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import Alert
>>> api = CBCloudAPI(profile='sample')
>>> alerts = api.select(Alert).add_criteria("device_id", [123, 456])
>>> alerts = api.select(Alert).add_criteria("device_target_value", ["MISSION_CRITICAL",
↪ "HIGH"])
```

Fields in the [Alert Search Fields](#) identified only with “Searchable” require the criteria to be a single value instead of a list of values. The SDK has hand-crafted methods to set the criteria for these fields.

The following code snippet shows the methods for `alert_notes_present` and `minimum_severity`, and the alerts that meet each criteria.

```
>>> alerts = api.select(Alert).set_alert_notes_present(True)
>>> print(len(alerts))
3
```

(continues on next page)

(continued from previous page)

```
>>> alerts = api.select(Alert).set_minimum_severity(9)
>>> print(len(alerts))
1072
>>> alerts = api.select(Alert).set_minimum_severity(3)
>>> print(len(alerts))
69100
>>>
```

You can use the `where` method to define a custom query to filter alerts. The `where` method supports strings and solr-like queries. Alternatively, you can use `solr` query objects for more complex searches. The following example searches by using a solr query search string for alerts where the `device_target_value` is `MISSION_CRITICAL` or `HIGH` and is the equivalent of the preceding `add_criteria` clause.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import Alert
>>> api = CBCloudAPI(profile='sample')
>>> alerts = api.select(Alert).where("device_target_value:MISSION_CRITICAL or device_
↳target_value:HIGH")
>>> for alert in alerts:
...     print(alert.id, alert.device_os, alert.device_name, alert.device_target_value)
8aa6272a-17cb-31c0-9352-67e45c0251f3 WINDOWS jenkins MISSION_CRITICAL
d987a112-8b7b-18c9-43d9-76ced09d9ded WINDOWS MYDEMOMACHINE\DESKTOP-04 MISSION_CRITICAL
0f915c4d-5652-b3e5-50d8-f4dcfc632396 WINDOWS jenkins MISSION_CRITICAL
1f13e581-840f-1207-f661-d9b176ee9d6c WINDOWS jenkins MISSION_CRITICAL
6ae56007-1213-4ee1-a50c-d221066ce8c9 WINDOWS MYBUILDMACHINE\Desktop-01 HIGH
... truncated ...
```

Tip: When filtering by fields that take a list parameter, an empty list is treated as a wildcard and matches everything.

For example, the following snippet returns all types:

```
>>> alerts = api.select(Alert).set_types([])
```

It is equivalent to:

```
>>> alerts = api.select(Alert)
```

Tip: More information about the `solr` can be found in their [documentation](#).

Retrieving Alerts for Multiple Organizations

By using the following example, you can retrieve alerts for multiple organizations. Ensure you have a profile created for each org in the cbc credential file.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import Alert
>>> org_list = ["org1", "org2"]
>>> for org in org_list:
```

(continues on next page)

(continued from previous page)

```
...     org = "".join(org)
...     api = CBCloudAPI(profile=org)
...     alerts = api.select(Alert).set_minimum_severity(7)[:5]
...     print("Results for Org {}".format(org))
>>> for alert in alerts:
...     print(alert.id, alert.device_os, alert.device_name, alert.category)
```

You can also read from a csv file by using values that match the profile names in a credentials.cbc file.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import Alert
>>> import csv
>>> file = open("data.csv", "r", encoding="utf-8-sig")
>>> org_list = list(csv.reader(file, delimiter=","))
>>> file.close()
>>> for org in org_list:
...     org = "".join(org)
...     api = CBCloudAPI(profile=org)
...     alerts = api.select(Alert).set_minimum_severity(7)[:5]
...     print("Results for Org {}".format(org))
>>> for alert in alerts:
...     print(alert.id, alert.device_os, alert.device_name, alert.category)
```

Grouping Alerts

The examples below illustrates how to create and manipulate grouped alert objects. A Grouped Alert is a collections of alerts that share a common threat id. When grouping alerts by a threat id it allows greater context and insight surrounding the pervasiveness of a threat.

This first example retrieves all groupings of watchlist alerts from the past 10 days that have a minimum severity level of 3. If this feels familiar to basic alert retrieval, the only difference of note at this stage is that we select a GroupedAlert instead of an Alert.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import GroupedAlert
>>> api = CBCloudAPI(profile="sample")
>>> grouped_alert_search_query = api.select(GroupedAlert)
>>> grouped_alert_search_query = grouped_alert_search_query.set_time_range(range="-10d").
↳ add_criteria("type", "WATCHLIST").set_minimum_severity(3)
>>> # trigger the search to execute:
>>> grouped_alert = grouped_alert_search_query.first()
>>> print("Number of groups: {}, Total alerts in all groups {}".format(grouped_alert_
↳ search_query._total_results, grouped_alert_search_query._group_by_total_count))
Number of groups: 19, Total alerts in all groups 2454
```

Also like Alerts, first() can be used on the query to retrieve the first grouping of alerts and study the metadata for a given threat id.

```
>>> first_alert_grouping = grouped_alert_search_query.first()
>>> print(first_alert_grouping.count, first_alert_grouping.highest_severity, first_alert_
↳ grouping.device_count, first_alert_grouping.workflow_states)
534 7 3 ("OPEN": 534)
```

(continues on next page)

(continued from previous page)

```
>>> most_recent_alert = first_alert_grouping.most_recent_alert_  
>>> print(most_recent_alert.threat_id)
```

It may be necessary to retrieve all of the alerts from a threat id grouping for further inspection, it is possible to directly retrieve the associated alert search query from a given grouped alert

```
>>> alert_search_query = first_alert_grouping.get_alert_search_query()  
>>> alerts = alert_search_query.all()
```

It is also possible to create grouped facets from the group alert search query

```
>>> grouped_alert_facets = grouped_alert_search_query.facets(["type", "THREAT_ID"], 0, ↵  
↵ True)
```

Suppose instead of grouped alerts, you had been working with alerts and wanted to crossover to grouped alerts. Instead of building a new group alert query from scratch you can transform an alert search query into a grouped alert search query or vice versa!

```
>>> from cbc_sdk import CBCloudAPI  
>>> from cbc_sdk.platform import Alert, GroupedAlert  
>>> api = CBCloudAPI(profile="sample")  
>>> alert_search_query = api.select(Alert)  
>>> alert_search_query = alert_search_query.set_time_range(range="-10d").add_criteria(  
↵ "type", "WATCHLIST").set_minimum_severity(3)  
>>> group_alert_search_query = alert_search_query.set_group_by("threat_id")  
>>> alert_search_query = group_alert_search_query.get_alert_search_query()
```

Note: When transforming from one query type to another the sort order parameter is not preserved. If it is necessary, it will have to be added to the queries criteria manually.

Retrieving Observations to Provide Context About an Alert

All alert types other than Watchlist Alerts have associated Observations that provide more information about the interesting events that contributed to the identification of an Alert.

The Alert v7 object (supported in SDK 1.5.0 onwards) has significantly more metadata when compared to the earlier Alerts v6 API (in the SDK version 1.4.3 and earlier). Therefore, the enrichment might not be required depending on your use case. New fields include process, child process, and parent process commandlines and IP addresses for network events. Find the complete list of fields in the [Alert Search Fields](#)

Observations are part of [Investigate Search Fields](#). Available fields are identified by the route “Observation”. Methods on the Observation Class, which can be found here: [Observation\(\)](#)

For the entire Observation details including fields marked with OBSERVATION*** in the [Investigate Search Fields](#) then use `get_details()` on the Observation object.

```
>>> from cbc_sdk import CBCloudAPI  
>>> from cbc_sdk.platform import CBAnalyticsAlert  
>>> api = CBCloudAPI(profile="sample")  
>>> alert = api.select(Alert).add_criteria("type", "CB_ANALYTICS").first()  
>>> observations = alert.get_observations()
```

(continues on next page)

(continued from previous page)

```
>>> observations
[<cbc_sdk.platform.observations.Observation: id_
↳ a5aa40856d5511ee8059132eb84e1d6d:470147c9-d79b-3f01-2083-b30bc0c0629f> @ https://
↳ defense.conferdeploy.net]
>>> print(observations[0])
Observation object, bound to https://defense.conferdeploy.net.
-----
                alert_id: [list:1 item]:
                        [0]: 470147c9-d79b-3f01-2083-b30bc0c0629f
        backend_timestamp: 2023-10-18T01:28:59.900Z
        blocked_effective_reputation: KNOWN_MALWARE
                blocked_hash: [list:1 item]:
                        [0]:
↳ 659e469f8dadcb6c32ab1641817ee57c327003dffa443c3...
                blocked_name: c:\windows\system32\fltlb.dll
        childproc_effective_reputation: KNOWN_MALWARE
        childproc_effective_reputation_source: HASH_REP
                childproc_hash: [list:1 item]:
                        [0]:
↳ 659e469f8dadcb6c32ab1641817ee57c327003dffa443c3...
... truncated ...
```

Retrieving Processes to Provide Context About an Alert

You can retrieve process details on any Alert with a `process_guid`. You can use list slicing to retrieve the first `n` results (in the example, this value is 10). The full list of attributes and methods are in the `Process()` class.

For the entire process details including fields marked with `PROCESS***` in the [Investigate Search Fields](#) then use `get_details()` on the Process object.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import WatchlistAlert, Process
>>> api = CBCloudAPI(profile='sample')
>>> alerts = api.select(WatchlistAlert)[:10]
>>> for alert in alerts:
...     process = alert.get_process()
...     print(process)
{'alert_id': ['0a3c45bf-fce6-4a63', '12030b8f-ce3f-48bd'], 'attack_tactic': 'TA0002' <truncated>
↳ ...}
{'alert_id': ['02f6aecd-73d7-456d', 'e47c13dd-75a9-44de'], 'attack_tactic': 'TA0002' <truncated>
↳ ...}
... truncated ...
```

Get Process Events

You can fetch every event that corresponds with a Process by calling `process.events()`.

Note: Because calling the events can be an intensive task, in following example fetches only the first 10 events. Be cautious when calling `all()`.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import WatchlistAlert, Process
>>> api = CBCloudAPI(profile='sample')
>>> alert = api.select(WatchlistAlert).first()
>>> process = alert.get_process()
>>> events = process.events()[:10]
>>> print(events[0].event_description) # Note that I've stripped the '<share>' and '<link>'
    ↪ tags, which are also available in the response.
The application c:\program files (x86)\google\chrome\application\chrome.exe
    ↪ attempted to modify the memory of "c:\program files (x86)\google\chrome\
    ↪ application\chrome.exe", by calling the function "NtWriteVirtualMemory". The
    ↪ operation was successful.'
...

```

Device Control Alerts

Device Control Alerts are explained in the [Device Control](#) guide.

Container Runtime Alerts

Container Runtime Alerts represent alerts for behavior that is noticed inside a Kubernetes container. These alerts are based on network traffic and are triggered by anomalies from the learned behavior of workloads or applications. For these events, the `type` is `CONTAINER_RUNTIME`. Additional fields such as `connection_type` and `egress_group_name` are also available.

To see all available fields, filter Alert Types Supported to `CONTAINER_RUNTIME` on the [Alert Search Fields](#).

Alert Workflow

The Alert Closure workflow enables Alert lifecycle management.

An alert goes through the states of Open, In Progress, and Closed. Any transition can occur, including from Closed back to Open or In Progress.

The workflow leverages the alert search structure to specify the alerts to close.

1. Use an Alert Search to specify which Alerts will have their status updated.
 - The request body is a search request and all alerts matching the request will be updated.
 - Two common uses are to update one alert, or to update all alerts with a specific threat id.
 - Any search request can be used as the criteria to select alerts to update the alert status.

```
>>> # This query will select only the alert with the specified id
>>> ALERT_ID = "id of the alert that you want to close"
>>> alert_query = api.select(Alert).add_criteria("id", [ALERT_ID])
>>> # This query will select all alerts with the specified threat id. It is not used,
↳ again in this example
>>> alert_query_for_threat = api.select(Alert).add_criteria("threat_id",
↳ "CFED0B211ED09F8EC1C83D4F3FBF1709")
```

2. Submit a job to update the status of Alerts.

- The status can be OPEN, IN PROGRESS or CLOSED (previously DISMISSED).
- You may include a Closure Reason.

```
>>> # by calling update on the alert_query, the a request to change the status
>>> # for all alerts matching that criteria will be submitted
>>> job = alert_query.update("CLOSED", "RESOLVED", "NONE", "Setting to closed for SDK,
↳ demo")
```

3. The immediate response confirms that the job was successfully submitted.

```
>>> print("job.id = {}".format(job.id))
job.id = 1234567
```

4. Use the Job() cbc_sdk.platform.jobs.Job class to determine when the update is complete.

Use the Job object to wait until the Job has completed. The python script will wait while the SDK polls to determine when the job is complete.

```
>>> completed_job = job.await_completion().result()
```

5. Refresh the Alert Search to get the updated alert data into the SDK.

```
>>> alert.refresh()
>>> print("Status = {}, Expecting CLOSED".format(alert.workflow["status"]))
```

6. You can dismiss future Alerts that have the same threat id.

Use the sequence of calls to update future alerts that have the same threat id. This sequence is usually used in conjunction with with the alert closure; that is, you can use the dismiss future alerts call to close future occurrences and call an alert closure to close current open alerts that have the threat id.

```
>>> alert_threat_query = api.select(Alert).add_criteria("threat_id",
↳ "CFED0B211ED09F8EC1C83D4F3FBF1709")
>>> alert.dismiss_threat("threat remediation done", "testing dismiss_threat in the SDK")
>>> # To undo the dismissal, call update
>>> alert.update_threat("threat remediation un-done", "testing update_threat in the SDK")
```

High Volume and Streaming Solution for Alerts

For near-real-time streaming of alerts, see [Data Forwarder](#).

Asset Groups

Asset Groups provide a way to organize and manage your fleet of Endpoints, VM Workloads, and VDIs. Create groups of assets and apply policies to the groups so the protections of all similar assets are synchronized. The ability to add one asset to multiple groups, and rank policies for precedence in application, gives added flexibility and fine tuning for complex organizations.

You can locate the full list of operations and attributes in the [AssetGroup\(\)](#) class.

Resources

- [API Documentation](#) on Developer Network
- Example script in [GitHub](#)

Retrieve Asset Groups

There are two options for getting a list of asset groups. The function `get_all_groups()` does exactly that; returns all Asset Groups in your organization.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import AssetGroup
>>> api = CBCloudAPI(profile='sample')
>>> all_asset_groups = AssetGroup.get_all_groups(api)
>>> print("There are {} asset groups. First group: {}".format(len(all_asset_groups), all_
↪asset_groups[0]))
There are 1 asset groups. This is the first: AssetGroup object, bound to https://defense.
↪conferdeploy.net.
Partially initialized. Use .refresh() to load all attributes
-----
create_time: 2024-01-24T04:38:26.930Z
description: Windows No Policy
discovered: False
id: 34fc5890-caf0-400a-98ba-a81763960f6e
member_count: 1030
member_type: DEVICE
name: Windows No Policy
org_key: 7desj9gn
query: os.equals: "WINDOWS"
status: OK
update_time: 2024-01-24T04:38:27.972Z
```

Asset groups can also be searched using `name`, `policy_id` or `group_id` in the criteria element.

The example shows creating an `AssetGroupQuery` class, then adding criteria to limit the results and specifying the field to sort by. The query is not executed until it is accessed, in this case by iterating over the results.

Summary information for each asset group is printed, and then the devices in that asset group are listed.

```
>>> search_asset_group_query = api.select(AssetGroup)
>>> search_asset_group_query.add_criteria("name", "Second demo group")
>>> search_asset_group_query.sort_by("name", "ASC")
>>> for ag in search_asset_group_query:
>>>     print("\nAsset group name = {}. It has {} members".format(ag.name, ag.member_
    ↪count))
>>>     print("Policy assigned to the Asset Group is Name: {}, Id: {}".format(ag.policy_
    ↪name, ag.policy_id))
>>>     for d in ag.list_members():
>>>         print("Device Name: {}, Id: {}".format(d.name, d.id))
Asset group name = Second demo group. It has 3 members
Policy assigned to the Asset Group is Name: DemoPolicy, Id: 123456
Device Name: DemoDevice, Id: 2468642
Device Name: SDKDemo, Id: 1357975
Device Name: AnotherDemoMachine, Id: 19283746
...truncated ...
```

Create an Asset Group

The only required field when creating an asset group is the Asset Group Name.

Creating a group without a policy assigned enables the use of a group for visibility of specific assets. After creation, it is possible in use any of combination of assigning assets directly, adding a query or assigning a policy.

```
>>> new_asset_group = AssetGroup.create_group(api, "My Example Asset Group", description=
    ↪"Demonstrating the SDK")
>>> print(new_asset_group)
AssetGroup object, bound to https://defense.conferdeploy.net.
-----
create_time: 2024-01-24T05:47:34.378Z
description: Demonstrating the SDK
discovered: False
            id: aae06712-96d4-43ea-ae67-07112d6f670e
member_count: 0
member_type: DEVICE
            name: My Example Asset Group
            org_key: ABCD1234
            status: OK
update_time: 2024-01-24T05:47:34.378Z
```

Now add a query which will dynamically include any asset with the Windows operating system and a policy:

```
>>> new_asset_group.query = "os.equals:WINDOWS"
>>> new_asset_group.policy_id = 12345
>>> new_asset_group.save()
```

Parts of Carbon Black Cloud have asynchronous processing and are eventually consistent. When writing automated scripts, use the status field to determine when the asset group membership has finished updating.

- OK indicates the membership evaluation is complete
- UPDATING indicates that group's dynamic memberships are being re-evaluated

```
>>> import time
>>> while new_asset_group.status != "OK":
>>>     print("waiting")
>>>     time.sleep(5)
>>>     new_asset_group.refresh()
```

Then print the new asset:

```
>>> print("new_asset_group {}".format(new_asset_group))
new_asset_group, bound to https://defense.conferdeploy.net.
Last refreshed at Tue Jan 23 22:47:47 2024
-----
create_time: 2024-01-24T05:47:35.150Z
description: Demonstrating the SDK
discovered: False
            id: ceb27e6c-7c23-4dd5-af7a-3b0c14363240
member_count: 204
member_type: DEVICE
            name: My Example Asset Group
            org_key: ABCD1234
            policy_id: 12345
            policy_name: DemoPolicy
            query: os.equals:WINDOWS
            status: OK
            update_time: 2024-01-24T05:47:35.585Z
AssetGroup object, bound to https://defense.conferdeploy.net.
```

All attributes can also be provided to the create method:

```
>>> second_asset_group = AssetGroup.create_group(api, "Second example group", "Second_
↳ group description",
...                                             query = "os.equals:MAC", policy_id =
↳ 12345)
```

The `add_member()` function is used to assign a device directly to the group. (Compared to dynamically added, when the device matches the query on the asset group.)

```
>>> from cbc_sdk.platform import Device
>>> random_device = api.select(Device).first()
>>> second_asset_group.add_members(random_device)
```

Delete an Asset Group

To delete an Asset Group, use the delete method:

```
>>> second_asset_group.delete()
```

Preview Policy Rank Changes

The effective policy on a specific device is determined by the rank of policies the device is assigned, with higher ranked policies taking precedence.

The [example script](#) includes finding two policies that are likely have impactful changes. This snippet uses hard-coded values so the focus is on the method being called and output.

The preview method is a static class method on Policy, since it is a policy change that is being previewed.

The result is a `DevicePolicyChangePreview()` class, which contains information about all the device that would have a change in effective policy.

```
>>> from cbc_sdk.platform import Policy
>>> api = CBCloudAPI(profile='sample')
>>> policy_id = 1234
>>> # to get a policy that exists in your org: policy_id = api.select(Policy).first().id
>>> new_policy_position = 1
>>> changes = Policy.preview_policy_rank_changes(api, [(policy_id, new_policy_position)])
>>> print(changes[0])
DevicePolicyChangePreview object, bound to https://defense.conferdeploy.net.
-----
Current policy: #98765 at rank 7
  New policy: #1234 at rank 1
  Asset count: 264
  Asset query: ((-_exists_:ag_agg_key_manual AND ag_agg_key_
↳dynamic:9b0a62b19086bdbfcff5c62e581304a28cd445aee86d87c6d95c57483ae5e05b AND policy_
↳id:100714 AND policy_override:false) AND (os.equals: "WINDOWS"))
```

This change says there's an asset group that is currently using policy id 98765 which is ranked 7. If the change was processed the asset group would use a new policy, id 1234 which is at rank 1. This would affect 264 Assets and the Asset query can be used to find those Assets.

The Asset Query is a class of type `DeviceSearchQuery` which can be executed:

```
>>> devices = changes[0].asset_query
>>> print("type of devices object is {}".format(type(devices)))
>>> print(len(devices))
type of devices object is <class 'cbc_sdk.platform.devices.DeviceSearchQuery'>
264
```

Preview Asset Group Changes

Previewing the changes that would happen if an asset group was changed is very similar to the Preview Policy Rank Changes above.

Once Asset Groups have been created and policies assigned, the preview asset group changes function can be used to identify the devices that would have their group membership or effective policy impacted by creating or deleting an Asset Group, or by changing the query on the asset group.

Here we're working with a random asset group and policy, using the `first()` function.

A new policy is assigned and the existing query is not changed.

```
>>> asset_group = api.select(AssetGroup).first()
>>> policy_id = api.select(Policy).first().id
>>> api = CBCloudAPI(profile='sample')
>>> changes = AssetGroup.preview_update_asset_groups(api, [asset_group], policy_id,
↳asset_group.query)
>>> print("There are {} changes that would result from the proposed change. The first_
↳change:".format(len(changes)))
>>> print(changes[0])
DevicePolicyChangePreview object, bound to https://defense.conferdeploy.net.
-----
Current policy: #148443 at rank 96
  New policy: #80947 at rank 1
  Asset count: 117
  Asset query: ((-_exists_:ag_agg_key_manual AND -_exists_:ag_agg_key_dynamic AND_
↳policy_id:148443 AND policy_override:false) AND (os.equals:MAC))
```

Audit Log Events

In the Carbon Black Cloud, *audit logs* are records of various organization-wide events, such as:

- Log in attempts by users
- Updates to connectors
- Creation of connectors
- LiveResponse events

The Audit Log API allows these records to be retrieved in JSON format, sorted by time in ascending order (oldest records come first). The API call returns only *new* audit log records that have been added since the last time the call was made using the same API Key ID. Once records have been returned, they are *cleared* and will not be included in future responses.

When reading audit log records using a *new* API key, the queue for reading audit logs will begin three days earlier. This may lead to duplicate data if audit log records were previously read with a different API key.

Note: Future versions of the Carbon Black Cloud and this SDK will support a more flexible API for finding and retrieving audit log records. This Guide will be rewritten to cover this when it is incorporated into the SDK.

API Permissions

To call this API function, use a custom API key created with a role containing the `READ` permission on `org.audits`.

Example of API Usage

```
import time
from cbc_sdk import CBCloudAPI
from cbc_sdk.platform import AuditLog

cb = CBCloudAPI(profile='yourprofile')
running = True

while running:
    events_list = AuditLog.get_auditlogs(cb)
    for event in events_list:
        print(f"Event {event['eventId']}:")
        for (k, v) in event.items():
            print(f"\t{k}: {v}")
    # omitted: decide whether running should be set to False
    if running:
        time.sleep(5)
```

Check out the example script `audit_log.py` in the `examples/platform` directory on [GitHub](#).

Developing New Credential Providers

The credentials management framework for the CBC SDK is designed to allow different handlers to be implemented, which may supply credentials to the `CBCloudAPI` in ways not implemented by existing credential handlers.

Writing the Credential Provider

Find all classes required to implement a new credential provider in the `cbc_sdk.credentials` package. See below for descriptions of the classes. It is recommended, but not required, that your new credential provider inherit from the `CredentialProvider` abstract class, and that you implement the methods from that abstract class as detailed.

The arguments to the standard `__init__()` method are not defined by the interface specification; those may be used to initialize your credential provider in any desired fashion.

Using the Credential Provider

Create an instance of your credential provider object and pass it as the keyword parameter `credential_provider` when creating your `CBCloudAPI` object.

Example:

```
>>> provider = MyCredentialProvider()
>>> cbc_api = CBCloudAPI(credential_provider=provider, profile='default')
```

Your credential provider's `get_credentials()` method will be called, passing in any profile specified in the `profile` keyword parameter used when creating `CBCloudAPI`.

Credential Provider Reference

These are the classes from the `cbc_sdk.credentials` package that are used in making a credential provider.

CredentialValue class

This class is of an enumerated type, and represents the various credential items loaded by the credential provider and fed to the rest of the SDK code. The possible values are:

- URL - The URL used to access the Carbon Black Cloud. This value *must* be specified.
- TOKEN - The access token to be used to authenticate to the server. It is the same structure as the `X-Auth-Token`: defined for direct API access in [the developer documentation](#). This value *must* be specified.
- ORG_KEY - The organization key specifying which organization to work with. This value *must* be specified.
- SSL_VERIFY - A Boolean value indicating whether or not to validate the SSL connection. The default is `True`.
- SSL_VERIFY_HOSTNAME - A Boolean value indicating whether or not to verify the host name of the server being connected to. The default is `True`.
- SSL_CERT_FILE - The name of an optional certificate file used to validate the certificates of the SSL connection. If not specified, the standard system certificate verification will be used.
- SSL_FORCE_TLS_1_2 - A Boolean value. If this is `True`, the connection will be forced to use TLS 1.2 rather than any later version. The default is `False`.
- PROXY - If specified, this is the name of a proxy host to be used in making the connection.
- IGNORE_SYSTEM_PROXY - A Boolean value. If this is `True`, any system proxy settings will be ignored in making the connection to the server. The default is `False`.
- INTEGRATION - The name of the integration to use these credentials. The string may optionally end with a slash character, followed by the integration's version number. Passed as part of the `User-Agent`: HTTP header on all requests made by the SDK.

Values of this type have one method:

requires_boolean_value

```
def requires_boolean_value(self):
```

Returns whether or not this particular credential item takes a Boolean value.

Returns: `True` if the credential item takes a Boolean value, `False` if the credential item takes a string value.

Credentials class

The class that holds credentials retrieved from the credential provider, and is used by the rest of the SDK. It is effectively immutable after creation.

__init__

```
def __init__(self, values=None):
```

Initializes a new `Credentials` object.

Parameters:

- **values** (type dict): A dictionary containing the values to initialize the `Credentials` object with. The keys of this dictionary may be either `CredentialValue` objects or their lowercase string equivalents, e.g. `CredentialValue.URL` or `"url"`. The values in the dict are strings for those credential items with string values. For credential items with Boolean values, the values may be either `bool` values, numeric values (with 0 being treated as `False` and non-zero values treated as `True`), or string values. In the case of string values, the value must be `"0"`, `"false"`, `"off"`, or `"no"` to be treated as a `False` value, or `"1"`, `"true"`, `"on"`, or `"yes"` to be treated as a `True` value (all values case-insensitive). If an unrecognized string is used for a Boolean value, `CredentialError` will be raised. Unrecognized keys in the dict are ignored. Any missing items will be replaced by the default for that item.

Raises:

- `CredentialError` - If there is an error parsing a Boolean value string.

get_value

```
def get_value(self, key):
```

Retrieves a specific credential value from this object.

Parameters:

- **key** (type `CredentialValue`): Indicates which item to retrieve.

Returns: The value of that credential item (`str` or `bool` type).

__getattr__

```
def __getattr__(self, name):
```

Retrieves a specific credential value from this object. This is a bit of “syntactic sugar” allowing other code to access credential values, for instance, as `cred_object.url` instead of `cred_object.get_value(CredentialValue.URL)`.

Parameters:

- **name** (type `str`): Indicates which item to retrieve.

Returns: The value of that credential item (`str` or `bool` type).

Raises:

- `AttributeError` - If the credential item `name` was unrecognized.

CredentialProvider class

All credential providers *should* extend this abstract class, but, in any event, *must* implement the protocol it defines.

get_credentials

```
def get_credentials(self, section=None):
```

Return a `Credentials` object containing the configured credentials.

Parameters:

- **section** (type `str`): Indicates the credential section to retrieve. May be interpreted by the credential provider in any manner it likes; may also be ignored.

Returns: A `Credentials` object containing the retrieved credentials.

Raises:

- `CredentialError` - If there is an error retrieving the credentials.

Device Control

Using the Carbon Black Cloud SDK, you can retrieve information about USB devices used in your organization, and manage the blocking of such devices from access by your endpoints.

Note: USBDevice is distinct from either the Platform API Device or the Endpoint Standard Device. Access to USB devices is through the Endpoint Standard package from `cbc_sdk.endpoint_standard import USBDevice`.

Retrieving the List of Known USB Devices

Using a query of the USBDevice object, you can see which USB devices have been used on any endpoint in your organization:

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.endpoint_standard import USBDevice
>>> query = api.select(USBDevice).where('1')
>>> for usb in query:
...     print(f"{usb.vendor_name} {usb.product_name} {usb.serial_number} {usb.status}")
...
SanDisk Ultra 4C531001331122115172 UNAPPROVED
SanDisk Cruzer Dial 4C530000110722114075 UNAPPROVED
PNY USB 2.0 FD 07189613DD84E242 UNAPPROVED
USB Flash Disk FBI1305031200020 APPROVED
```

Note that individual USB devices may be APPROVED or UNAPPROVED. USB devices which are UNAPPROVED cannot be read on any endpoint with a policy that blocks unknown USB devices.

A USB device query can also be exported to either CSV or JSON format, for use by other software systems:

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.endpoint_standard import USBDevice
>>> query = api.select(USBDevice).where('1')
>>> job = query.export('CSV')
>>> csv_report = job.get_output_as_string()
>>> # can also get the output as a file or as enumerated lines of text
```

Approving A Specific Device

We can create an approval for a USB device by using the device's `approve()` method. First, we'll get a list of all unapproved USB devices:

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.endpoint_standard import USBDevice
>>> query = api.select(USBDevice).where('1').set_statuses(['UNAPPROVED'])
>>> usb_list = list(query)
>>> for usb in usb_list:
...     print(f"{usb.vendor_name} {usb.product_name} {usb.serial_number}")
```

(continues on next page)

(continued from previous page)

```
...
SanDisk Ultra 4C531001331122115172
SanDisk Cruzer Dial 4C530000110722114075
PNY USB 2.0 FD 07189613DD84E242
```

Now we'll select one of these devices and approve it:

```
>>> usb = usb_list[1]
>>> print(usb.status)
UNAPPROVED
>>> approval = usb.approve('Test1', 'API Testing')
>>> print(approval.approval_name)
Test1
>>> print(approval.notes)
API Testing
>>> print(approval.serial_number)
4C530000110722114075
>>> print(approval.id)
1ffd0a16-28ad-3fba-981d-d1c29c2903da
>>> print(usb.status)
APPROVED
```

The `approve()` method creates a `USBDeviceApproval` representing that particular device's approval, and also reloads the `USBDevice` so its status reflects the fact that it's been approved.

Removing A Device's Approval

Device approvals may be removed via the API as well. Starting from the end of the previous example:

```
>>> approval.delete()
>>> usb.refresh()
True
>>> print(usb.status)
UNAPPROVED
```

The `delete()` method is what causes the approval to be removed. We then use `refresh()` on the actual `USBDevice` object to allow its status to be updated.

Retrieving the List of Approvals

USB device approvals can also be enumerated directly:

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.endpoint_standard import USBDeviceApproval
>>> query = api.select(USBDeviceApproval)
>>> for approval in query:
...     print(f"{approval.id} {approval.approval_name} {approval.serial_number}")
... 
```

They can also be exported in a similar manner to USB devices:

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.endpoint_standard import USBDeviceApproval
>>> query = api.select(USBDeviceApproval)
>>> job = query.export('CSV')
>>> csv_report = job.get_output_as_string()
>>> # can also get the output as a file or as enumerated lines of text
```

Device Control Alerts

When an endpoint attempts to access a blocked USB device (the endpoint has USB device blocking configured and the USB device is not approved), a `DeviceControlAlert` is generated. These alerts may be queried using the standard Platform API components.

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import DeviceControlAlert
>>> query = api.select(DeviceControlAlert).where('1')
>>> alerts_list = list(query)
>>> for alert in alerts_list:
...     print(f"{alert.vendor_name} {alert.product_name} {alert.serial_number}")
...
USB Flash Disk FBI1305031200020
USB Flash Disk FBI1305031200020
USB Flash Disk FBI1305031200020
USB Flash Disk FBI1305031200020
PNY USB 2.0 FD 07189613DD84E242
PNY USB 2.0 FD 07189613DD84E242
PNY USB 2.0 FD 07189613DD84E242
```

There are a number of fields supported by `DeviceControlAlert` over and above the standard alert fields; see the [developer documentation](#) for details.

Differential Analysis

Differential Analysis provides the ability to compare and understand the changes between two [Live Query](#) runs. The differential is calculated based on point-in-time snapshots. These features answer the question, “What changed on endpoints, and when?”.

Overview

This guide follows the steps for comparing two “point-in-time snapshots” of endpoints using a few different options and downloading the results using the Differential object. This example aims to understand what Firefox add-ons were added or removed between the two Live Query snapshot intervals.

1. Prerequisites

To perform a Differential Analysis, create the “point-in-time” snapshots of your endpoints with Live Query or use existing ones. You can find a step-by-step Live Query API guide [here](#) and a version for the CBC Python SDK [here](#). The example Live Query runs look for added or removed Firefox add-ons.

2. Query Comparison

Start a Query Comparison with the ID's you received from step 1. If the supplied `newer_run_id` is from a recurring Live Query run, the `older_run_id` is not required - the backend will automatically compare it to previous to the supplied one. The backend will throw a specific error if you provide a query id from a single Live Query run. You can read more about it [here](#).

Query Comparison

Basic Query

This example shows the basic result of the `Differential` object. The `.newer_run_id()` method is required - it accepts the run id that you want to mark as the starting point-in-time snapshot. By default, only the number of changes between the two runs are returned. To receive the actual differential data, use the `.count_only()` method, as featured in the Actual Changes example.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.audit_remediation import Differential
>>>
>>> cb = CBCloudAPI(profile='sample')
>>>
>>> query = cb.select(Differential).newer_run_id('jcdqsju4utpaayj5dh5r2llzffeolg0u').
↳ older_run_id('yhb3wcea9y1l4asiltky5tupkgauzas')
>>> run = query.submit()
>>> print(run)
Differential object, bound to https://defense-dev01.cbdtest.io.
-----

    diff_processed_time: 0.037
      diff_results: [list:1 item]:
                    [0]: {'device_id': 11412673, 'change_count': 19, 'ad...
newer_run_create_time: 2022-10-19T13:29:34.429Z
      newer_run_id: n6cv24lh3pnh4zbciotahl82tm4tsuo7
newer_run_not_responded_devices: [list:1 item]:
                                [0]: 17331059
      older_run_create_time: 2022-10-19T13:19:49.812Z
      older_run_id: olquodvqz8kekxug2o2jsxcdnltak9hu
older_run_not_responded_devices: [list:1 item]:
                                [0]: 17331059
```

You can also access a dictionary representation of the response with the `.to_json()` method.

```
>>> print(run.to_json())
{'diff_processed_time': 0.037,
 'diff_results': [{'added_count': 1,
                   'change_count': 1,
                   'changes': None,
                   'device_id': 12345,
                   'newer_run_row_count': 21,
                   'older_run_row_count': 20,
                   'removed_count': 0}],
 'newer_run_create_time': '2022-08-10T13:07:44.194Z',
 'newer_run_id': 'jcdqsju4utpaayj5dh5r2llzffeolg0u',
 'newer_run_not_responded_devices': [],
```

(continues on next page)

(continued from previous page)

```
'older_run_create_time': '2022-08-10T12:57:03.872Z',
'older_run_id': 'yhb93wcea9y1l4asiltky5tupkgauzas',
'older_run_not_responded_devices': []}
```

Actual Changes

Using the `.count_only()` method with a value of `False` will allow you to see the actual changes between the two snapshots. To use this method, append it to the rest of the Differential object query. The actual changes will be in the `changes` property, under `diff_results`.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.audit_remediation import Differential
>>>
>>> cb = CBCloudAPI(profile='sample')
>>>
>>> query = cb.select(Differential).newer_run_id('jcdqsju4utpaayj5dh5r2llzffeolg0u').
↳older_run_id('yhb93wcea9y1l4asiltky5tupkgauzas').count_only(False)
>>> actual_changes = query.submit()
>>> print(actual_changes.diff_results)
[{'device_id': 11412673, 'change_count': 19, 'added_count': 19, 'removed_count': 0, 'changes': [
↳{'action': 'ADDED', 'fields': [{'key': 'name', 'value': 'Visionary - Soft'}]}, {'action': 'ADDED',
↳'fields': [{'key': 'name', 'value': 'Activist - Balanced'}]}, {'action': 'ADDED', 'fields': [{'key':
↳'name', 'value': 'Visionary - Balanced'}]}, {'action': 'ADDED', 'fields': [{'key': 'name',
↳'value': 'Innovator - Soft'}]}, {'action': 'ADDED', 'fields': [{'key': 'name', 'value': 'Activist -
↳Bold'}]}, {'action': 'ADDED', 'fields': [{'key': 'name', 'value': 'Dreamer - Soft'}]}, {'action':
↳'ADDED', 'fields': [{'key': 'name', 'value': 'Dreamer - Balanced'}]}, {'action': 'ADDED',
↳'fields': [{'key': 'name', 'value': 'Expressionist - Bold'}]}, {'action': 'ADDED', 'fields': [{'
↳key': 'name', 'value': 'Innovator - Bold'}]}, {'action': 'ADDED', 'fields': [{'key': 'name',
↳'value': 'AdGuard AdBlocker'}]}, {'action': 'ADDED', 'fields': [{'key': 'name', 'value':
↳'Expressionist - Balanced'}]}, {'action': 'ADDED', 'fields': [{'key': 'name', 'value':
↳'Visionary - Bold'}]}, {'action': 'ADDED', 'fields': [{'key': 'name', 'value': 'Playmaker - Soft
↳'}]}, {'action': 'ADDED', 'fields': [{'key': 'name', 'value': 'Innovator - Balanced'}]}, {'action':
↳'ADDED', 'fields': [{'key': 'name', 'value': 'Expressionist - Soft'}]}, {'action': 'ADDED',
↳'fields': [{'key': 'name', 'value': 'Playmaker - Balanced'}]}, {'action': 'ADDED', 'fields': [{'
↳key': 'name', 'value': 'Playmaker - Bold'}]}, {'action': 'ADDED', 'fields': [{'key': 'name',
↳'value': 'Activist - Soft'}]}, {'action': 'ADDED', 'fields': [{'key': 'name', 'value': 'Dreamer -
↳Bold'}]}], 'older_run_row_count': 26, 'newer_run_row_count': 45}]
```

In the example response you can see that 19 items were added between the two snapshot intervals.

Filter Devices

Using the `.set_device_ids()` you can narrow down the query to a specific devices only. The method accepts an array of integers. To use this method, append it to the rest of the Differential object query or combine it with any of the other methods.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.audit_remediation import Differential
>>>
>>> cb = CBCloudAPI(profile='sample')
```

(continues on next page)

(continued from previous page)

```

>>>
>>> query = cb.select(Differential).newer_run_id('jcdqsju4utpaayj5dh5r2llzffeolg0u').
↳older_run_id('yhb3wcea9y1l4asiltky5tupkgauzas')
>>> actual_changes = query.count_only(False).set_device_ids([12345])
>>> run = actual_changes.submit()
>>> print(run.to_json())
{'diff_processed_time': 0.039,
 'diff_results': [{'added_count': 1,
                    'change_count': 1,
                    'changes': [{'action': 'ADDED',
                                   'fields': [{'key': 'name',
                                                'value': 'AdBlocker Ultimate'}]}],
                    'device_id': 12345,
                    'newer_run_row_count': 21,
                    'older_run_row_count': 20,
                    'removed_count': 0}],
 'newer_run_create_time': '2022-08-10T13:07:44.194Z',
 'newer_run_id': 'jcdqsju4utpaayj5dh5r2llzffeolg0u',
 'newer_run_not_responded_devices': [],
 'older_run_create_time': '2022-08-10T12:57:03.872Z',
 'older_run_id': 'yhb3wcea9y1l4asiltky5tupkgauzas',
 'older_run_not_responded_devices': []}]

```

Export Results

Using the `.async_export()` you can create an asynchronous job that exports the results from the run. To use this method, append it to the rest of the Differential object query or combine it with any of the other methods.

```

>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.audit_remediation import Differential
>>>
>>> cb = CBCloudAPI(profile='sample')
>>>
>>> query = cb.select(Differential).newer_run_id('jcdqsju4utpaayj5dh5r2llzffeolg0u').
↳older_run_id('yhb3wcea9y1l4asiltky5tupkgauzas')
>>> export = query.count_only(False).set_device_ids([12345]).async_export()
>>> export.await_completion()
>>> # write the results to a file
>>> export.get_output_as_file("example_data.json")

```

Live Query

With Live Query, you can ask questions of endpoints and quickly identify areas for improving security and IT hygiene.

You can use recommended queries created by Carbon Black security experts or craft your own SQL queries. Live Query is powered by <https://osquery.io>, an open source project that uses an SQLite interface. This guide will get you started using Live Query via the Python SDK.

More information about the Audit and Remediation product which uses Live Query is available in the [Carbon Black Cloud user guide](#)

More information about Live Query APIs is available on the [Developer Network](#).

Overview

This guide shows how to find specific files on a system. This is the same scenario as the Quick Start Guide for the APIs on the [Developer Network](#)

The steps we'll go through are:

1. Set up the python imports and Carbon Black Cloud credentials
2. Start the Query Run
3. Look at the results
4. Write the results to a file
5. Clean up since this is a tutorial
6. Get the run information for scheduled queries (templates)

Setting up

The code snippets assume that the python environment has been set up with the necessary imports and credentials.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk import audit_remediation
>>> from cbc_sdk.audit_remediation import Run, RunHistory, Result, ResultQuery
>>> api = CBCloudAPI(profile='sample')
```

For more information on credential handling in the SDK, see [Authentication](#)

Start a Query Run

Set up the query for the file you are looking for. Then create a query object, execute it, and get the id of the run.

```
>>> query_string = "SELECT filename, path FROM file WHERE path = 'C:\\\\Windows\\Temp\\\\"
↳ dbutil_2_3.sys\\' OR path LIKE 'C:\\\\Users\\%\\AppData\\Local\\Temp\\dbutil_2_3.sys';"
>>> query_object = api.select(Run).where(sql=query_string)
>>> run = query_object.submit()
>>> print(f'Run id: {run.id} has {run.active_org_devices} active devices in the org of,
↳ which {run.in_progress_count} are in progress and {run.not_started_count} have not,
↳ started')
Run id: vsc2be500dcuhc1q5bhvq7kdwoqh367i has 97 active devices in the org of which 0 are,
↳ in progress and 97 have not started
```

Check status

Give the run a few seconds to initialise, then refresh the information and print some statistics.

```
>>> run.refresh()
True
>>> print(f'Run id: {run.id} has {run.active_org_devices} active devices in the org of,
↳ which {run.in_progress_count} are in progress and {run.not_started_count} have not,
↳ started')
```

(continues on next page)

(continued from previous page)

```
Run id: vsc2be500dcuhc1q5bhvq7kdwoqh367i has 97 active devices in the org of which 45
↪are in progress and 33 have not started
```

The run status returns all the information about the progress of query execution. These are some of the interesting fields that show the number of devices available to be queried and progress.

- active_org_devices: 97
- error_count: 3
- in_progress_count: 45
- last_result_time: 2021-12-23T21:21:26.437Z
- match_count: 0
- no_match_count: 45
- not_started_count: 40
- status: ACTIVE
- total_results: 0

All details of the run can be pretty printed with

```
>>> print(run)
```

Get the results

Partial results can be reviewed while the query is running. This snippet gets the results and prints the device information for each.

```
>>> result_query = api.select(Result).run_id(run.id)
>>> list_result = list(result_query)
>>> for result in list_result:
>>>     print(f'Device: {result.device_id} has status {result.status}. Device message:
↪{result.device_message}')
Device: 1234578 has status matched. Device message:
Device: 3456789 has status error. Device message: Error: database or disk is full
Device: 8765432 has status matched. Device message:
```

There is also a helper option to get the results:

```
>>> results_by_helper = run.query_results()
```

Export results

It is possible to export the results in several formats including csv, zipped csv and streaming lines. These options are documented in [cbc_sdk.audit_remediation.base.ResultQuery\(\)](#)

This snippet shows writing the results to a zipped csv file.

```
>>> result_query.export_zipped_csv("/Users/myname/mydir/livequeryresults.zip")
```

For very large result sets there is an asynchronous API call. The SDK makes use of Python Futures to wait for the underlying call to complete.

For this call, in addition to live query permissions the API Key will require `jobs.status(READ)`.

The sequence of calls are:

```
>>> # first an extra import
>>> from cbc_sdk.platform import Job
>>> # then start the job
>>> job = result_query.async_export()
>>> # show the status in progress
>>> print(job.status)
IN_PROGRESS
>>> # wait for it to finish and refresh the information in the SDK
>>> job_future = job.await_completion()
>>> finished_job = job_future.result()
>>> finished_job.refresh()
>>> # show the job has completed
>>> print(finished_job.status)
COMPLETED
>>> # write the results to a csv file
>>> finished_job.get_output_as_file("/Users/myname/mydir/livequeryresults_async.csv")
```

Scroll results

If you would like to ingest all the Live Query results whether that be from one Run or multiple Runs consider using the scroll option to fetch the latest results. The scroll option is limited to the last 24 hours for results across all Runs. You either need to specify a `time_received` or a list of one or more Run ids

```
>>> result_query = api.select(Result).set_time_received(range="-3h")
>>> list_results = result_query.scroll(10)
>>> print(f"num_remaining: {result_query.num_remaining}")
num_remaining: 35
>>> while result_query.num_remaining > 0:
>>>     list_results.extend(result_query.scroll(10))
>>> print(f"total results: {len(list_results)}")
total_results: 45
```

Alternatively if you wanted to get all the results over multiple days for a single Run then use the Run's id

```
>>> result_query = api.select(Result).set_run_ids([run.id])
>>> list_results = result_query.scroll(10)
>>> print(f"num_remaining: {result_query.num_remaining}")
num_remaining: 62
>>> while result_query.num_remaining > 0:
>>>     list_results.extend(result_query.scroll(10))
>>> print(f"total results: {len(list_results)}")
total_results: 72
```

Clean up

Since this is a tutorial we'll clean up when we're done by first stopping the run and then deleting it.

Stopping the run will prevent the request going to any devices that have not yet checked in but will not stop the query running on any that are in progress. Checking in the console, the run and results will be visible with a status of Stopped.

```
>>> run.stop()
True
>>> print(run.status)
CANCELLED
```

Since this is a tutorial, we can fully clean up. This deletes the results so is probably not what you usually want. It will not be visible in the console and attempting to refresh the object will return the error "cannot refresh a deleted query".

```
>>> run.delete()
True
```

Scheduled runs (templates)

A template is a query that is scheduled to run periodically. It is likely easier to configured these using the Carbon Black Cloud console, but retrieving the result for import to another system may be useful.

An additional import:

```
>>> from cbc_sdk.audit_remediation import Template, TemplateHistory
```

List all the templates (scheduled queries):

```
>>> all_templates = api.select(TemplateHistory)
>>> for t in list(all_templates):
>>>     print(f'Name = {t.name}, id = {t.id}, next run time = {t.next_run_time}')
```

A where clause can be added to limit the templates returned. Each time the scheduled query has executed is a run.

```
>>> templates = list(api.select(TemplateHistory).where("CBC SDK Demo Template"))
>>> for template in templates:
>>>     print(f'template name = {template.name}, id = {template.id}, next run time = {t.
↳ next_run_time}')
```

and then get all the runs for each template

```
>>> runs = list(api.select(Template, template.id).query_runs())
>>> for run in runs:
>>>     print(f'Run id = {run.id}, Run Status = {run.status}, Run create time = {run.
↳ create_time}, Results Returned = {run.total_results}, Template Id = {run.template_id}')
```

name = CBC SDK Demo Template id = p7qtvxms0oaju46whcrfmyppa9fiqpn9
Run id = huobhstdtxpzhmg52yys7wmsuvjyx, Run Status = ACTIVE, Run create time = 2022-
↳ 01-19T21:00:00.000Z, Results Returned = 2333, Template Id =
↳ p7qtvxms0oaju46whcrfmyppa9fiqpn9
Run id = bdygnd8jvpjdqjmatdsuqzopaxebquqb, Run Status = TIMED_OUT, Run create time =
↳ 2022-01-18T21:00:00.000Z, Results Returned = 2988, Template Id =
↳ p7qtvxms0oaju46whcrfmyppa9fiqpn9

Live Response

You can use Live Response with the Carbon Black Cloud Python SDK to:

- Upload, download, or remove files
- Create, retrieve and remove registry entries
- Dump contents of physical memory
- Execute, terminate and list processes

Before any commands are sent to the live response session, the proper permissions need to be configured for the Custom Key that is used. The below table explains what permissions are needed for each of the SDK commands.

Command	Required Permissions	Explanation
Create LR session for device device.lr_session()	CREATE , org.liveresponse.session	READ CREATE is needed to start the LR session and READ is needed to check the status of the command
Close session lr_session.close()	READ , org.liveresponse.session	DELETE DELETE is needed to terminate the LR session and READ is needed to check the status of the command
Get Raw File lr_session.get_raw_file(...)	READ org.liveresponse.file	
Get File lr_session.get_file(...)	READ org.liveresponse.file	
Upload File lr_session.put_file(...)	CREATE , org.liveresponse.file	READ CREATE is needed to upload the file and READ is needed to check the status of the command
Delete file lr_session.delete_file(...)	READ , org.liveresponse.file	DELETE DELETE is needed to delete the file and READ is needed to check the status of the command
List Directory lr_session.list_directory(...)	READ org.liveresponse.file	
Create Directory lr_session.create_directory(...)	CREATE , org.liveresponse.file	READ CREATE is needed to create the directory and READ is needed to check the status of the command
Walk Directory lr_session.walk(...)	READ org.liveresponse.file	
Kill Process lr_session.kill_process(...)	READ , org.liveresponse.process	DELETE DELETE is needed to kill the process and READ is needed to check the status of the command
Create Process lr_session.create_process(...)	EXECUTE org.liveresponse.process OR EXECUTE org.liveresponse.process	If wait_for_completion = False, wait_for_output = False only EXECUTE is needed. Otherwise also file permissions are needed.
4.5. Guides	READ , DELETE org.liveresponse.file	59

To send commands to an endpoint, first establish a “session” with a device.

Note: As of version 1.3.0, Live Response has been changed to support CUSTOM type API Keys which enables the platform Device model and Live Response session to be used with a single API key. Ensure your API key has the Device READ permission along with the desired Live Response permissions.

Establish A Session With A Device

Connect to a device by querying the Device object.

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import Device
>>> device = api.select(Device).first()
>>> lr_session = device.lr_session()
```

File Commands

Once a session is established, create a directory and upload a file to that directory. The `list_directory` command returns the content of the directory, including the uploaded file.

```
>>> lr_session.create_directory('C:\\\\demo\\\\')
>>> lr_session.put_file(open("demo.txt", "r"), 'C:\\\\demo\\\\demo.txt')
>>> directories = lr_session.list_directory('C:\\\\demo\\\\')
>>> for directory in directories:
...     print(f"{directory['attributes'][0]} {directory['filename']}")
...
DIRECTORY .
DIRECTORY ..
ARCHIVE demo.txt
```

Note that the creation of the directory will fail if the directory already exists.

Next, get the contents of the file and then delete the file and the directory.

```
>>> contents = lr_session.get_file('C:\\\\demo\\\\demo.txt')
>>> lr_session.delete_file('C:\\\\demo\\\\demo.txt')
>>> lr_session.delete_directory('C:\\\\demo\\\\')
```

Note: you can also delete a directory with the delete file command.

Process Commands

You can also execute commands to manage processes. Once you have established a session, you can check running processes.

```
>>> processes = lr_session.list_processes()
>>> for process in processes:
...     print(f"{process['process_pid']} {process['process_path']}")
...
42 c:\windows\explorer.exe
43 c:\windows\system32\svchost.exe
```

You can also create or kill a process.

```
>>> lr_session.create_process(r'cmd.exe /c "ping.exe -t 127.0.0.1"',
                             wait_for_completion=False, wait_for_output=False)
>>> processes = lr_session.list_processes()
>>> for process in processes:
...     if 'ping.exe' in process['process_path']:
...         lr_session.kill_process(process['process_pid'])
```

Note: you must pass the PID of the process to kill it.

Additional Resources

Find a full list of supported commands in the [Live Response API documentation](#).

For tips on migrating from Live Response v3 to v6, check the [migration guide](#).

Policy

A policy determines preventative behavior and establishes sensor settings. Each endpoint sensor or sensor group is assigned a policy.

Policies are a collection of prevention rules and behavioral settings that define how your sensor interacts and prevents or allows behavior on your endpoint. Within Policies, you can create custom blocking rules, allow applications, and modify the way your sensor communicates with the Carbon Black Cloud.

Example scripts are available in the GitHub repository in `examples/platform` that demonstrate

- Basic Create, Read, Update, Delete and Export/Import operations for Prevention, Local Scan and Sensor rules
 - `policy_service_crud_operations.py`
- Core Prevention policy rule operations
 - `policy_core_prevention.py`
- Host-Based Firewall policy rule operations
 - `policy_host_based_firewall.py`
- Data Collection policy rule operations
 - Demonstrates how to enable and disable Auth Event collection.
 - `policy_data_collection.py`

Recommendations

Recommendations offer a quick shortcut for helping tune your policy configurations in an environment, by providing suggested reputation overrides which you may add to improve your policy. They can speed up the process of tuning your policy to an environment, rather than having to manually investigate endpoint activity and reconfigure the policy in response to those investigations.

The Carbon Black Cloud SDK for Python offers assistance for dealing with Recommendations.

Getting the List of Recommendations

By querying the Recommendation object, you can see which recommendations have already been generated for you by the Carbon Black Cloud.

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.endpoint_standard import Recommendation
>>> query = api.select(Recommendation).set_statuses(['NEW', 'ACCEPTED', 'REJECTED']).
↳ sort_by('impact_score', 'DESC')
>>> recslist = list(query)
>>> for rec in recslist:
...     print(rec)
...
```

Recommendation object, bound to https://example.org.

```

    impact: [RecommendationImpact object]:
        event_count: 2
        impact_score: 1.1710311
        impacted_devices: 44
        org_adoption: HIGH
        update_time: 2021-05-18T16:37:07.000Z

    new_rule: [RecommendationNewRule object]:
        filename: zoom.exe
        override_list: WHITE_LIST
        override_type: SHA256
        sha256_hash: 56f560d8254ebb453daef9abe5c3c6de2e18eafaa5a9e4...

    policy_id: 0
    recommendation_id: 5e6926d4-0c55-4757-a94d-e05883d5ee4c
    rule_type: reputation_override
    workflow: [RecommendationWorkflow object]:
        changed_by: estark@example.com
        comment: test_recommendation_review_dismissed
        create_time: 2021-05-18T16:37:07.000Z
        ref_id: 6d90188a0d4f11ecb02e15835b040340
        status: ACCEPTED
        update_time: 2021-09-04T07:12:13.000Z

```

Recommendation object, bound to https://example.org.

(continues on next page)

(continued from previous page)

```

    impact: [RecommendationImpact object]:
        event_count: 9
        impact_score: 0.2678737
        impacted_devices: 5
        org_adoption: HIGH
        update_time: 2021-05-18T16:37:07.000Z

    new_rule: [RecommendationNewRule object]:
        filename: cxuiuexe.exe
        override_list: WHITE_LIST
        override_type: SHA256
        sha256_hash: 90b196987fe62657bfce2627ab0a08a7096737363e13806...

    policy_id: 0
    recommendation_id: 100503cd-1897-425f-93b5-1ccba320438d
    rule_type: reputation_override
    workflow: [RecommendationWorkflow object]:
        changed_by: jbaratheon@example.com
        comment:
        create_time: 2021-05-18T16:37:07.000Z
        status: NEW
        update_time: 2021-09-14T07:12:13.000Z

```

Recommendation object, bound to <https://example.org>.

```

    impact: [RecommendationImpact object]:
        event_count: 12
        impact_score: 0.11177378
        impacted_devices: 315
        org_adoption: MEDIUM
        update_time: 2021-05-18T16:37:07.000Z

    new_rule: [RecommendationNewRule object]:
        filename: mbcloudea.exe
        override_list: WHITE_LIST
        override_type: SHA256
        sha256_hash: 0a2190c4ccfde82ef950836d014f31b2b188423bb67b51a...

    policy_id: 0
    recommendation_id: 3f89a837-034c-4b81-9f4c-f673a36ccb5c
    rule_type: reputation_override
    workflow: [RecommendationWorkflow object]:
        changed_by: tlannister@example.com
        comment: test_recommendation_review_dismissed
        create_time: 2021-05-18T16:37:07.000Z
        ref_id: 16e842eb152b11eca8407fb13248831f
        status: ACCEPTED
        update_time: 2021-09-14T07:12:15.000Z

```

Recommendation object, bound to <https://example.org>.

(continues on next page)

(continued from previous page)

```

        impact: [RecommendationImpact object]:
            event_count: 20
            impact_score: 0.05499694
            impacted_devices: 44
            org_adoption: MEDIUM
            update_time: 2021-05-18T16:37:07.000Z

        new_rule: [RecommendationNewRule object]:
            filename: svctcom.exe
            override_list: WHITE_LIST
            override_type: SHA256
            sha256_hash: d49a2beb44a603faf8aab2f5dfae3a292497c63f0b30d0e...

        policy_id: 0
        recommendation_id: 26ddb565-aff6-4b68-895c-fc286aa5f101
        rule_type: reputation_override
        workflow: [RecommendationWorkflow object]:
            changed_by: mtyrell@example.com
            comment: test_recommendation_review_dismissed
            create_time: 2021-05-18T16:37:07.000Z
            status: REJECTED
            update_time: 2021-09-11T07:12:14.000Z

```

N.B.: If you do not set status values on the recommendation query with `set_statuses()`, the search defaults to looking for NEW recommendations *only*.

Recommendations Workflow

Individual recommendations in the NEW state may be accepted or rejected by calling their `accept()` or `reject()` methods, respectively.

```

>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.endpoint_standard import Recommendation
>>> query = api.select(Recommendation).set_statuses(['NEW'])
>>> recommendation = query[0]
>>> recommendation.accept('Comment for acceptance')
>>> print(recommendation.workflow_.status)
ACCEPTED
>>> recommendation = query[1]
>>> recommendation.reject('Comment for rejection')
>>> print(recommendation.workflow_.status)
REJECTED

```

Individual recommendations in the ACCEPTED or REJECTED states may be reverted to the NEW state by calling their `reset()` method.

```

>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.endpoint_standard import Recommendation

```

(continues on next page)

(continued from previous page)

```
>>> query = api.select(Recommendation).set_statuses(['REJECTED'])
>>> recommendation = query.first()
>>> recommendation.reset()
>>> print(recommendation.workflow_.status)
NEW
```

Recommendations and Reputation Overrides

A recommendation in the ACCEPTED state will have a reputation override created for it. You can retrieve that object with the `reputation_override()` method.

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.endpoint_standard import Recommendation
>>> query = api.select(Recommendation).set_statuses(['ACCEPTED'])
>>> reputation_override = query.first().reputation_override()
>>> print(reputation_override)
ReputationOverride object, bound to https://example.org.
Last refreshed at Wed Oct 6 08:51:49 2021
-----

create_time: 2021-09-15T07:12:12.594Z
created_by: estark@example.com
description: test_recommendation_review
filename: panghip.exe
id: 3fa9f84515f411ecb2525dd14785e643
override_list: WHITE_LIST
override_type: SHA256
sha256_hash: 6a2cac7f36af5cebe0debbdb161d4f66b694b75192f1af4...
source: RECOMMENDATION
source_ref: 7b4e20d9-db28-408b-b7e9-af4008fa65cc
```

More information about reputation overrides may be found in [Reputation Override](#).

Reputation Override

Using the Carbon Black Cloud SDK, you can manage your ReputationOverrides to create a list of approved or banned applications using a SHA-256 hash, a certificate signer, or a path to a known IT tool application

Creating a Reputation Override

Using the ReputationOverride model, you can create new overrides directly provided you have the necessary required properties. For a SHA256 you need the hash and optionally the filename, IT_TOOL needs a file path with or without wildcards and optionally an indicator for including the child processes, CERT needs the signer of the application and optionally the certificate authority. See [the developer documentation](#) for more details.

```
>>> from cbc_sdk import CBCloudAPI
>>> cb = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import ReputationOverride
```

(continues on next page)

(continued from previous page)

```
>>> ReputationOverride.create(cb, {
...     "description": "An override for a sha256 hash",
...     "override_list": "BLACK_LIST",
...     "override_type": "SHA256",
...     "sha256_hash": "af62e6b3d475879c4234fe7bd8ba67ff6544ce6510131a069aaac75aa92aee7a",
...     "filename": "foo.exe"
... })
<cbc_sdk.platform.reputation.ReputationOverride: id 83008db065a611eb9a953907c2e1ed66> @_
↳ https://defense.conferdeploy.net
>>> ReputationOverride.create(cb, {
...     "description": "An override for an IT Tool",
...     "override_list": "WHITE_LIST",
...     "override_type": "IT_TOOL",
...     "path": "C://tools//*.exe",
...     "include_child_processes": True
... })
<cbc_sdk.platform.reputation.ReputationOverride: id 9e5c7a2f5ef140a989550c2351de1a32> @_
↳ https://defense.conferdeploy.net
>>> ReputationOverride.create(cb, {
...     "description": "An override for a CERT",
...     "override_list": "WHITE_LIST",
...     "override_type": "CERT",
...     "signed_by": "VMware Inc.",
...     "certificate_authority": "VMware"
... })
<cbc_sdk.platform.reputation.ReputationOverride: id 1768b71d356744498eec5ecd6526ca10> @_
↳ https://defense.conferdeploy.net
```

If you have an `EnrichedEvent` or `Process` object then you can use either `ban_process_sha256` or `approve_process_sha256` to add the applications sha256 hash to either the `WHITE_LIST` or `BLACK_LIST`.

```
>>> from cbc_sdk import CBCloudAPI
>>> cb = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import Process
>>> proc = cb.select(Process, "ABCD1234-00348f83-00000015c-000000000-1d667eb58a2ec94")
>>> proc.approve_process_sha256("Example approved sha256 from Process")
<cbc_sdk.platform.reputation.ReputationOverride: id 829e252b65aa11ebb1c7a965f279498c> @_
↳ https://defense.conferdeploy.net
```

Retrieving existing Reputation Overrides

Using a query of the `ReputationOverride` object, you can see the reputation overrides that have been created within your organization. If you want to filter the results try including `set_override_list` or `set_override_type` in your query or include a more restrictive where clause which can include wildcards such as `*tools*`.

```
>>> from cbc_sdk import CBCloudAPI
>>> cb = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import ReputationOverride
>>> overrides = cb.select(ReputationOverride).where("1")
>>> for override in overrides:
...     print(override)
```

(continues on next page)

(continued from previous page)

```
...
-----

    create_time: 2021-02-02T22:32:20.176Z
    created_by: ABCDE12345
    description: An override for an IT Tool
                 id: 83008db065a611eb9a953907c2e1ed66
include_child_processes: True
    override_list: WHITE_LIST
    override_type: IT_TOOL
                 path: C://tools//*.exe
```

If you already have an id for a ReputationOverride then you can make a query including the id as seen below.

```
>>> override = cb.select(ReputationOverride, 83008db065a611eb9a953907c2e1ed66)
>>> print(override)
-----

    create_time: 2021-02-02T22:32:20.176Z
    created_by: ABCDE12345
    description: An override for an IT Tool
                 id: 83008db065a611eb9a953907c2e1ed66
include_child_processes: True
    override_list: WHITE_LIST
    override_type: IT_TOOL
                 path: C://tools//*.exe
```

Deleting a Reputation Override

If you no longer need a ReputationOverride then you can delete the override using `delete()` or `bulk_delete([])` if you have a few that need deleted at once.

```
>>> from cbc_sdk import CBCloudAPI
>>> cb = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import ReputationOverride
>>> override = cb.select(ReputationOverride, 83008db065a611eb9a953907c2e1ed66)
>>> override.delete()
>>> ReputationOverride.bulk_delete([
...     "9e5c7a2f5ef140a989550c2351de1a32",
...     "1768b71d356744498eec5ecd6526ca10"
... ])
```

Unified Binary Store

The unified binary store (UBS) is a centralized service that is part of the Carbon Black Cloud. The UBS is responsible for storing all binaries and corresponding metadata for those binaries. The UBS is a feature of Enterprise EDR.

Get Download URL

```
>>> from cbc_sdk import CBCloudAPI
>>> cb = CBCloudAPI(profile='sample')
>>> from cbc_sdk.enterprise_edr.ubs import Binary
>>> sha256_hash = '8005557c1614c1e2c89f7db3702199de2b1e4605718fa32ff6ffdb2b41ed3759'
>>> binary = Binary(cb, sha256_hash)
>>> download_url = binary.download_url()
>>> print(download_url)
...
https://cdc-file-storage-staging-us-east-1.s3.amazonaws.com/80/05/55/7c/16/14/c1/<...
↪truncated...>
```

Note: The download link for the binary will be active for 1 hour (default expiration period).

Get Download URL Valid For Specific Period

We could set expiration period for the download link (in seconds).

```
>>> from cbc_sdk import CBCloudAPI
>>> cb = CBCloudAPI(profile='sample')
>>> from cbc_sdk.enterprise_edr.ubs import Binary
>>> sha256_hash = '8005557c1614c1e2c89f7db3702199de2b1e4605718fa32ff6ffdb2b41ed3759'
>>> binary = Binary(cb, sha256_hash)
>>> download_url = binary.download_url(expiration_seconds=30)
>>> print(download_url)
...
https://cdc-file-storage-staging-us-east-1.s3.amazonaws.com/80/05/55/7c/16/14/c1/<...
↪truncated...>
```

Note: The download link for the binary will be active for 30 seconds.

Searching Binaries

Currently searching binaries is not possible, but we could use the following syntax to obtain a single binary.

```
>>> from cbc_sdk import CBCloudAPI
>>> cb = CBCloudAPI(profile='sample')
>>> from cbc_sdk.enterprise_edr.ubs import Binary
>>> sha256_hash = '8005557c1614c1e2c89f7db3702199de2b1e4605718fa32ff6ffdb2b41ed3759'
>>> binary = cb.select(Binary, sha256_hash)
>>> print(download_url)
...
https://cdc-file-storage-staging-us-east-1.s3.amazonaws.com/80/05/55/7c/16/14/c1/<...
↪truncated...>
```


Note: If we try to use `binary = cb.select(Binary)`, it will fail with exception that the model is a non queryable model.

Find the full documentation at [Unified Binary Store](#).

Users and Grants

Using the Carbon Black Cloud SDK, you can work with the users in your organization, as well as their access grants and profiles.

Audience for This Guide

This guide is geared towards SDK users seeking to automate specialized management tasks in the Carbon Black Cloud. Typically, they will have administrative privilege.

Uniform Resource Names (URNs)

The various API functions that work with users and grants often make use of *uniform resource names* (URNs) that uniquely represent various pieces of the Carbon Black Cloud environment. These pieces include:

- **Organizations**, represented as `psc:org:ORGKEY`, where `ORGKEY` is the organization's alphanumeric key value.
- The special URN `psc:org:ORGKEY:CHILDREN`, where `ORGKEY` is the organization's alphanumeric key value, refers to all the *child organizations* of that organization, but *not* the organization itself.
- **Users**, represented as `psc:user:ORGKEY:USERID`, where `ORGKEY` is the organization's alphanumeric key value and `USERID` is the user's numeric login ID.
- **Access roles**, represented as `psc:role:OPT-ORGKEY:NAME`, where `OPT-ORGKEY` is (optionally) the alphanumeric key value of the organization containing that role, and `NAME` is the name of the role. A role that does not have an `OPT-ORGKEY` is a default/global role created for all organizations.

Most of these are dealt with for you by the Carbon Black Cloud SDK.

Getting a List of Users

We can do a query on the `User` object to get a list of users within the organization we're accessing.

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import User
>>> query = api.select(User)
>>> user_list = list(query)
>>> for user in user_list:
...     print(f"{user.first_name} {user.last_name} ({user.login_id}) <{user.email}>")
...
Lysa Arryn (#2345670) <larryn@example.com>
Olenna Redwyne (#2345671) <oredwyne@example.com>
Arianne Martell (#2345672) <amartell@example.com>
Jorah Mormont (#2345673) <jmormont@example.com>
```

We can restrict the query by user IDs or E-mail addresses by using the `user_ids([str])` or `email_addresses([str])` methods on the query object returned by `select()` before enumerating its results.

Modifying a User

A User can be modified by changing one or more of its fields and then calling its `save()` method.

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import User
>>> user = api.select(User, 2345672)
>>> print(user.phone)
800-555-0000
>>> user.phone = '888-555-9753'
>>> user.save()
<cbc_sdk.platform.users.User: id 2345672> @ https://defense.conferdeploy.net (*)
>>> print(user.phone)
888-555-9753
```

Note: A user's *role* can only be modified by updating the user's *access grant*, detailed below.

Creating a New User

Creating a user may be done with the help of a *builder object*, which is returned from the `User.create()` function.

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import User
>>> builder = User.create(api)
>>> builder.set_first_name('Samwell').set_last_name('Tarly')
<cbc_sdk.platform.users.User.UserBuilder object at 0x00000209B8123D00>
>>> builder.set_email('starly@example.com').set_phone('800-555-8008')
<cbc_sdk.platform.users.User.UserBuilder object at 0x00000209B8123D00>
>>> builder.set_role('psc:role::BETA_SYSTEM_ADMIN')
<cbc_sdk.platform.users.User.UserBuilder object at 0x00000209B8123D00>
>>> builder.build()
```

Alternately, you may construct a *template object* (a Python dict) that contains the user's information and create the user directly.

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import User
>>> user_template = {'first_name': 'Selyse', 'last_name': 'Florent', 'email':
↳ 'sflorent@example.com',
...                  'phone': '877-555-9099', 'role_urn': 'psc:role::BETA_SYSTEM_ADMIN'}
>>> User.create(api, user_template)
```

Note: A user that has just been created will *not* be visible in either the UI or in a User query as detailed above, until the user activates their account through the invitation E-mail message and sets a password.

User Access Grants

Every user object has an *access grant* object associated with it, defining the access roles they are permitted to use. You can use the `grant()` method on a `User` to get the grant and inspect or modify it.

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import User
>>> user = api.select(User, 2345672)
>>> print(f'{user.first_name} {user.last_name}')
Arianne Martell
>>> grant = user.grant()
>>> print(grant.roles)
['psc:role::BETA_SYSTEM_ADMIN']
>>> grant.roles = ['psc:role::BETA_VIEW_ONLY']
>>> grant.save()
<cbc_sdk.platform.grants.Grant: id psc:user:1A2B3C4DE:2345672> @ https://defense.
conferdeploy.net
>>> print(grant.roles)
['psc:role::psc:role::BETA_VIEW_ONLY']
```

You can see what roles your API key is able to access and assign using the `get_permitted_role_urns()` function:

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import Grant
>>> for index, role_urn in enumerate(Grant.get_permitted_role_urns(api)):
...     print(f'{index}. {role_urn}')
...
0. psc:role::BETA_LEVEL_3_ANALYST
1. psc:role::KUBERNETES_SECURITY_DATAPLANE_ONLY
2. psc:role::ALL_AND_LR
3. psc:role::BETA_LEVEL_1_ANALYST
4. psc:role::BETA_SYSTEM_ADMIN
5. psc:role::KUBERNETES_SECURITY_DATAPLANE
6. psc:role::VIEW_ONLY
7. psc:role::ALL
8. psc:role::KUBERNETES_SECURITY_ADMIN_USER
9. psc:role::BETA_SUPER_ADMIN
10. psc:role::KUBERNETES_SECURITY_READ_ONLY_USER
11. psc:role::CONTAINER_IMAGE_CLI_TOOL
12. psc:role::KUBERNETES_SECURITY_DEVOPS
13. psc:role::BETA_VIEW_ALL
14. psc:role::KUBERNETES_SECURITY_DEVOPS_VIEW_ONLY
15. psc:role::BETA_LEVEL_2_ANALYST
16. psc:role::KUBERNETES_SECURITY_DEVELOPER
```

Users created in the Carbon Black Cloud console employ *access profiles* on the access grants, which allow roles for a user to be specified for the organization and/or any child organizations. Access profiles may be accessed and manipulated through the access grant object.

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import User
```

(continues on next page)

(continued from previous page)

```
>>> user = api.select(User, 3456789)
>>> grant = user.grant()
>>> for profile in grant.profiles_:
...     print(f"{profile.allowed_orgs} - {profile.roles}")
...
['psc:org:1A2B3C4DE'] - ['psc:role::BETA_LEVEL_3_ANALYST']
['psc:org:2F3G4H5JK'] - ['psc:role::BETA_LEVEL_1_ANALYST']
```

Adding an access profile may be done via the `create_profile()` method on `Grant`:

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import User
>>> user = api.select(User, 3450987)
>>> grant = user.grant()
>>> builder = grant.create_profile()
>>> builder.add_org('psc:org:2F3G4H5JK').add_role('psc:role::BETA_VIEW_ALL')
<cbc_sdk.platform.grants.Grant.ProfileBuilder object at 0x00000232942C8400>
>>> profile = builder.build()
{'orgs': {'allow': ['psc:org:2F3G4H5JK']}, 'roles': ['psc:role::BETA_VIEW_ALL']}
```

Or it may be added via a template object (as with `User`):

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import User
>>> user = api.select(User, 3450987)
>>> grant = user.grant()
>>> profile_template = {'orgs': {'allow': ['psc:org:2F3G4H5JK']}, 'roles': [
↳ 'psc:role::BETA_VIEW_ALL']}
>>> profile = grant.create_profile(profile_template)
{'orgs': {'allow': ['psc:org:2F3G4H5JK']}, 'roles': ['psc:role::BETA_VIEW_ALL']}
```

Vulnerabilities

The Vulnerability Assessment API allows users to view asset (Endpoint or Workload) vulnerabilities, increase security visibility, and undertake prioritized proactive security patching on critical systems. The API provides a summary of vulnerability information filtered at the organization level, by device, or by vulnerability CVE ID. With a list of vulnerabilities prioritized by severity, exploitability, and current activity, users can apply proactive and impactful vulnerability patches. The Carbon Black Cloud Python SDK provides all of the functionalities you might need to use vulnerabilities efficiently. You can use all of the operations shown in the API such as retrieving, filtering, exporting, and performing actions. The full list of operations and attributes can be found in the [Vulnerability\(\)](#) class.

For more information see [the developer documentation](#)

Retrieving Vulnerabilities

With the example below, you can retrieve the 5 most recent non-critical vulnerabilities for an organization.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import Vulnerability
>>> api = CBCloudAPI(profile='sample')
>>> vulnerabilities = api.select(Vulnerability).set_severity("CRITICAL", "NOT_EQUALS
↳")[:5]
>>> print(vulnerabilities[0])

affected_assets: [list:1 item]:
    [0]: DESKTOP-KLVRRM4
    category: APP
    cve_id: CVE-1999-0794
    device_count: 1
    os_info: [dict] {
        os_arch: 64-bit
        os_name: Microsoft Windows 10 Pro
        os_type: WINDOWS
        os_version: 10.0.18363
    }
    os_product_id: 37_282511
    product_info: [dict] {
        arch:
        product: Microsoft Office
        release: None
        vendor: Microsoft Corporation
        version: 15.0.4693.1005
    }
    vuln_info: [dict] {
        active_internet_breach: False
        created_at: 1999-10-01T04:00:00Z
        cve_description: Microsoft Excel does not warn a user_
↳when a mac...
        cve_id: CVE-1999-0794
        cvss_access_complexity: Low
        cvss_access_vector: Local access
        cvss_authentication: None required
        cvss_availability_impact: Partial
        cvss_confidentiality_impact: Partial
        cvss_exploit_subscore: 3.9
        cvss_impact_subscore: 6.4
        cvss_integrity_impact: Partial
        cvss_score: 4.6
        cvss_v3_exploit_subscore: None
        cvss_v3_impact_subscore: None
        cvss_v3_score: None
        cvss_v3_vector: None
        cvss_vector: AV:L/AC:L/Au:N/C:P/I:P/A:P
        easily_exploitable: False
        fixed_by: None
        malware_exploitable: False
```

(continues on next page)

(continued from previous page)

```

    nvd_link: https://nvd.nist.gov/vuln/detail/CVE-
    ↪ 1999-0794
    risk_meter_score: 1.6
    severity: LOW
    solution: None
}

```

With the example below, you can retrieve the most recent vulnerability for a specific device type and operating system type.

```

>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import Vulnerability
>>> api = CBCloudAPI(profile='sample')
>>> vulnerability = api.select(Vulnerability).set_device_type("ENDPOINT", "EQUALS").set_
    ↪ os_type("WINDOWS", "EQUALS").first()
>>> print(vulnerability)

affected_assets: [list:1 item]:
    [0]: DESKTOP-KLVRRM4
    category: APP
    cve_id: CVE-1999-0794
    device_count: 1
    os_info: [dict] {
        os_arch: 64-bit
        os_name: Microsoft Windows 10 Pro
        os_type: WINDOWS
        os_version: 10.0.18363
    }
    os_product_id: 37_282511
    product_info: [dict] {
        arch:
        product: Microsoft Office
        release: None
        vendor: Microsoft Corporation
        version: 15.0.4693.1005
    }
    vuln_info: [dict] {
        active_internet_breach: False
        created_at: 1999-10-01T04:00:00Z
        cve_description: Microsoft Excel does not warn a user_
    ↪ when a mac...
        cve_id: CVE-1999-0794
        cvss_access_complexity: Low
        cvss_access_vector: Local access
        cvss_authentication: None required
        cvss_availability_impact: Partial
        cvss_confidentiality_impact: Partial
        cvss_exploit_subscore: 3.9
        cvss_impact_subscore: 6.4
        cvss_integrity_impact: Partial
        cvss_score: 4.6
        cvss_v3_exploit_subscore: None

```

(continues on next page)

(continued from previous page)

```

        cvss_v3_impact_subscore: None
        cvss_v3_score: None
        cvss_v3_vector: None
        cvss_vector: AV:L/AC:L/Au:N/C:P/I:P/A:P
        easily_exploitable: False
        fixed_by: None
        malware_exploitable: False
        nvd_link: https://nvd.nist.gov/vuln/detail/CVE-
↪ 1999-0794
        risk_meter_score: 1.6
        severity: LOW
        solution: None
    }

```

With the example below you can retrieve the 5 most recent vulnerabilities for a device type sorted by status.

```

>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import Vulnerability
>>> api = CBCloudAPI(profile='sample')
>>> vulnerabilities = api.select(Vulnerability).set_device_type("WORKLOAD", "EQUALS").
↪ sort_by("status")[:5]
>>> for vulnerability in vulnerabilities:
...     print(vulnerability.cve_id, vulnerability.category, vulnerability.device_count,
↪ vulnerability.os_product_id)
...
CVE-2008-5915 APP 1 4_820212
CVE-2008-5915 APP 1 4_1027024
CVE-2008-5915 APP 1 4_1107922
CVE-2008-5915 APP 1 4_1336654
CVE-2008-5915 APP 1 7_64452

```

Filtering

You can use the `where` method to filter the vulnerabilities. The `where` supports strings and solr like queries, alternatively you can use the `solrq` query objects for more complex searches. The example below will search with a solr query search string for the last 5 vulnerabilities in the OS category.

```

>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import Vulnerability
>>> api = CBCloudAPI(profile='sample')
>>> vulnerabilities = api.select(Vulnerability).where("OS")[:5]
>>> for vulnerability in vulnerabilities:
...     print(vulnerability.cve_id, vulnerability.category, vulnerability.device_count,
↪ vulnerability.os_product_id)
...
CVE-2010-3974 OS 2 14_0
CVE-2010-3974 OS 1 61_0
CVE-2011-0032 OS 2 14_0
CVE-2011-0032 OS 1 61_0

```

(continues on next page)

(continued from previous page)

CVE-2011-0034 OS 2 14_0

Tip: More information about the solrq can be found in the their [documentation](#).

Retrieving Vulnerability Details

With the example below, you can retrieve vulnerability details for the most recent vulnerability.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import Vulnerability
>>> api = CBCloudAPI(profile='sample')
>>> vulnerability = api.select(Vulnerability).first()
>>> print(vulnerability.vuln_info)

{
  'cve_id': 'CVE-1999-0794',
  'cve_description': 'Microsoft Excel does not warn a user when a macro is present in a_
↳ Symbolic Link (SYLK) format file.',
  'risk_meter_score': 1.6,
  'severity': 'LOW',
  'fixed_by': None,
  'solution': None,
  'created_at': '1999-10-01T04:00:00Z',
  'nvd_link': 'https://nvd.nist.gov/vuln/detail/CVE-1999-0794',
  'cvss_access_complexity': 'Low',
  'cvss_access_vector': 'Local access',
  'cvss_authentication': 'None required',
  'cvss_availability_impact': 'Partial',
  'cvss_confidentiality_impact': 'Partial',
  'cvss_integrity_impact': 'Partial',
  'easily_exploitable': False,
  'malware_exploitable': False,
  'active_internet_breach': False,
  'cvss_exploit_subscore': 3.9,
  'cvss_impact_subscore': 6.4,
  'cvss_vector': 'AV:L/AC:L/Au:N/C:P/I:P/A:P',
  'cvss_v3_exploit_subscore': None,
  'cvss_v3_impact_subscore': None,
  'cvss_v3_vector': None,
  'cvss_score': 4.6,
  'cvss_v3_score': None
}
```


Retrieving Affected Assets for a Vulnerability

With the example below, you can retrieve a list of affected assets for the last 5 critical vulnerabilities.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import Vulnerability
>>> api = CBCloudAPI(profile='sample')
>>> vulnerabilities = api.select(Vulnerability).set_severity("CRITICAL", "EQUALS")[:5]
>>> for vulnerability in vulnerabilities:
...     print(vulnerability.affected_assets)
...
['DESKTOP-KLVRRM4']
['DESKTOP-KLVRRM4']
['DESKTOP-KLVRRM4']
['Windowhost-MAD', 'WINDOWHOST2-MAD']
['Windowhost-MAD', 'WINDOWHOST2-MAD']
```

Watchlists, Feeds, Reports, and IOCs

Watchlists are a powerful feature of Carbon Black Cloud Enterprise EDR. They allow an organization to set-and-forget searches on their endpoints' incoming events data, providing the administrator the opportunity to sift through high volumes of activity and focus attention on those that matter.

Note: Use of these APIs requires that the organization be enabled for Enterprise EDR. Verify this by logging into the Carbon Black Cloud Console, opening the menu in the upper right corner, and checking for an **ENABLED** flag against the “Enterprise EDR” entry.

All examples here assume that a Carbon Black Cloud SDK connection has been set up, such as with the following code:

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
```

Setting up a connection is documented here: [Getting Started with the Carbon Black Cloud Python SDK - “Hello CBC”](#)

About the Objects

An *indicator of compromise* (IOC) is a query, list of strings, or list of regular expressions which constitutes actionable threat intelligence that the Carbon Black Cloud is set up to watch for. Any activity that matches one of these may indicate a compromise of an endpoint.

A *report* groups one or more IOCs together, which may reflect a number of possible conditions to look for, or a number of conditions related to a particular target program or type of malware. Reports can be used to organize IOCs.

A *watchlist* contains reports (either directly or through a feed) that the Carbon Black Cloud is matching against events coming from the endpoints. A positive match will trigger a “hit,” which may be logged or result in an alert.

A *feed* contains reports which have been gathered by a single source. They resemble “potential watchlists.” A watchlist may be easily subscribed to a feed, so that any reports in the feed act as if they were in the watchlist itself, triggering logs or alerts as appropriate.

Setting Up a Basic Custom Watchlist

Creating a custom watchlist that can watch incoming events and/or generate alerts requires three steps:

1. Create a report including one or more Indicators of Compromise (IOCs).
2. Add that report to a watchlist.
3. Enable alerting on the watchlist.

Creating a Report

In this example, a report is created, adding one or more IOCs to it:

```
>>> from cbc_sdk.enterprise_edr import Report, IOC_V2
>>> builder = Report.create(api, "Unsigned Browsers", "Unsigned processes impersonating_
↳ browsers", 5)
>>> builder.add_tag("compliance").add_tag("unsigned_browsers")
>>> builder.add_ioc(IOC_V2.create_query(api, "unsigned-chrome",
...                                     "process_name:chrome.exe NOT process_publisher_state:FILE_SIGNATURE_
↳ STATE_SIGNED"))
>>> report = builder.build()
>>> report.save_watchlist()
```

Reports should always be given a title that's sufficiently unique within your organization, so as to minimize the chances of confusing two or more Reports with each other. Carbon Black Cloud will generate unique id values for each report, but does not enforce any uniqueness constraint on the title of reports.

Alternatively, you can update an existing report, adding more IOCs and/or replacing existing ones. To find an existing report associated with a watchlist, you must look in the watchlist's reports collection:

```
>>> from cbc_sdk.enterprise_edr import Watchlist, Report, IOC_V2
>>> watchlist = api.select(Watchlist, 'R4cMgFIhRaakgk749MRr6Q')
>>> report_list = [report for report in watchlist.reports if report.id == '47474d40-1f94-
↳ 4995-b6d9-1d1eea3528b3']
>>> report = report_list[0]
>>> report.append_iocs([IOC_V2.create_query(api, 'evil-connect', 'netconn_ipv4:10.8.16.4
↳ ')])
>>> report.update()
```

Adding the Report to a Watchlist

Now, add the new Report to a new Watchlist:

```
>>> from cbc_sdk.enterprise_edr import Watchlist
>>> builder = Watchlist.create(api, "Suspicious Applications")
>>> builder.set_description("Any signs of suspicious applications running on endpoints").
↳ add_reports([report])
>>> watchlist = builder.build()
>>> watchlist.save()
```

If you already have an existing Watchlist you wish to enhance, you can add Reports to the existing Watchlist:

```
>>> # "report" contains the Report that was created in the previous example
>>> from cbc_sdk.enterprise_edr import Watchlist
>>> watchlist = api.select('Watchlist', 'R4cMgFIhRaakgk749MRr6Q')
>>> watchlist.add_reports([report])
>>> watchlist.save()
```

Enabling Alerting on a Watchlist

When either the `alerts_enabled` or `tags_enabled` attributes of a watchlist are `True`, that Watchlist will create data you can act on - either alerts or hits, respectively; if both are `False`, the Watchlist is effectively disabled.

Once you have the Watchlist configured with the IOCs that are generating the kinds of hits (results) you are after, you can enable Alerting for the Watchlist, which will allow matches against the reports in the watchlist to generate alerts. If a watchlist identifies suspicious behavior and known threats in your environment, you will want to enable alerts to advise you of situations where you may need to take action or modify policies.

```
>>> watchlist.enable_alerts()
```

A Closer Look at IOCs

In this document, only the “v2” IOCs are covered; the “v1” IOCs are only provided for backwards compatibility reasons. They are officially deprecated, and are converted, internally, to this type.

IOCs can be classified into two general types, depending on their `match_type` value:

Query IOCs are those with a `match_type` of `query`; their `values_list` contains a single string that specifies a query compatible with process searches. For example, the following IOC looks for the process `git.exe` that does *not* connect to one of a specified list of IP addresses:

```
{
  "id": "example_1",
  "match_type": "query",
  "values": ["process_name:git.exe NOT (netconn_ipv4:35.158.151.206 OR netconn_ipv4:1.
  ↪1.244.78
              OR netconn_ipv4:80.18.61.229 OR netconn_ipv4:80.18.61.228)"]
}
```

Query IOCs must always use field-prefixed queries (key-value pairs); they do not support just searching for a value without a field specified. Values in query clauses that do not specify fields will be ignored.

Wrong

```
process_name:chrome.exe AND 192.168.1.1
```

Right

```
process_name:chrome.exe AND netconn_ipv4:192.168.1.1
```

Query IOCs may search on CIDR address ranges, e.g.: `netconn_ipv4:192.168.0.0/16`.

Query IOCs are searched every 5 minutes by the Carbon Black Cloud, and are tested against a rolling window of the last hour’s worth of data for the organization. (They will *not* generate hits or alerts for process attributes that were reported more than an hour in the past.) They may employ any searchable field as documented [here](#), and may employ complex query logic.

Ingress IOCs are those with a `match_type` of `equality` or `regex`; they use the `field` element to specify the name of a field to examine the value of, and the `values_list` element to specify a list of values to match against (in the case

of `match_type` being equality) or regular expressions to match against (in the case of `match_type` being `regex`). For example, this IOC will match any process that initiates a connection to one of two listed IP addresses:

```
{
  "id": "example_2",
  "match_type": "equality",
  "field": "netconn_ipv4",
  "values": ["8.8.8.8", "1.160.120.15"]
}
```

This IOC will match any process running with an executable name beginning with “quake”:

```
{
  "id": "example_3",
  "match_type": "regex",
  "field": "process_name",
  "values": ["quake.*\\.exe"]
}
```

(Note the use of the backslash to escape the ‘.’ that separates the file extension from the name. It must be doubled to escape it in Python itself.)

Ingress IOCs are searched as soon as the data is received from any endpoint, and may use any process field (as documented [here](#); the fields that may be used in this context are tagged with `PROCESS`) in their `field` element, whether searchable or not. For the searches they are capable of, they are more efficient than query IOCs, and also easier to add additional search target values to. They can, however, only search on a single field at a time.

Note: Ingress IOCs cannot be edited in the Carbon Black Cloud console UI at this time, due to a UI limitation on editing two properties of an IOC at the same time.

You *can* include more than one entry (query or match element) in an individual IOC, but in order to ignore or disable one of those entries, you would either have to edit the IOC or disable it entirely (thus disabling *all* entries in that IOC). It is recommended to use only one entry per IOC, for ease of management, unless you have already vetted the entries and don’t expect to have to disable them individually.

Both IOCs and reports may include a `link` property, which is used by the Carbon Black Cloud console UI as a hint to indicate that this IOC or report is being managed outside of the console. If this property is not `None`, the console UI will disable the ability to edit the IOC or report, but they can still be edited via the API.

Creating an IOC

You can create an IOC via the `IOC_V2` class, there are 3 available methods that you can use to initiate your IOC: `IOC_V2.create_query`, `IOC_V2.create_equality`, `IOC_V2.create_regex`.

Creating an equality IOC

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.enterprise_edr import IOC_V2
>>> cbcsdk = CBCloudAPI(profile="default")
>>> IOC_V2.create_equality(cbcsdk, None, "netconn_domain", ["localhost.local"])
<cbc_sdk.enterprise_edr.threat_intelligence.IOC_V2: id ad361179-d586-4c99-af3e-
821224cc0fd9> @ https://<CBCInstanceURL>
```

Creating a query IOC

```
>>> IOC_V2.create_query(cbcsdk, None, "{process_hash:098f6bcd4621d373cade4e832627b4f6}")
<cbc_sdk.enterprise_edr.threat_intelligence.IOC_V2: id 36d68cab-4739-4aa6-afcc-
↳2921d2e5573e> @ https://<CBCInstanceURL>
```

Creating a regex IOC

```
>>> IOC_V2.create_regex(cbcsdk, None, "process_name", r"(^/usr/.*)|(^/bin/.*)")
<cbc_sdk.enterprise_edr.threat_intelligence.IOC_V2: id 5170a04c-bbfc-4449-b939-
↳d5fc9f55d555> @ https://<CBCInstanceURL>
```

Removing and adding an IOC from a Report

If you want to remove an IOC from a report, you will need the IOC id and the report id.

```
>>> from cbc_sdk.enterprise_edr import Report
>>> ioc_id = "<ioc_id>"
>>> report = cbc_sdk.select(Report).where(id="<report_id>", feed_id="<feed_id>")[0]
<cbc_sdk.enterprise_edr.threat_intelligence.Report: id 1e69c54e-7cc9-41b8-9d1d-
↳3fd59a003d8a> @ https://<CBCInstanceURL>
>>> report.remove_iocs_by_id([ioc_id])
>>> report.update()
<cbc_sdk.enterprise_edr.threat_intelligence.Report: id 1e69c54e-7cc9-41b8-9d1d-
↳3fd59a003d8b> @ https://<CBCInstanceURL> (*)
```

Adding the IOC into the report works the same way:

```
>>> from cbc_sdk.enterprise_edr import Report, IOC_V2
>>> ioc_id = "<ioc_id>"
>>> report = cbc_sdk.select(Report).where(id="<report_id>", feed_id="<feed_id>")[0]
<cbc_sdk.enterprise_edr.threat_intelligence.Report: id 1e69c54e-7cc9-41b8-9d1d-
↳3fd59a003d8a> @ https://<CBCInstanceURL>
>>> ioc = IOC_V2.create_regex(cbcsdk, None, "process_name", r"(^/usr/.*)|(^/bin/.*)")
>>> report.append_iocs([ioc])
>>> report.update()
<cbc_sdk.enterprise_edr.threat_intelligence.Report: id 1e69c54e-7cc9-41b8-9d1d-
↳3fd59a003d8b> @ https://<CBCInstanceURL> (*)
```

Note: Calling the *Report.save()* method after the insertion or removal of IOC does not update the report and it's likely to result in a bad call to the API.

If the report is in a watchlist instead of a feed then you have to get the appropriate watchlist and iterate over the reports.

```
>>> from cbc_sdk.enterprise_edr import Watchlist, Report, IOC_V2
>>> ioc_id = "<ioc_id>"
>>> report_id = "<report_id>"
>>> watchlist = cbc_sdk.select(Watchlist, "<watchlist_id>")
```

(continues on next page)

(continued from previous page)

```

<cbc_sdk.enterprise_edr.threat_intelligence.Watchlist: id <watchlist_id>> @ https://
↳<CBCInstanceURL>
>> ioc = IOC_V2.create_regex(cbc_sdk, None, "process_name", r"(^/usr/.*)|(^/bin/.*)")
>> reports = watchlist.reports
>> report = [report_ for report_ in reports if report_.id == report_id][0]
>> report.append_iocs([ioc])
>> report.update()
<cbc_sdk.enterprise_edr.threat_intelligence.Report: id 1e69c54e-7cc9-41b8-9d1d-
↳3fd59a003d8b> @ https://<CBCInstanceURL> (*)

```

Tips for Using IOCs

- You can safely ignore certain fields in an IOC. For example, fields like `alert_id` and `process_guid` will always uniquely identify just a single record in your organization's data, whereas a field like `org_id` will be a constant across *all* your organization's data.
- Timestamp fields such as `backend_timestamp` are useful in ad-hoc queries, to look for data occurring before or after a certain date, but are of limited usefulness over the span of time a watchlist may be running.
- A list of hashes (such as with `process_sha256`) can be of limited value. They are inconvenient to keep current, especially as software (whether legitimate or malicious) gets updated over time, but are definitely easier to manage with equality IOCs.
- Counter fields (such as `netconn_count`) can be useful with range queries to locate processes that are using a large number of resources. For example, the query `netconn_count:[500 TO *]` will match only processes that make a large number of network connections.
- When using ingress IOCs, be careful of errant characters in the values list, such as leading or trailing whitespace or embedded newline characters. These errant characters may cause the IOCs to fail to match, leading to false negative results.
- equality IOCs for IPv4 fields (e.g. `netconn_remote_ipv4`) cannot support CIDR notation; full IP addresses must be used.
- equality IOCs for IPv6 fields (e.g. `netconn_remote_ipv6`) do not support standard or CIDR notation at this time. All IPv6 addresses must omit colon characters, spell out all zeroes in the address, and represent all alphabetic characters in uppercase. For example, "ff02::fb" becomes "FF0200000000000000000000000000FB".

Feeds

Another way of managing reports is to attach them to a *feed*. Feeds can contain multiple reports, and a feed can be attached to a watchlist, effectively making the contents of the watchlist equivalent to the contents of the feed.

Feeds are in effect "potentially-subscribable Watchlists". A Feed has no effect on your organization until it is subscribed to, by creating a Watchlist containing that feed. Once subscribed (and until it's disabled or unsubscribed), a watchlist will generate hits (and alerts if you have enabled them) for any matches against any of the IOCs in any of that feed's enabled reports.

Note: The feeds that are created by these examples are *private feeds*, meaning they are only visible within an organization and can be created by anyone with sufficient privileges in the organization. There are additional types of feeds; *reserved feeds* can only be created by MSSPs, and *public feeds* can only be created or edited by VMware Carbon Black.

A new feed may be created as follows (assuming the new report for that feed is stored in the `report` variable):

```
>>> from cbc_sdk.enterprise_edr import Feed
>>> builder = Feed.create(api, 'Suspicious Applications', 'http://example.com/location',
...                       'Any signs of suspicious applications running on our endpoints
↳', 'external_threat_intel')
>>> builder.set_source_label('Where the info is coming from')
>>> builder.add_reports([report])
>>> feed = builder.build()
>>> feed.save()
```

If you have an existing feed, a new report may be added to it as follows (assuming the new report is stored in the `report` variable):

```
>>> from cbc_sdk.enterprise_edr import Feed
>>> feed = cb.select(Feed, 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
>>> feed.append_reports([report])
```

To update or delete an existing report in a feed, look for it in the feed's `reports` collection, then call the `update()` method on the report to replace its contents, or the `delete()` method on the report to delete it entirely. The `replace_reports()` method on the `Feed` object may also be used, but caution must be taken, as that method will replace *all* of the reports in a feed at once.

To subscribe to a feed, a new watchlist must be created around it:

```
>>> watchlist = Watchlist.create_from_feed(feed, "Subscribed feed", "Subscription to the_
↳new feed")
>>> watchlist.save()
```

Limitations of Reports and Watchlists

Individual reports may contain no more than 10,000 IOCs. Reports containing more than 1,000 IOCs will not be editable via the Carbon Black Cloud console UI, but may still be managed using APIs.

Individual watchlists may contain no more than 10,000 reports. Any more than that may lead to timeouts when managing the watchlist through the Carbon Black Cloud console UI, and possibly when managing it through APIs as well.

Workloads

These APIs allow you to visualize the inventory of compute resources available under either vSphere or AWS.

Note: A *compute resource* is a virtual machine without a sensor installed.

The API operations center around the `VCenterComputeResource` object for vSphere compute resources, or around the `AWSComputeResource` for AWS compute resources.

Note: The object name `ComputeResource` is an alias for `VCenterComputeResource`, provided for backwards compatibility with earlier versions of the SDK.

Search Compute Resources

By querying on one of the compute resource object types, you can obtain a list of matching compute resources. The SDK supports filtering by a number of different criteria, which are different for each compute resource type.

For VCenterComputeResource:

- appliance_uuid
- cluster_name
- datacenter_name
- esx_host_name
- esx_host_uuid
- vcenter_name
- vcenter_host_url
- vcenter_uuid
- name
- host_name
- ip_address
- device_guid
- registration_id
- eligibility
- eligibility_code
- installation_status
- installation_type
- uuid
- os_description
- os_type
- os_architecture
- vmwaretools_version

For AWSComputeResource:

- auto_scaling_group_name
- availability_zone
- cloud_provider_account_id
- cloud_provider_resource_id
- cloud_provider_tags
- id
- installation_status
- name
- platform

- platform_details
- region
- subnet_id
- virtual_private_cloud_id

Any of these criteria may be specified to be included in search results by calling the method `set_XXX`, or excluded by calling the method `exclude_XXX`, where `XXX` is the specific criteria name.

Example (vSphere workloads):

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import VCenterComputeResource

>>> cbc = CBCloudAPI()
>>> query = cbc.select(VCenterComputeResource).set_os_type(['WINDOWS']).set_cluster_
->name(['example-cluster-name'])
>>> for result in list(query):
...     print(result)
```

Example Output:

VCenterComputeResource object, bound to https://defense-dev01.cbdtest.io.

```
-----

  appliance_uuid: c74bca54-e903-49e8-9962-2bb895f428c1
    cluster_name: example-cluster-name
      created_at: 2021-02-25T04:54:41.362Z
  datacenter_name: cwp-bucket-1-datacenter
    eligibility: ELIGIBLE
  eligibility_code: None
    esx_host_name: 10.105.17.113
    esx_host_uuid: a2311b42-3e53-8f21-97d7-66680007185f
      host_name: appd2012
        id: 19902164
  installation_status: NOT_INSTALLED
  installation_status_code:
    ip_address: 10.105.17.84
      name: cwp-bucket-1-windows_2012
  os_architecture: 64
    os_description: Microsoft Windows Server 2012 (64-bit)
      os_type: WINDOWS
        uuid: 500e14e6-3ea6-23aa-11bd-8e68444c6ce4
  vcenter_host_url: 10.105.17.114
    vcenter_name: VMware vCenter Server 6.7.0 build-14368073
    vcenter_uuid: 9a8a0be5-ae1e-49ce-b2aa-34bc7dc445e3
  vmwaretools_version: 11328
VCenterComputeResource object, bound to https://defense-dev01.cbdtest.io.
-----

  appliance_uuid: c74bca54-e903-49e8-9962-2bb895f428c1
    cluster_name: example-cluster-name
      created_at: 2021-02-25T04:54:41.362Z
  datacenter_name: cwp-bucket-1-datacenter
```

(continues on next page)

(continued from previous page)

```

        eligibility: ELIGIBLE
    eligibility_code: None
    esx_host_name: 10.105.17.113
    esx_host_uuid: a2311b42-3e53-8f21-97d7-66680007185f
        host_name: appd2k8r2
            id: 19902168
    installation_status: NOT_INSTALLED
    installation_status_code:
        ip_address: 10.105.17.237
            name: cwp-bucket-1-windows_2008
    os_architecture: 64
    os_description: Microsoft Windows Server 2008 R2 (64-bit)
    os_type: WINDOWS
        uuid: 500e51ff-ca0d-5a70-a799-2595c9e87000
    vcenter_host_url: 10.105.17.114
        vcenter_name: VMware vCenter Server 6.7.0 build-14368073
        vcenter_uuid: 9a8a0be5-ae1e-49ce-b2aa-34bc7dc445e3
    vmwaretools_version: 11328 ComputeResource object, bound to https://defense-dev01.
    ↪ cbdtest.io.

```

Example (AWS workloads):

```

>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import AWSComputeResource

>>> cbc = CBCloudAPI()
>>> query = cbc.select(AWSComputeResource).set_region(['us-west-1'])
>>> results = list(query)
>>> for result in results:
...     print(result)

```

Example Output:

```

AWSComputeResource object, bound to https://defense-dev01.cbdtest.io.
-----

    auto_scaling_group_name: Demo-AutoScalingGroup
    availability_zone: us-west-1c
    cloud_provider_account_id: 267678331262
    cloud_provider_resource_id: i-043de738ce129b77a
    cloud_provider_tags: [list:4 items]:
        [0]: Name##Demo-ASG
        [1]: aws:ec2launchtemplate:id##lt-0e3d35dba4f5ba16f
        [2]: aws:autoscaling:groupName##Demo-AutoScalingGroup
        [...]
    create_time: 2022-06-02T05:23:27Z
    deployment_type: AWS
    eligibility: NOT_ELIGIBLE
    eligibility_code: [list:1 item]:
        [0]: SSM_DOC_NOT_INSTALLED
    external_ip: 18.144.80.202
        id: 8x5tjvywq-aws-i-043de738ce129b77a

```

(continues on next page)

(continued from previous page)

```

image_description: Amazon Linux 2 Kernel 5.10 AMI 2.0.20220426.0 x...
image_id: ami-02541b8af977f6cdd
image_name: amzn2-ami-kernel-5.10-hvm-2.0.20220426.0-x86_64...
installation_status: NOT_INSTALLED
installation_status_code: None
installation_status_code_key: None
instance_state: running
instance_type: t2.micro
internal_ip: 172.31.11.73
name: Demo-ASG
org_key: 8X5TJVVWQ
platform: Unix/Linux
platform_details: Linux/UNIX
platform_name: None
platform_version: None
region: us-west-1
security_group_id: [list:1 item]:
[0]: sg-085972ee2f0be60aa
subnet_id: subnet-03cb2d09e07350698
virtual_private_cloud_id: vpc-0faa4803c3de51c87
AWSComputeResource object, bound to https://defense-dev01.cbdtest.io.

```

```

auto_scaling_group_name: None
availability_zone: us-west-1c
cloud_provider_account_id: 267678331262
cloud_provider_resource_id: i-0febda35fcaf2dbd1
cloud_provider_tags: [list:1 item]:
[0]: Name##Rushit-Test-2
create_time: 2022-07-11T08:26:58Z
deployment_type: AWS
eligibility: NOT_ELIGIBLE
eligibility_code: [list:1 item]:
[0]: SSM_DOC_NOT_INSTALLED
external_ip: 54.193.100.2
id: 8x5tjvywq-aws-i-0febda35fcaf2dbd1
image_description: Amazon Linux 2 Kernel 5.10 AMI 2.0.20220606.1 x...
image_id: ami-0d9858aa3c6322f73
image_name: amzn2-ami-kernel-5.10-hvm-2.0.20220606.1-x86_64...
installation_status: NOT_INSTALLED
installation_status_code: None
installation_status_code_key: None
instance_state: running
instance_type: t2.micro
internal_ip: 172.31.7.55
name: Rushit-Test-2
org_key: 8X5TJVVWQ
platform: Unix/Linux
platform_details: Linux/UNIX
platform_name: None
platform_version: None
region: us-west-1

```

(continues on next page)

(continued from previous page)

```

security_group_id: [list:1 item]:
[0]: sg-08473e77b9e4921e3
subnet_id: subnet-03cb2d09e07350698
virtual_private_cloud_id: vpc-0faa4803c3de51c87
AWSComputeResource object, bound to https://defense-dev01.cbdtest.io.
-----

auto_scaling_group_name: Demo-AutoScalingGroup
availability_zone: us-west-1a
cloud_provider_account_id: 267678331262
cloud_provider_resource_id: i-0b8b62d7c3aea1f9f
cloud_provider_tags: [list:5 items]:
[0]: Name##Demo-ASG
[1]: Test##Rushit-ASG
[2]: aws:ec2launchtemplate:id##lt-0e3d35dba4f5ba16f
[...]
create_time: 2022-06-02T05:21:26Z
deployment_type: AWS
eligibility: NOT_ELIGIBLE
eligibility_code: [list:1 item]:
[0]: SSM_DOC_NOT_INSTALLED
external_ip: 54.176.174.194
id: 8x5tjvywq-aws-i-0b8b62d7c3aea1f9f
image_description: Amazon Linux 2 Kernel 5.10 AMI 2.0.20220426.0 x...
image_id: ami-02541b8af977f6cdd
image_name: amzn2-ami-kernel-5.10-hvm-2.0.20220426.0-x86_64...
installation_status: NOT_INSTALLED
installation_status_code: None
installation_status_code_key: None
instance_state: running
instance_type: t2.micro
internal_ip: 172.31.17.166
name: Demo-ASG
org_key: 8X5TJVYWQ
platform: Unix/Linux
platform_details: Linux/UNIX
platform_name: None
platform_version: None
region: us-west-1
security_group_id: [list:1 item]:
[0]: sg-085972ee2f0be60aa
subnet_id: subnet-02ccab8946d24f386
virtual_private_cloud_id: vpc-0faa4803c3de51c87

```

Fetch Compute Resource by ID

Using a query of the `VCenterComputeResource` or `AWSComputeResource` objects, you can get the compute resource by ID from your organization.

Example (vCenter workloads):

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import VCenterComputeResource

>>> # This is an example id that we want to query
>>> id = 15054425

>>> cbc = CBCloudAPI()
>>> query = cbc.select(VCenterComputeResource, id)

>>> # A string object is returned here, so we can print the result directly.
>>> print(query)
```

VCenterComputeResource object, bound to https://defense-dev01.cbdtest.io.

Last refreshed at Mon Mar 1 12:02:14 2021

```
-----

  appliance_uuid: c89f183b-f201-4bca-bacc-80184b5b8823
  cluster_name: example-cluster-name
  created_at: 2020-11-18T07:41:16.834Z
  datacenter_name: None
  eligibility: NOT_ELIGIBLE
  eligibility_code: ['Launcher not found']
  esx_host_name: 10.105.7.129
  esx_host_uuid: bb8d2842-0438-9a74-7964-1d0efad10f28
  host_name: localhost.localdomain
  id: 15054425
  installation_status: NOT_INSTALLED
  installation_status_code: None
  ip_address: 10.105.7.201
  name: CB-ServiceTest
  os_architecture: 64
  os_description: CentOS 7 (64-bit)
  os_type: CENTOS
  uuid: 5022227f-947a-84f8-5816-747f5e18e5ac
  vcenter_host_url: 10.105.5.63
  vcenter_name: VMware vCenter Server 7.0.0 build-15952599
  vcenter_uuid: 4a6b1382-f917-4e1a-8564-374cb7274bd7
  vmwaretools_version: 10336
```

Example (AWS workloads):

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import AWSComputeResource

>>> # This is an example id that we want to query
>>> id = '8x5tjvywq-aws-i-043de738ce129b77a'
```

(continues on next page)

(continued from previous page)

```
>>> cbc = CBCloudAPI()
>>> query = cbc.select(AWSComputeResource, id)

>>> # A string object is returned here, so we can print the result directly.
>>> print(query)
AWSComputeResource object, bound to https://defense-dev01.cbdtest.io.
Last refreshed at Wed Oct 12 11:11:41 2022
-----

    auto_scaling_group_name: Demo-AutoScalingGroup
    availability_zone: us-west-1c
    cloud_provider_account_id: 267678331262
    cloud_provider_resource_id: i-043de738ce129b77a
    cloud_provider_tags: [list:4 items]:
                        [0]: Name##Demo-ASG
                        [1]: aws:ec2launchtemplate:id##lt-0e3d35dba4f5ba16f
                        [2]: aws:autoscaling:groupName##Demo-AutoScalingGroup
                        [...]
    create_time: 2022-06-02T05:23:27Z
    deployment_type: AWS
    eligibility: NOT_ELIGIBLE
    eligibility_code: [list:1 item]:
                    [0]: SSM_DOC_NOT_INSTALLED
    external_ip: 18.144.80.202
    id: 8x5tjvywq-aws-i-043de738ce129b77a
    image_description: Amazon Linux 2 Kernel 5.10 AMI 2.0.20220426.0 x...
    image_id: ami-02541b8af977f6cdd
    image_name: amzn2-ami-kernel-5.10-hvm-2.0.20220426.0-x86_64...
    installation_status: NOT_INSTALLED
    installation_status_code: None
    installation_status_code_key: None
    instance_state: running
    instance_type: t2.micro
    internal_ip: 172.31.11.73
    name: Demo-ASG
    org_key: 8X5TJVVWQ
    platform: Unix/Linux
    platform_details: Linux/UNIX
    platform_name: None
    platform_version: None
    region: us-west-1
    security_group_id: [list:1 item]:
                    [0]: sg-085972ee2f0be60aa
    subnet_id: subnet-03cb2d09e07350698
    virtual_private_cloud_id: vpc-0faa4803c3de51c87
```

Facet Compute Resources

Any compute resource search may be turned into a *faceting* by calling the `facet()` method on the query object returned by `select()`, after setting search criteria. A faceting breaks down each specified field for all compute resources matching the criteria, showing which values that field can take and how many times that field value shows up in the matching compute resources. Only a subset of fields can be faceted on, as listed here:

For VCenterComputeResource:

- `eligibility`
- `installation_status`
- `vmwaretools_version`
- `os_type`

For AWSComputeResource:

- `auto_scaling_group_name`
- `cloud_provider_tags`
- `platform`
- `platform_details`
- `virtual_private_cloud_id`

Example (vCenter workloads):

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import VCenterComputeResource
>>> cbc = CBCloudAPI()
>>> query = cbc.select(VCenterComputeResource)
>>> facets = query.facet(['os_type', 'eligibility'])
>>> for facet in facets:
...     print facet
...
ComputeResourceFacet object, bound to https://defense-dev01.cbdtest.io.
```

```
-----

field: os_type
id: os_type
values: [list:6 items]:
  [0]: [ComputeResourceFacetValue object]:
        id: OTHER
        name: OTHER
        total: 230

  [1]: [ComputeResourceFacetValue object]:
        id: UBUNTU
        name: UBUNTU
        total: 68

  [2]: [ComputeResourceFacetValue object]:
        id: WINDOWS
        name: WINDOWS
        total: 46
```

(continues on next page)

(continued from previous page)

```

[...]
ComputeResourceFacet object, bound to https://defense-dev01.cbdtest.io.
-----

```

```

    field: eligibility
    id: eligibility
values: [list:3 items]:
    [0]: [ComputeResourceFacetValue object]:
        id: NOT_ELIGIBLE
        name: NOT_ELIGIBLE
        total: 237

    [1]: [ComputeResourceFacetValue object]:
        id: UNSUPPORTED
        name: UNSUPPORTED
        total: 185

    [2]: [ComputeResourceFacetValue object]:
        id: ELIGIBLE
        name: ELIGIBLE
        total: 25

```

Example (AWS workloads):

```

>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import AWSComputeResource
>>> cbc = CBCloudAPI()
>>> query = cbc.select(AWSComputeResource)
>>> facets = query.facet(['platform', 'virtual_private_cloud_id'])
>>> for facet in facets:
...     print facet
...
ComputeResourceFacet object, bound to https://defense-dev01.cbdtest.io.
-----

```

```

    field: virtual_private_cloud_id
    id: virtual_private_cloud_id
values: [list:8 items]:
    [0]: [ComputeResourceFacetValue object]:
        id: vpc-02371233d7ac6d33c
        name: vpc-02371233d7ac6d33c
        total: 28

    [1]: [ComputeResourceFacetValue object]:
        id: vpc-5102d53a
        name: vpc-5102d53a
        total: 12

    [2]: [ComputeResourceFacetValue object]:
        id: vpc-0968a1d4ea101fc26
        name: vpc-0968a1d4ea101fc26

```

(continues on next page)

(continued from previous page)

```

        total: 7

    [...]
ComputeResourceFacet object, bound to https://defense-dev01.cbdtest.io.
-----

    field: platform
    id: platform
values: [list:2 items]:
    [0]: [ComputeResourceFacetValue object]:
        id: Unix/Linux
        name: Unix/Linux
        total: 56

    [1]: [ComputeResourceFacetValue object]:
        id: Windows
        name: Windows
        total: 5

```

Download Compute Resource Listings

The details of compute resources matching a search may be directly downloaded from the Carbon Black Cloud by calling the `download()` method on the query object returned by `select()`, after setting search criteria. The format for downloading may be specified as either JSON or CSV.

The `download()` method returns a Job object, which is processed asynchronously and from which the results are available once the job has been completed.

Example (vCenter workloads):

```

>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import VCenterComputeResource
>>> cbc = CBCloudAPI()
>>> query = cbc.select(VCenterComputeResource).set_os_type(["UBUNTU"]).set_eligibility([
↳ "ELIGIBLE"])
>>> query.set_installation_status(["ERROR"])
>>> job = query.download("CSV")
>>> job.await_completion()
>>> print(job.get_output_as_string())
Eligibility,Install Status,Name,OS,VMware Tools,Added Time,VM ID,VM name,IP address,
↳ Datacenter,Cluster,vCenter [...]
"ELIGIBLE","ERROR","", "wdc-10-180-200-134", "UBUNTU", "10336", "2021-07-27T11:01:01.636",
↳ "776bf589-923e-4ccd-869d-[]"
"ELIGIBLE","ERROR","", "UBUNTU", "0", "2021-11-19T08:49:20.882", "50294288-5baa-6e71-18f0-
↳ 71c8a17f0caf", "POC-DB-[]"
"ELIGIBLE","ERROR","", "ubunt1804desktop", "UBUNTU", "10338", "2022-04-04T04:54:50.861",
↳ "503410f6-80aa-1f69-0285-[]"
"ELIGIBLE","ERROR","", "ubunt1804desktop", "UBUNTU", "10338", "2022-02-28T09:22:32.235",
↳ "503410f6-80aa-1f69-0285-[]"
>>> # note: lines truncated in above output for formatting purposes

```

Example (AWS workloads):

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import AWSComputeResource

>>> cbc = CBCloudAPI()
>>> query = cbc.select(AWSComputeResource).set_region(['us-west-1'])
>>> job = query.download("CSV")
>>> job.await_completion()
>>> print(job.get_output_as_string())
Instance ID,Platform,Account ID,VPC ID,Added Time,AWS Tags,ASG,Instance Type,Image ID,
↳ Image name,Image [...]
"i-043de738ce129b77a","Unix/Linux","267678331262","vpc-0faa4803c3de51c87","2022-06-
↳ 02T05:23:27",[...]
"i-0febda35fcdf2dbd1","Unix/Linux","267678331262","vpc-0faa4803c3de51c87","2022-07-
↳ 11T08:26:58",[...]
"i-0b8b62d7c3aea1f9f","Unix/Linux","267678331262","vpc-0faa4803c3de51c87","2022-06-
↳ 02T05:21:26",[...]
>>> # note: lines truncated in above output for formatting purposes
```

Summarize Compute Resources

Note: This functionality is not available for vCenter compute resources.

By calling the `summarize()` method on the query object returned by `select()`, after setting search criteria, a summary of compute resources may be generated. The fields which may be summarized are as follows:

For AWSComputeResource:

- `availability_zone`
- `region`
- `subnet_id`
- `virtual_private_cloud_id`
- `security_group_id`

Example (AWS workloads):

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import AWSComputeResource
>>> cbc = CBCloudAPI()
>>> query = cbc.select(AWSComputeResource)
>>> summary = query.summarize(['availability_zone', 'region', 'virtual_private_cloud_id
↳ '])
>>> print(summary)
{'region': 5, 'availability_zone': 12, 'virtual_private_cloud_id': 8}
```

Interactive example script featuring Workloads Search

We have a number of example scripts you can use with the CBC SDK.

This interactive script highlights the capabilities of the CBC SDK. It uses user input to guide you through the functionalities of the Workloads Search.

You can download it from: [here](#)

- *Searching* - Most operations in the SDK will require you to search for objects.
- *Alerts* - Work and manage different types of alerts such as CB Analytics Alert, Watchlist Alerts and Device Control Alerts.
- *Asset Groups* - Create and modify Asset Groups, and preview the impact changes to policy ranking or asset group definition will have.
- *Alert Migration* - Update from SDK 1.4.3 or earlier to SDK 1.5.0 or later to get the benefits of the Alerts v7 API.
- *Audit Log Events* - Retrieve audit log events indicating various “system” events.
- *Device Control* - Control the blocking of USB devices on endpoints.
- *Differential Analysis* - Provides the ability to compare and understand the changes between two Live Query runs
- *Live Query* - Live Query allows operators to ask questions of endpoints
- *Live Response* - Live Response allows security operators to collect information and take action on remote endpoints in real time.
- *Notifications to Alerts Migration* - Update from Notifications to Alerts in SDK 1.5.0 or later to get the benefits of the Alerts v7 API.
- *Policy* - Use policies to define and prioritize rules for how applications can behave on groups of assets
- *Recommendations* - Work with Endpoint Standard recommendations for reputation override.
- *Reputation Override* - Manage reputation overrides for known applications, IT tools or certs.
- *Unified Binary Store* - The unified binary store (UBS) is responsible for storing all binaries and corresponding metadata for those binaries.
- *Users and Grants* - Work with users and access grants.
- *Vulnerabilities* - View asset (Endpoint or Workload) vulnerabilities to increase security visibility.
- *Watchlists, Feeds, Reports, and IOCs* - Work with Enterprise EDR watchlists, feeds, reports, and Indicators of Compromise (IOCs).
- *Workloads* - Advanced protection purpose-built for securing modern workloads to reduce the attack surface and strengthen security posture.

4.5.3 Migration Guides

Alert Migration

Use this guide to update from SDK v1.4.3 or earlier (using Alerts v6 API) to SDK v1.5.0 or (Alerts v7 API).

We recommend that customers evaluate the new fields that are available in Alerts v7 API and supported in SDK 1.5.0 onwards to maximize the benefits from the new data. A lot of new metadata is included in the Alert record that can help simplify your integration. For example, if you were previously getting process information to enrich the command line, the process commandline is now included in the Alert record.

Resources

- [Alerts Migration Guide](#)
- [Alerts v7 Announcement](#)
- [Alert Search and Response Fields](#)
- Example script showing breaking and compatibility features `alert_v6_v7_migration.py` in [GitHub Examples](#).
- SDK 1.5.0 Alert Example Script `alerts_common_scenarios.py` in [GitHub Examples](#).

Overview

In SDK 1.5.0, we balance backwards compatibility with making breaking changes apparent to avoid silent integration failures. Such failures might lead to the perception that things continue to work when they do not work.

- Breaking Changes
 - Default Search Time Period is reduced to two weeks. See [Default Search Time Period](#).
 - Fields that do not exist in Alert v7 API: `FunctionalityDecommissioned` exception is raised if called. See [SDK Treatment of Fields that have been removed](#).
 - `get_events()` method has been removed. See [Enriched Events have been Replaced by Observations](#).
 - Facet terms match the field names. See [Facet Terms](#).
 - Workflow is rebuilt. See [Streamlined Alert Workflow](#).
 - Create Note returns a single `Note` instance instead of a list. See [create_note\(\) return type](#).
- Backwards compatibility:
 - Class name change: Alert replaces `BaseAlert`, but `BaseAlert` is retained. See [Class Name Changes](#).
 - Field name changes: The previous name is aliased to the new name on get, set, and access by property name. See [Field names aliased](#).
 - The single field port is separated into local and remote fields. See [Port - split into local and remote](#).

New Features

Enjoy all the new features!

See an example script that demonstrates the SDK 1.5.0 features in [GitHub Examples](#), `alerts_common_scenarios.py`.

- New metadata fields include command lines. View the new fields and identify which fields can be used in criteria, exclusions, and as a facet term on the [Developer Network Alerts Search Fields](#).
- `add_exclusions()`: This new method exposes the exclusion element. Any records that match these values are excluded from the result set.
- `get_observations()`: Gets the Observations that are related to the alert. This feature is available for most Alert types.
- `get_process()`: This method previously got the process related to a Watchlist Alert. It is extended to get processes for other Alert Types if the Alert has a `process_guid` set.
- Notes can be added to an Alert or a Threat.
- Alert History can be retrieved.

- `to_json(version)` is a new method that returns the alert object in json format.
 - This method has been added to replace the use of the `_info` attribute because it is an internal representation.
 - If no version parameter is provided, the version will default to API version v7.
 - “v6” can be passed as a parameter and the attribute names will be translated to the Alert v6 names.
 - `to_json("v6")` translates field names from the v7 field name to v6 field names and returns a structure as close to v6 (SDK 1.4.3) as possible. The fields that do not have equivalents in the v7 API will be omitted.
 - The `to_json` method is intended to ease the update path if the `_info` attribute was being used.
 - Example method: `show_to_json(api)`.

The following code snippet shows how to call the `to_json` method for an alert:

```
>>> cb = get_cb_cloud_object(args)
>>> alert_query = cb.select(Alert)
>>> alert = alert_query.first()
>>> v7_dict = alert.to_json()
>>> v6_dict = alert.to_json("v6")
```

The returned object `v7_dict` will have a dictionary representation of the alert using v7 attribute names and structure.

The returned object `v6_dict` will have a dictionary representation of the alert using v6 attribute names and structure. If the field does not exist in v7, the field will be omitted from the json representation.

Breaking Changes

The following changes require integration updates to avoid using functionality that is no longer available.

The “Example Method” refers to the example script `alert_v6_v7_migration.py` in [GitHub](#).

Default Search Time Period

The default search period was one month. The default search period is now two weeks.

- The SDK does not make any compensating changes for this change of time.
- Example method: `base_class_and_default_time_range(api)`.

The following snippet shows how to set the search window to the previous month. See the Developer Network for details on the [Time Range Filter](#)

```
>>> alerts = api.select(Alert).set_time_range(range="-1M")
```

SDK Treatment of Fields that have been removed

Some fields from the Alert API v6 (SDK 1.4.3 and earlier) do not have an equivalent in Alert v7 API (SDK 1.5.0+). A `FunctionalityDecommissioned` exception will be raised if they are used.

See [Removed Fields](#) for a list of these fields.

We recommend that you do the following:

- Review the fields that do not have an equivalent.
- After updating to the SDK 1.5.0, check your integrations for error logs that contain `FunctionalityDecommissioned` exceptions.
- Review the new fields and determine what changes can enhance your use cases.
- Use the `add_criteria` method to search for alerts. This method replaces the hand-crafted `set_<field_name>` methods.
- Example method: `set_methods_backwards_compatibility(api)`.

For [Removed Fields](#), the SDK 1.5.0+ has the following behavior:

- `set_<v6 field name>()` will raise a `FunctionalityDecommissioned` exception.
- `get(<v6 field name>)` will raise a `FunctionalityDecommissioned` exception.
- `alert.field_name` will raise a `FunctionalityDecommissioned` exception.
- Example method: `get_methods_backwards_compatibility(api)` and `category_monitored_removed(api)`.

Details of all changes to API endpoints and fields are in the [Alerts Migration Guide](#) on the Developer Network.

The following code block calls the decommissioned method

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import BaseAlert
>>> api = CBCloudAPI(profile="sample")
>>> alert_query = api.select(BaseAlert).set_blocked_threat_categories(["NON_MALWARE"])
```

It generates the following exception:

```
cbc_sdk.errors.FunctionalityDecommissioned: The set_kill_chain_statuses method does not
↳ exist in in SDK v1.5.0
because kill_chain_status is not a valid field on Alert v7 API. The functionality has
↳ been decommissioned.
```

Similarly, the following code block calls the get attribute function by using the decommissioned attribute: `blocked_threat_categories`:

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import BaseAlert
>>> api = CBCloudAPI(profile="sample")
>>> alert_query = api.select(BaseAlert)
>>> alert = alert_query.first()
>>> alert.get("blocked_threat_category")
```

It generates the following exception:

```
cbc_sdk.errors.FunctionalityDecommissioned:
The Attribute 'blocked_threat_category' does not exist in object 'WatchlistAlert'
↳ because it was
deprecated in Alerts v7. In SDK 1.5.0 the functionality is decommissioned.
```

Removed Fields

Table 1: Field that have been removed from Alert v7 API

Field Name	Alert Types
blocked_threat_category	CB Analytics
category	All
count	Watchlist
document_guid	Watchlist
group_details	All
kill_chain_status	CB Analytics
not_blocked_threat_category	CB Analytics
target_value	Container Runtime
threat_activity_dlp	CB Analytics
threat_activity_phish	CB Analytics
threat_cause_threat_category	All
threat_cause_vector	All
threat_indicators	Watchlist
workload_id	Container Runtime

Enriched Events have been Replaced by Observations

CBAnalytics get_events() is removed.

- The Enriched Events that this method returns have been deprecated.
- Instead, use [Observations](#).
- More information is on the Developer Network Blog, [How to Take Advantage of the New Observations API](#).

Instead of:

```
>>> cb = get_cb_cloud_object(args)
>>> alert_query = cb.select(CBAnalyticsAlert)
>>> alert = alert_query.first()
>>> alert.get_events()
```

Use get_observations. Observations are available for many Alert Types whereas Enriched Events were limited to CB_Analytics Alerts. Watchlist Alerts do not have associated observations, so Alerts of type Watchlist are excluded from the search.

```
>>> alert_query = cb.select(Alert).add_exclusions("type", "WATCHLIST")
>>> alert = alert_query.first()
>>> observations_list = alert.get_observations()
>>> len(observations_list) # execute the query
```

- Example method: `observation_replaces_enriched_event(api)`

Facet Terms

In Alerts v6 API and SDK 1.4.3, the terms available for use in facet requests were very limited and the facet terms did not always match the field name upon which it operated.

In Alerts v7 API and SDK 1.5.0, more fields are available and the facet term matches the field name.

- If the term used in v6 is the same as the field in v7, the facet term continues to work
- If the term used in v6 is not the same as v7, a `FunctionalityDecommissioned` exception is raised.
 - Raising the exception was a conscious decision to reduce the complexity and ongoing maintenance effort in the SDK, and to ensure visibility to customers that the Facet capability has significant improvements from which integrations will benefit.
 - Example method: `facet_terms(api)`

The following snippet shows a pre-SDK 1.4.3 facet request and the `FunctionalityDecommissioned` exception that the SDK 1.5.0 SDK generates.

```
>>> from cbc_sdk.errors import FunctionalityDecommissioned
>>> try:
...     print("Calling facets with invalid term.")
...     facet_list = api.select(BaseAlert).facets(["ALERT_TYPE"])
... except FunctionalityDecommissioned as e:
...     print(e)
...
Calling facets with invalid term.
The Field 'ALERT_TYPE' is not a valid facet name because it was deprecated in Alerts v7.
↳ functionality has been decommissioned.
```

The following snippet shows a valid request and printed response.

```
>>> import json
>>> facet_list = api.select(Alert).facets(["policy_applied", "attack_technique"])
>>> print("This is a valid facet response: {}".format(json.dumps(facet_list, indent=4)))
This is a valid facet response: [
  {
    "field": "attack_technique",
    "values": [
      {
        "total": 2,
        "id": "T1048.002",
        "name": "T1048.002"
      },
      {
        "total": 1,
        "id": "T1490",
        "name": "T1490"
      }
    ]
  },
  {
    "field": "policy_applied",
    "values": [
      {
```

(continues on next page)

(continued from previous page)

```

        "total": 69224,
        "id": "NOT_APPLIED",
        "name": "NOT_APPLIED"
    },
    {
        "total": 450,
        "id": "APPLIED",
        "name": "APPLIED"
    }
]
}
]
```

Streamlined Alert Workflow

The Alert Closure workflow is updated to be more streamlined and improves Alert lifecycle management.

The workflow leverages the alert search structure to specify the alerts to close and has the following status’:

- **Open**, the initial status
- **In Progress**, a new intermediate status
- **Closed** which replaces *Dismissed*

As a result of the underlying change, the workflow does not have backwards compatibility built into it. The new workflow is:

1. Use an Alert Search to specify which Alerts will have their status updated.
 - The request body is a search request and all alerts matching the request will be updated.
 - Two common uses are to update one alert or to update all alerts that have a specific threat id.
 - Any search request can be used as the criteria to select alerts to update the alert status.

```

>>> # This query selects only the alert that has the specified id:
>>> ALERT_ID = "id of the alert to close"
>>> alert_query = api.select(Alert).add_criteria("id", [ALERT_ID])
>>> # This query selects all alerts that have the specified threat id. It_
↪is not used again in this example
>>> alert_query_for_threat = api.select(Alert).add_criteria("threat_id",
↪"CFED0B211ED09F8EC1C83D4F3FBF1709")
```

2. Submit a job to update the status of Alerts.
 - The status can be OPEN, IN PROGRESS or CLOSED (previously DISMISSED).
 - You can include a Closure Reason.
3. The immediate response confirms that the job was successfully submitted.
4. Use the Job() `cbc_sdk.platform.jobs.Job` class to determine when the update is complete.

Use the Job object to wait until the Job has completed. Your python script will wait while the SDK manages the polling to determine when the job is complete.

```
>>> job.await_completion().result()
```

5. Refresh the Alert Search to get the updated alert data into the SDK.

```
>>> alert.refresh()
>>> print("Status = {}, Expecting CLOSED".format(alert.workflow["status"]))
```

6. The Dismissal of Future Alerts for the same threat id has not changed.

The following sequence of calls updates future alerts that have the same threat id. It is usually used in combination with the alert closure; that is, you can use it to dismiss future alerts call to close future occurrences and call alert closure to close current open alerts that have the threat id.

```
>>> alert_threat_query = api.select(Alert).add_criteria("threat_id",
↳ "CFED0B211ED09F8EC1C83D4F3FBF1709")
>>> alert.dismiss_threat("threat remediation done", "testing dismiss_threat_
↳ in the SDK")
>>> # To undo the dismissal, call update
>>> alert.update_threat("threat remediation un-done", "testing update_
↳ threat in the SDK")
```

create_note() Return Type

alert.create_note() returns a Note object instead of a list.

```
>>> alert_query = api.select(Alert)
>>> alert = alert_query.first()
>>> new_note = alert.create_note("Adding note from SDK with current timestamp: {}".
↳ format(time.time()))
>>> print(type(new_note))
<class 'cbc_sdk.platform.alerts.Alert.Note'>
```

Backwards Compatibility

The following changes have code in the SDK to map updated functionality to previous SDK functions. The SDK will continue to work, but new features should be reviewed to enhance integration and automation.

The “Example Method” refers to the example script `alert_v6_v7_migration.py` in [GitHub](#).

Class Name Changes

- The base class for Alerts in the SDK has changed from `BaseAlert` to `Alert`.
 - Backwards compatibility is retained.
 - Example method: `base_class_and_default_time_range(api)`.

Field Names Aliased

To align with other parts of Carbon Black Cloud and industry conventions, many fields were deprecated from Alerts API v6 and have equivalent fields using a different name in v7. In the SDK v1.5.0, aliases are in place to minimize breaks.

Details of all changes to API endpoints and fields are in the [Alerts Migration Guide](#) on the Developer Network.

`set_<v6 field name>()` on the query object translates to the new field name for the request.

- Update to use `add_criteria(field_name, [field_value])`.
- You can use many new fields in criteria to search Alerts using `add_criteria`, but do not have `set_<field_name>` methods.
- Example method: `set_methods_backwards_compatibility(api)`.

`get_<v6 field name>()` translates to the new field name to look up the value.

- Example method: `get_methods_backwards_compatibility(api)`.

`alert.field_name` translates the field name to the new name and returns the matching value.

- Example method: `set_methods_backwards_compatibility(api)`.

The following fields have a new name in Alert v7 and the new field name contains the same value.

Table 2: Field mappings where the field has been renamed

Alert v6 API - SDK 1.4.3 or earlier	Alert v7 API - SDK 1.5.0 or later
cluster_name	k8s_cluster
create_time	backend_timestamp
first_event_time	first_event_timestamp
last_event_time	last_event_timestamp
last_update_time	backend_update_timestamp
namespace	k8s_namespace
notes_present	alert_notes_present
policy_id	device_policy_id
policy_name	device_policy
port	netconn_local_port
protocol	netconn_protocol
remote_domain	netconn_remote_domain
remote_ip	netconn_remote_ip
remote_namespace	remote_k8s_namespace
remote_replica_id	remote_k8s_pod_name
remote_workload_kind	remote_k8s_kind
remote_workload_name	remote_k8s_workload_name
replica_id	k8s_pod_name
rule_id	rule_id
run_state	run_state
target_value	device_target_value
threat_cause_actor_certificate_authority	process_issuer
threat_cause_actor_name	process_name. Note that <i>threat_cause_actor_name</i> was only the name of the executable. <i>process_name</i> contains the full path.
threat_cause_actor_publisher	process_publisher
threat_cause_actor_sha256	process_sha256
threat_cause_cause_event_id	primary_event_id

continues on next page

Table 2 – continued from previous page

Alert v6 API - SDK 1.4.3 or earlier	Alert v7 API - SDK 1.5.0 or later
threat_cause_md5	process_md5
threat_cause_parent_guid	parent_guid
threat_cause_reputation	process_reputation
threat_indicators	ttps
watchlists	watchlists.id
workflow.last_update_time	workflow.change_timestamp
workload_kind	k8s_kind
workload_name	k8s_workload_name

Port - split into local and remote

- In SDK 1.4.3 and earlier, there was a single field port.
- In Alerts v7 API and SDK 1.5.0, there are two fields; `netconn_local_port` and `netconn_remote_port`.
- The legacy method `set_ports()` sets the criteria for `netconn_local_port`.

```
>>> # This legacy search request:  
>>> api.select(BaseAlert).set_ports(["NON_MALWARE"])
```

Migration Guide For Live Response From v3 To v6

This guide will help you migrate from Live Response v3 to v6.

Overview

Most of the changes from v3 to v6 are on the routes. The updated API (v6) includes a more granular approach to roles-based access control (RBAC).

This change was implemented in CBC SDK 1.3.0, Released June 8, 2021. If you are on a more recent version of this SDK, you are already using the new version.

Access Permissions

A key with a Custom Access Level with appropriate permissions needs to be created for the Live Response. The following table shows the corresponding permissions that needs to be enabled, based on the existing roles.

Permission	What it controls (commands)	Which existing roles have access
org.liveresponse	Permanently disabling the Live Response feature on an individual endpoint: Disable Live Response on the Endpoints page	Level 3 Analyst Live Response Admin - Legacy Super Admin
org.liveresponse.files	Read, write and/or delete files on the endpoint: cd, delete, dir, drives, get, mkdir, put, pwd	Level 2 Analyst Level 3 Analyst Live Response Admin - Legacy Super Admin
org.liveresponse.memdump	Dump kernel memory on the endpoint: memdump	Level 3 Analyst Live Response Admin - Legacy Super Admin
org.liveresponse.process	List, stop and execute processes on the endpoint: exec, execfg, kill, ps	Level 2 Analyst (cannot execute) Level 3 Analyst Live Response Admin - Legacy Super Admin
org.liveresponse.registry	View, add, edit and delete registry entries: reg add, reg delete, reg query, reg set	Level 2 Analyst Level 3 Analyst Live Response Admin - Legacy Super Admin
org.liveresponse.session	Initiate live response sessions, plus: clear, help	Level 2 Analyst Level 3 Analyst Live Response Admin - Legacy Super Admin

Changes in the routes and response codes

v3	v6
/integrationServices/v3/cblr/	/appservices/v6/orgs/{org_key}/liveresponse/
POST /sessions/{session_id} 200	POST /sessions 201
POST /session/{session_id}/file 200	POST /sessions/{session_id}/files 201
POST /session/{session_id}/command	POST /sessions/{session_id}/commands
PUT /session {"session_id": "1:37191", "status": "CLOSE"}	DELETE /sessions/{session_id} 204
GET /session/{sessionId}/file/{fileId}/content 200	GET /sessions/{session_id}/files/{file_id}/content 302
DELETE /session/{sessionId}/file/{fileId} 200	DELETE /sessions/{session_id}/files/{file_id} 204

Changes in some of the request/response fields

Where is the change?	v3	v6
All API endpoints	sensor_id	device_id
Process command	username	process_username
Process command	path	process_path
Process command	pid	process_pid
Process command	command_line	process_cmdline
Process command	parent	parent_pid
Registry command	valueType	value_type
Registry command	valueData	value_data
Registry command	valueName	value_name

Additional Information

- (CBC) Live Response API releasing v6: now with granular RBAC!
- [Live Response Documentation](#)
- [Live Response API Migration Guide](#)

Notifications to Alerts Migration

Use this guide to update from using `get_notifications()`, which leverages the `/integrationServices/v3/notification` API to using Alerts in SDK v1.5.0 or higher with Alerts v7 API.

Note: The `/integrationServices/v3/notification` API is deprecated, and deactivation is planned for 31 October 2024.

The Access Level Type `SIEM` used to access the Notifications API is also deprecated. Deactivation of the legacy access level type `SIEM` is planned for 31 January 2025.

For more information about migrating from the API and alternative solutions, see [IntegrationService notification v3 API Migration Guide](#)

The key differences between Notifications and Alerts are:

- In Notifications, the criteria that defines when a notification is sent is defined in the Carbon Black Cloud console. When using the Alerts v7 API, the criteria is part of the API request
- Notifications work on a subscription-based principle and they require a SIEM authentication key. By using that key, you are subscribing to a certain criteria of alerts.
- As the API Notification API is deprecated, new alert types such as Intrusion Detection System Alerts cannot be retrieved from the Notifications API.
- The Notifications endpoint is a read-once queue whereas the Alerts v7 is a search request. When calling the Alerts v7 API, the caller (your script) must manage state, keeping track of the timestamp of the last Alert retrieved and using that for the start timestamp on the next request. See the Alert Bulk Export guide for details on the polling algorithm.

We recommend that customers evaluate the new fields that are available in Alerts v7 API and supported in SDK 1.5.0 onwards to maximize the benefits from the new data. A lot of new metadata is included in the Alert record that can help simplify your integration. For example, if you were previously getting process information to enrich the command line, the process commandline is now included in the Alert record.

As at SDK 1.5.0, Notifications are deprecated and functional; there has not been a breaking change. The underlying API will be deactivated on October 31, 2024 so you must move to Alerts in SDK 1.5.0 or newer which uses Alerts v7 API, or to the [Data Forwarder](#) with Alert Schema 2.x before then.

Resources

- [IntegrationServices Notification v3 API Migration](#)
- [Carbon Black Cloud Syslog Connector 2.0](#)
- [Alert Bulk Export](#)
- [Alerts Migration Guide](#)
- [Alerts v7 Announcement](#)
- [Alert Search and Response Fields](#)
- [SDK 1.5.0 Alert Example Script `alerts_common_scenarios.py` in GitHub Examples.](#)
- [Alerts Bulk Export Example Script `alerts_bulk_export.py` in GitHub Examples.](#)

How to Update the SDK Usage

This screen shot shows the Notification configuration page in the Carbon Black Cloud console.

Add Notification

*

Name

SDK SIEM Demo

When do you want to be notified?

Alert crosses a threshold

Alert severity

7+

As of June 2023, Observed activity alerts have been transitioned to the Investigate page as Observations. Existing notification rules containing observed activity alerts will continue to receive emails at this time, but new or edited rules will no longer have this option.

*

Alert types

All types

Select types

CB Analytics

Watchlists

USB Device Control

Containers Runtime

Host Based Firewall

Intrusion Detection System (email only)

*

Policy

All policies

Select policies

Standard

Search for a policy

If multiple policies are selected, a separate notification will be created for each

How do you want to be notified?

Email

Search for a user

Send only 1 email notification for each threat type per day

API Key

SDK SIEM Demo (ABSP987P3V)

Search for an API key

Save

Cancel

You can replicate the settings shown in the screenshot by running the following search on Alerts:

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.platform import Alert
>>> alerts = api.select(Alert).set_minimum_severity(7).\
>>>     add_criteria("type", ["CB_ANALYTICS", "DEVICE_CONTROL"]).\
>>>     add_criteria("device_policy", "Standard")
```

An Alert contains a lot more information than a Notification, and most of the fields are available for searching.

The other modification required is that where the Notifications was a read one queue, Alerts are retrieved using a search. An example script with the polling logic implemented is in the GitHub Repository, [alerts_bulk_export.py](#) in [GitHub Examples](#).

There is also a guide to [Alert Bulk Export](#) on the developer network with a detailed explanation of the logic.

Porting Applications from CBAPI to Carbon Black Cloud SDK

This guide will help you migrate from CBAPI to the Carbon Black Cloud Python SDK.

This is necessary to take advantage of new functionality in Carbon Black Cloud and also to ensure that functionality is not lost from your integrations when APIs are deactivated in July 2024. Read more about the new features in the [Developer Network Blogs](#).

Note: CBAPI applications using Carbon Black EDR (Response) or Carbon Black App Control (Protection) cannot be ported, as support for on-premise products is not present in the CBC SDK. Continue to use CBAPI for these applications.

Overview

CBC SDK has changes to package names, folder structure, and functions. Import statements will need to change for the packages, modules, and functions listed in this guide.

Package Name Changes

A number of packages have new name equivalents in the CBC SDK. Endpoint Standard and Enterprise EDR have had parts replaced to use the most current API routes.

Top-level Package Name Change

The top-level package name has changed from CBAPI to CBC SDK.

CBAPI Name (old)	CBC SDK Name (new)
cbapi.psc	cbc_sdk

Product Name Changes

Carbon Black Cloud product names have been updated in the SDK.

CBAPI Name (old)	CBC SDK Name (new)
<code>cbapi.psc.defense</code>	<code>cbc_sdk.endpoint_standard</code>
<code>cbapi.psc.livequery</code>	<code>cbc_sdk.audit_remediation</code>
<code>cbapi.psc.threathunter</code>	<code>cbc_sdk.enterprise_edr</code>
<code>cbapi.psc</code>	<code>cbc_sdk.platform</code>

Features for new products such as Container Security and Workload Security have also been added in the appropriate namespace.

APIs that have been deprecated or deactivated

Some modules made use of APIs that have been deactivated and are either no longer included in the Carbon Black Cloud, or are planned for deprecation in the second half of 2024. The following table shows the original module, the replacement module, and where to find more information.

For a complete list of APIs that are deprecated and the associated migration information, see the [Migration Guide](#) on the Developer Network. This is important if you have integrations with Carbon Black Cloud that do not use the Carbon Black Cloud Python SDK (this).

Table 3: Deprecated Modules and their replacements

CBAPI module	Replacement CBC SDK Module	More Information
<code>cbapi.psc.defense Event</code>	<code>cbc_sdk.platform Observation</code>	This was deactivated in January 2021. Review the Carbon Black Cloud User Guide to learn more about Observations
<code>cbapi.psc.defense Policy</code>	<code>cbc_sdk.platform Policy</code>	IntegrationServices Policy v3 API Migration
<code>cbc_sdk.endpoint_standard EnrichedEvent</code>	<code>cbc_sdk.platform Observation</code>	Enriched Events will remain available until July 2024. Enriched Events API Migration
<code>cbc_sdk.platform Alert</code>	Module path is unchanged. Attributes and methods will change	In SDK 1.5.0 the Alert module will be updated to use the new Alert v7 API. A migration guide will be included with that release. Planned for October 2023.
SIEM Notifications - <code>cbc_sdk.rest_api CBCloudAPI get_notifications()</code>	<code>cbc_sdk.platform Alert</code> or <code>Alert Data Forwarder</code>	Notification Migration

Modules that have been moved and need new import statements

Import statements will need to change:

```
# Endpoint Standard (Defense)

# CBAPI
from cbapi.psc.defense import Device

# CBC SDK
from cbc_sdk.platform import Device


# Audit and Remediation (LiveQuery)

# CBAPI
from cbapi.psc.livequery import Run, RunHistory, Result, DeviceSummary

# CBC SDK
from cbc_sdk.audit_remediation import Run, RunHistory, Result, DeviceSummary


# Enterprise EDR (ThreatHunter)

# CBAPI
from cbapi.psc.threathunter import Feed, Report, Watchlist

# CBC SDK
from cbc_sdk.enterprise_edr import Feed, Report, Watchlist
```

Moved Packages and Models

Some modules have been moved to a more appropriate location.

CBAPI Name (old)	CBC SDK Name (new)
cbapi.example_helpers	cbc_sdk.helpers
cbapi.psc.alerts_query	cbc_sdk.platform
cbapi.psc.devices_query	cbc_sdk.platform

Import statements will need to change:

```
# Example Helpers

# CBAPI
from cbapi.example_helpers import build_cli_parser

# CBC SDK
from cbc_sdk.helpers import build_cli_parser

# Alerts
```

(continues on next page)

(continued from previous page)

```
# CBAPI
from cbapi.psc.alerts_query import *

# CBC SDK
from cbc_sdk.platform import *

# Devices

# CBAPI
from cbapi.psc.devices_query import *

# CBC SDK
from cbc_sdk.platform import *
```

Replaced Modules

In 2020, Carbon Black Cloud APIs were updated to provide a more consistent search experience. Platform search replaced Endpoint Standard Event searching, and Enterprise EDR Process and Event searching.

For help beyond import statement changes, check out these resources:

- [Unified Platform Experience: What to Expect](#)
- [Migration Guide: Carbon Black Cloud Events API](#)
- [Advanced Search Tips for Carbon Black Cloud Platform Search](#)

Endpoint Standard

Endpoint Standard Events have been replaced with Platform Observations and the old event functionality has been decommissioned:

```
# Endpoint Standard Enriched Events

# CBAPI
from cbapi.psc.defense import Event

# CBC SDK - decommissioned--do not use
from cbc_sdk.endpoint_standard import Event

# CBC SDK - deprecated--stop using before July 31st 2024
from cbc_sdk.endpoint_standard import EnrichedEvent

# CBC SDK - Observations. Use this!
from cbc_sdk.platform import Observation
```

Enterprise EDR

Enterprise EDR Processes and Events have been removed and replaced with Platform Processes and Events:

```
# Enterprise EDR Process and Event

# CBAPI
from cbapi.psc.threathunter import Process, Event

# CBC SDK
from cbc_sdk.platform import Process, Event
```

Folder Structure Changes

The directory structure for the SDK has been refined compared to CBAPI.

- Addition of the Platform folder
- Removal of Response and Protection folders
- Consolidation of model objects and query objects
- Product-specific `rest_api.py` files replaced with package level `rest_api.py`
 - `from cbapi.psc.threathunter import CbThreatHunterAPI` becomes `from cbc_sdk import CBCloudAPI`, etc.

Directory Tree Changes

In general, each module's `models.py` and `query.py` files were combined into their respective `base.py` files.

CBAPI had the following abbreviated folder structure:

```
src
├── cbapi
│   └── psc
│       ├── defense
│       │   ├── models.py
│       │   │   ├── Device
│       │   │   ├── Event
│       │   │   └── Policy
│       │   ├── rest_api.py
│       │   └── CbDefenseAPI
│       ├── livequery
│       │   ├── models.py
│       │   │   ├── Run
│       │   │   ├── RunHistory
│       │   │   ├── Result
│       │   │   ├── ResultFacet
│       │   │   ├── DeviceSummary
│       │   │   └── DeviceSummaryFacet
│       │   ├── rest_api.py
│       │   └── CbLiveQueryAPI
│       └── threathunter
```

(continues on next page)

(continued from previous page)

```

├── models.py
│   ├── Process
│   ├── Event
│   ├── Tree
│   ├── Feed
│   ├── Report
│   ├── IOC
│   ├── IOC_V2
│   ├── Watchlist
│   ├── ReportSeverity
│   ├── Binary
│   └── Downloads
├── rest_api.py
│   └── CbThreatHunterAPI

```

Each product had a `models.py` and `rest_api.py` file.

CBC SDK has the following abbreviated folder structure:

```

src
├── cbc_sdk
│   ├── audit_remediation
│   │   └── base.py
│   │       ├── Run
│   │       ├── RunHistory
│   │       ├── Result
│   │       ├── ResultFacet
│   │       ├── DeviceSummary
│   │       └── DeviceSummaryFacet
│   ├── endpoint_standard
│   │   └── base.py
│   │       ├── Device
│   │       ├── Event
│   │       ├── Policy
│   │       ├── EnrichedEvent
│   │       └── EnrichedEventFacet
│   ├── enterprise_edr
│   │   ├── base.py
│   │   ├── threat_intelligence.py
│   │   │   ├── Watchlist
│   │   │   ├── Feed
│   │   │   ├── Report
│   │   │   ├── ReportSeverity
│   │   │   ├── IOC
│   │   │   └── IOC_V2
│   │   └── ubs.py
│   │       ├── Binary
│   │       └── Downloads
│   └── platform
│       └── alerts.py
│           ├── WatchlistAlert
│           ├── CBAalyticsAlert
│           └── Workflow

```

(continues on next page)

(continued from previous page)

```

├── WorkflowStatus
├── processes.py
│   ├── Process
│   └── ProcessFacet
├── events.py
│   ├── Event
│   └── EventFacet
├── devices.py
│   └── Device
├── rest_api.py
└── CBCloudAPI.py

```

Now, each product has either a `base.py` file with all of its objects, or categorized files like `platform.alerts.py` and `platform.devices.py`. The package level `rest_api.py` replaced each product-specific `rest_api.py` file.

Function Changes

Helper Functions:

CBAPI Name (old)	CBC SDK Name (new)
<code>cbapi.example_helpers.get_cb_defense_object()</code>	<code>cbapi.cb_sdk.example_helpers.get_cb_defense_object()</code>
<code>example_helpers.get_cb_livequery_object()</code>	<code>cbapi.example_helpers.get_cb_livequery_object()</code>
<code>get_cb_threathunter_object()</code>	<code>cbapi.example_helpers.get_cb_psc_object()</code>
	<code>get_cb_cloud_object()</code>

Audit and Remediation Queries:

CBAPI Name (old)	CBC SDK Name (new)
<code>cb.query(sql_query)</code>	<code>cb.select(Run).where(sql=sql_query)</code>
<code>cb.query_history(query_string)</code>	<code>cb.select(RunHistory).where(query_string)</code>
<code>cb.query(sql_query).policy_ids()</code>	<code>cb.select(Run).policy_id()</code>

API Objects:

CBAPI Name (old)	CBC SDK Name (new)
<code>cbapi.psc.defense.CbDefenseAPI</code>	<code>cbapi.psc.livequery.CbLiveQueryAPI</code>
<code>cbapi.psc.threathunter.CbThreatHunterAPI</code>	<code>cbapi.psc.CbPSCBaseAPI</code>
	<code>cb_sdk.CBCloudAPI</code>

4.6 The CBCloudAPI Object

The CBCloudAPI object is the key object used in working with the Carbon Black Cloud. It represents the connection to the Carbon Black Cloud server, to the specific organization to which you have access. It is used to search for objects representing specific data items on the server, such as devices, alerts, policies, and so forth. It also has a number of utility functions and properties providing access to additional functionality on the server, such as [Live Response](#).

A program using the Carbon Black Cloud SDK will start by creating a CBCloudAPI object, passing it the parameters necessary to authenticate to the server. The authentication parameters may be specified as direct arguments when the object is created, or may be provided by a credential provider (see [Credential Providers Package](#)). This object is then called upon for SDK operations, or passed as a parameter to other SDK functions.

As the CBCloudAPI object relies upon REST calls to the server, it does not hold network connections open, and hence need not be explicitly closed.

4.6.1 CBCloudAPI Creation Examples

Authenticate to the Carbon Black Cloud server with directly-supplied parameters:

```
from cbc_sdk import CBCloudAPI
api = CBCloudAPI(url='https://defense.conferdeploy.net', token='ABCDEFGHJKLMNOPQRSTUVWXYZ/
↳YZ12345678',
                 org_key='ABCD1234')

# as an example, get the list of all watchlist alerts
from cbc_sdk.platform import WatchlistAlert
query = api.select(WatchlistAlert)
alerts_list = list(query)
```

Authenticate to the Carbon Black Cloud server using a profile with the default credential provider:

```
from cbc_sdk import CBCloudAPI
api = CBCloudAPI(profile='my_profile')

# as an example, get the list of all watchlist alerts
from cbc_sdk.platform import WatchlistAlert
query = api.select(WatchlistAlert)
alerts_list = list(query)
```

Authenticate to the Carbon Black Cloud server using a profile supplied by a different credential provider:

```
from cbc_sdk import CBCloudAPI
from cbc_sdk.credentials import KeychainCredentialProvider
creds = KeychainCredentialProvider('keychain-to-use', 'my-username')
api = CBCloudAPI(profile='my_profile', credential_provider=creds)

# as an example, get the list of all watchlist alerts
from cbc_sdk.platform import WatchlistAlert
query = api.select(WatchlistAlert)
alerts_list = list(query)
```


4.6.2 Class Documentation

class `CBCloudAPI(*args, **kwargs)`

Bases: `BaseAPI`

A connection to the Carbon Black Cloud.

The core object for interacting with the Carbon Black Cloud SDK.

Example

```
>>> from cbc_sdk import CBCloudAPI
>>> cb = CBCloudAPI(profile="production")
```

Create a new instance of the `CBCloudAPI` object.

Parameters

- ***args** (*list*) – List of arguments to pass to the API object.
- ****kwargs** (*dict*) – Keyword arguments to pass to the API object.

Keyword Arguments

- **credential_file** (*str*) – The name of a credential file to be used by the default credential provider.
- **credential_provider** (`cbc_sdk.credentials.CredentialProvider`) – An alternate credential provider to use to find the credentials to be used when accessing the Carbon Black Cloud.
- **csp_api_token** (*str*) – The CSP API Token for Carbon Black Cloud.
- **csp_oauth_app_id** (*str*) – The CSP OAuth App ID for Carbon Black Cloud.
- **csp_oauth_app_secret** (*str*) – The CSP OAuth App Secret for Carbon Black Cloud.
- **integration_name** (*str*) – The name of the integration using this connection. This should be specified as a string in the format ‘name/version’
- **max_retries** (*int*) – The maximum number of times to retry failing API calls. Default is 5.
- **org_key** (*str*) – The organization key value to use when accessing the Carbon Black Cloud.
- **pool_block** (*bool*) – True if the connection pool should block when no free connections are available. Default is False.
- **pool_connections** (*int*) – Number of HTTP connections to be pooled for this instance. Default is 1.
- **pool_maxsize** (*int*) – Maximum size of the connection pool. Default is 10.
- **profile** (*str*) – Use the credentials in the named profile when connecting to the Carbon Black Cloud server. Uses the profile named ‘default’ when not specified.
- **proxy_session** (`requests.session.Session`) – Proxy session to be used for cookie persistence, connection pooling, and configuration. Default is `None` (use the standard session).
- **thread_pool_count** (*int*) – The number of threads to create for asynchronous queries. Defaults to 3.

- **timeout** (*float*) – The timeout to use for for API connection requests. Default is `None` (no timeout).
- **token** (*str*) – The API token to use when accessing the Carbon Black Cloud.
- **url** (*str*) – The URL of the Carbon Black Cloud provider to use.

alert_search_suggestions(*query*)

Returns suggestions for keys and field values that can be used in a search.

Parameters

query (*str*) – A search query to use.

Returns

A list of search suggestions expressed as dict objects.

Return type

list[dict]

api_json_request(*method, uri, **kwargs*)

Submit a request to the server.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **method** (*str*) – HTTP method to use.
- **uri** (*str*) – URI to submit the request to.
- ****kwargs** (*dict*) – Additional arguments.

Keyword Arguments

- **data** (*object*) – Body data to be passed to the request, formatted as JSON.
- **headers** (*dict*) – Header names and values to pass to the request.

Returns

Result of the operation, as JSON

Return type

object

Raises

[**ServerError**](#) – If there's an error output from the server.

api_request_iterate(*method, uri, **kwargs*)

Submit a request to the specified URI and iterate over the response as lines of text.

Should only be used for requests that can be expressed as large amounts of text that can be broken into lines.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **method** (*str*) – HTTP method to use.
- **uri** (*str*) – The URI to send the request to.
- ****kwargs** (*dict*) – Additional arguments for the request.

Keyword Arguments

- **data** (*object*) – Body data to be passed to the request, formatted as JSON.

- **headers** (*dict*) – Header names and values to pass to the request.

Yields

str – Each line of text in the returned data.

api_request_stream(*method, uri, stream_output, **kwargs*)

Submit a request to the specified URI and stream the results back into the given stream object.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **method** (*str*) – HTTP method to use.
- **uri** (*str*) – The URI to send the request to.
- **stream_output** (*RawIOBase*) – The output stream to write the data to.
- ****kwargs** (*dict*) – Additional arguments for the request.

Keyword Arguments

- **data** (*object*) – Body data to be passed to the request, formatted as JSON.
- **headers** (*dict*) – Header names and values to pass to the request.

Returns

The return data from the request.

Return type

object

audit_remediation(*sql*)

Run an audit-remediation query.

Parameters

sql (*str*) – The SQL for the query.

Returns

The query object.

Return type

cbc_sdk.base.Query

audit_remediation_history(*query=None*)

Run an audit-remediation history query.

Parameters

query (*str*) – The SQL for the query.

Returns

The query object.

Return type

cbc_sdk.base.Query

bulk_threat_dismiss(*threat_ids, remediation=None, comment=None*)

Dismiss the alerts associated with multiple threat IDs.

The alerts will be left in a DISMISSED state.

Parameters

- **threat_ids** (*list[str]*) – List of string threat IDs.

- **remediation** (*str*) – The remediation state to set for all alerts.
- **comment** (*str*) – The comment to set for all alerts.

Returns

The request ID of the pending request, which may be used to select a WorkflowStatus object.

Return type

str

bulk_threat_update(*threat_ids, remediation=None, comment=None*)

Update the alert status of alerts associated with multiple threat IDs.

The alerts will be left in an OPEN state

Parameters

- **threat_ids** (*list[str]*) – List of string threat IDs.
- **remediation** (*str*) – The remediation state to set for all alerts.
- **comment** (*str*) – The comment to set for all alerts.

Returns

The request ID of the pending request, which may be used to select a WorkflowStatus object.

Return type

str

convert_feed_query(*query*)

Converts a legacy CB Response query to a ThreatHunter query.

Parameters

query (*str*) – The query to convert.

Returns

The converted query.

Return type

str

create(*cls, data=None*)

Creates a new model.

Parameters

- **cls** (*class*) – The model being created.
- **data** (*dict*) – The data to pre-populate the model with. Default None.

Returns

An instance of `cls`.

Return type

object

Examples

```
>>> feed = cb.create(Feed, feed_data)
```

property custom_severities

List of active ReportSeverity instances.

delete_object(uri)

Send a DELETE request to the specified URI.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

uri (*str*) – The URI to send the DELETE request to.

Returns

The return data from the DELETE request, as JSON.

Return type

object

device_background_scan(device_ids, scan)

Set the background scan option for the specified devices.

Parameters

- **device_ids** (*list[int]*) – List of IDs of devices to be set.
- **scan** (*bool*) – True to turn background scan on, False to turn it off.

Returns

The parsed JSON output from the request.

Return type

dict

Raises

[**ServerError**](#) – If the API method returns an HTTP error code.

device_bypass(device_ids, enable)

Set the bypass option for the specified devices.

Parameters

- **device_ids** (*list[int]*) – List of IDs of devices to be set.
- **enable** (*bool*) – True to enable bypass, False to disable it.

Returns

The parsed JSON output from the request.

Return type

dict

Raises

[**ServerError**](#) – If the API method returns an HTTP error code.

device_delete_sensor(device_ids)

Delete the specified sensor devices.

Parameters

device_ids (*list[int]*) – List of IDs of devices to be deleted.

Returns

The parsed JSON output from the request.

Return type

dict

Raises

ServerError – If the API method returns an HTTP error code.

device_quarantine(*device_ids*, *enable*)

Set the quarantine option for the specified devices.

Parameters

- **device_ids** (*list[int]*) – List of IDs of devices to be set.
- **enable** (*bool*) – True to enable quarantine, False to disable it.

Returns

The parsed JSON output from the request.

Return type

dict

Raises

ServerError – If the API method returns an HTTP error code.

device_uninstall_sensor(*device_ids*)

Uninstall the specified sensor devices.

Parameters

device_ids (*list[int]*) – List of IDs of devices to be uninstalled.

Returns

The parsed JSON output from the request.

Return type

dict

Raises

ServerError – If the API method returns an HTTP error code.

device_update_policy(*device_ids*, *policy_id*)

Set the current policy for the specified devices.

Parameters

- **device_ids** (*list[int]*) – List of IDs of devices to be changed.
- **policy_id** (*int*) – ID of the policy to set for the devices.

Returns

The parsed JSON output from the request.

Return type

dict

Raises

ServerError – If the API method returns an HTTP error code.

device_update_sensor_version(*device_ids*, *sensor_version*)

Update the sensor version for the specified devices.

Parameters

- **device_ids** (*list[int]*) – List of IDs of devices to be changed.
- **sensor_version** (*dict*) – New version properties for the sensor.

Returns

The parsed JSON output from the request.

Return type

dict

Raises

ServerError – If the API method returns an HTTP error code.

fetch_process_queries()

Retrieves a list of query IDs, active or complete, known by the ThreatHunter server.

get_auditlogs()

Retrieve queued audit logs from the Carbon Black Cloud Endpoint Standard server.

Note: While this can be used with an ‘API’ key generated in the Carbon Black Cloud console, those key types are officially deprecated. Use a Custom key type with permissions as given here.

Required Permissions:

org,audits(READ)

Deprecated:

Use `AuditLog.getAuditLogs` (from `cbc_sdk.platform`) instead.

Returns

List of dictionary objects representing the audit logs, or an empty list if none available.

Return type

list[dict]

get_notifications()

Retrieve queued notifications (alerts) from the Cb Endpoint Standard server.

Note: This can only be used with a ‘SIEM’ key generated in the Cb Endpoint Standard console.

Deprecated:

Use the Alerts API or the Data Forwarder to get similar notifications.

Returns

List of dictionary objects representing the notifications, or an empty list if none available.

Return type

list[dict]

get_object(uri, query_parameters=None, default=None)

Submit a GET request to the server and parse the result as JSON before returning.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **uri** (*str*) – The URI to send the GET request to.
- **query_parameters** (*dict*) – Parameters for the query.
- **default** (*object*) – What gets returned in the event of an empty response.

Returns

Result of the GET request, as JSON.

Return type

object

get_policy_ruleconfig_parameter_schema(*ruleconfig_id*)

Returns the parameter schema for a specified rule configuration.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **ruleconfig_id** (*str*) – The rule configuration ID (UUID).

Returns

The parameter schema for this particular rule configuration (as a JSON schema).

Return type

dict

Raises

[InvalidObjectError](#) – If the rule configuration ID is not valid.

get_raw_data(*uri*, *query_parameters=None*, *default=None*, ***kwargs*)

Submit a GET request to the server and return the result without parsing it.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **uri** (*str*) – The URI to send the GET request to.
- **query_parameters** (*dict*) – Parameters for the query.
- **default** (*object*) – What gets returned in the event of an empty response.
- ****kwargs** (*dict*) – Additional arguments.

Keyword Arguments

headers (*dict*) – Header names and values to pass to the GET request.

Returns

Result of the GET request.

Return type

object

property live_response

The Live Response session manager object.

It is created if it does not yet exist when this property is read.

notification_listener(*interval=60*)

Continually polls the Cb Endpoint Standard server for notifications (alerts).

Note: This can only be used with a ‘SIEM’ key generated in the Cb Endpoint Standard console.

Deprecated:

Use the Alerts API or the Data Forwarder to get similar notifications.

Parameters

interval (*int*) – Time period to wait in between polls for notifications, in seconds. Default is 60.

Yields

dict – A dictionary representing a notification.

property org_urn

The URN of the current organization, based on the configured `org_key`.

post_multipart(uri, param_table, **kwargs)

Send a POST request to the specified URI, with parameters sent as `multipart/form-data`.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **uri** (*str*) – The URI to send the POST request to.
- **param_table** (*dict*) – A dict of known parameters to the underlying method, each element of which is a parameter name mapped to a dict, which contains elements ‘filename’ and ‘type’ representing the pseudo-filename to be used for the data and the MIME type of the data.
- ****kwargs** (*dict*) – Arguments to pass to the API. Except for “headers,” these will all be added as parameters to the form data sent.

Keyword Arguments

headers (*dict*) – Header names and values to pass to the request.

Returns

The return data from the POST request.

Return type

object

post_object(uri, body, **kwargs)

Send a POST request to the specified URI.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **uri** (*str*) – The URI to send the POST request to.
- **body** (*object*) – The data to be sent in the body of the POST request, as JSON.
- ****kwargs** (*dict*) – Additional arguments for the HTTP POST.

Keyword Arguments

headers (*dict*) – Header names and values to pass to the request.

Returns

The return data from the POST request, as JSON.

Return type

object

process_limits()

Returns a dictionary containing API limiting information.

Examples

```
>>> cb.process_limits()
{'u'status_code': 200, u'time_bounds': {u'upper': 1545335070095, u'lower': 1542779216139}}
```

put_object(uri, body, **kwargs)

Send a PUT request to the specified URI.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **uri** (*str*) – The URI to send the PUT request to.
- **body** (*object*) – The data to be sent in the body of the PUT request.
- ****kwargs** (*dict*) – Additional arguments for the HTTP PUT.

Keyword Arguments

headers (*dict*) – Header names and values to pass to the request.

Returns

The return data from the PUT request, as JSON.

Return type

object

select(cls, unique_id=None, *args, **kwargs)

Prepare a query against the Carbon Black data store.

Most objects returned by the SDK are returned via queries created using this method.

Parameters

- **cls** (*class* / *str*) – The Model class (for example, Computer, Process, Binary, FileInstance) to query
- **unique_id** (*Any*) – The unique id of the object to retrieve, to retrieve a single object by ID. Default is None (create a standard query).
- ***args** (*list*) – Additional arguments to pass to a created object.
- ****kwargs** (*dict*) – Additional arguments to pass to a created object or query.

Returns

An instance of the Model class if a unique_id is provided, otherwise a Query object.

Return type

object

property url

The connection URL.

validate_process_query(query)

Validates the given IOC query.

Parameters**query** (*str*) – The query to validate.**Returns**

True if the query is valid, False if not.

Return type

bool

Examples

```
>>> cb.validate_process_query("process_name:chrome.exe") # True
```

4.7 Audit and Remediation Package

4.7.1 Base Module

Model and Query Classes for Audit and Remediation

class `DeviceSummary`(*cb, initial_data*)Bases: `UnrefreshableModel`

Represents the summary of results from a single device during a single Audit and Remediation *Run*.

Parameters

- **id** – The result's unique ID
- **total_results** – Number of results returned for this particular device
- **device** – Information associated with the device
- **time_received** – The time at which this result was received
- **status** – The result's status
- **device_message** – Placeholder
- **metrics** – Metrics associated with the device

Initialize a DeviceSummary object with initial_data.

Parameters

- **cb** (`BaseAPI`) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the result.

class `Metrics`(*cb, initial_data*)Bases: `UnrefreshableModel`

Represents the metrics for a result.

Initialize a DeviceSummary Metrics object with initial_data.

Parameters

- **cb** (`BaseAPI`) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the result.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

property metrics_

Returns the reified *DeviceSummary.Metrics* for this result.

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class DeviceSummaryFacet(*cb*, *initial_data*)

Bases: [ResultFacet](#)

Represents the summary of results for a single device summary in an Audit and Remediation *Run*.

Initialize a DeviceSummaryFacet object with initial_data.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the result.

class Values(*cb, initial_data*)

Bases: [*UnrefreshableModel*](#)

Represents the values associated with a field.

Initialize a ResultFacet Values object with *initial_data*.

Parameters

- **cb** ([*BaseAPI*](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the result.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

property values_

Returns the reified *ResultFacet.Values* for this result.

class FacetQuery(*doc_class, cb*)

Bases: [BaseQuery](#), [QueryBuilderSupportMixin](#), [IterableQueryMixin](#),
[CriteriaBuilderSupportMixin](#), [AsyncQueryMixin](#)

Represents a query that receives facet information from a LiveQuery run.

Initialize the FacetQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None, **kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

[Query](#)

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

facet_field(*field*)

Sets the facet fields to be received by this query.

Parameters

field (*str* or [*str*]) – Field(s) to be received.

Returns

FacetQuery that will receive field(s) facet_field.

Return type

FacetQuery

Example

```
>>> cb.select(ResultFacet).run_id(my_run).facet_field(["device.policy_name",
↪ "device.os"])
```

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(*q=None, **kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

run_id(*run_id*)

Sets the run ID to query results for.

Parameters

run_id (*str*) – The run ID to retrieve results for.

Returns

FacetQuery object with specified run_id.

Return type

FacetQuery

Example

```
>>> cb.select(ResultFacet).run_id(my_run)
```

set_device_ids(*device_ids*)

Sets the device.id criteria filter.

Parameters

device_ids (*[int]*) – Device IDs to filter on.

Returns

The FacetQuery with specified device.id.

Return type

FacetQuery

set_device_names(*device_names*)

Sets the device.name criteria filter.

Parameters

device_names (*[str]*) – Device names to filter on.

Returns

The FacetQuery with specified device.name.

Return type

FacetQuery

set_device_os(device_os)

Sets the device.os criteria.

Parameters

device_os (*[str]*) – Device OS's to filter on.

Returns

The FacetQuery object with specified device_os.

Return type

FacetQuery

Note: Device OS's can be one or more of ["WINDOWS", "MAC", "LINUX"].

set_policy_ids(policy_ids)

Sets the device.policy_id criteria.

Parameters

policy_ids (*[int]*) – Device policy ID's to filter on.

Returns

The FacetQuery object with specified policy_ids.

Return type

FacetQuery

set_policy_names(policy_names)

Sets the device.policy_name criteria.

Parameters

policy_names (*[str]*) – Device policy names to filter on.

Returns

The FacetQuery object with specified policy_names.

Return type

FacetQuery

set_statuses(statuses)

Sets the status criteria.

Parameters

statuses (*[str]*) – Query statuses to filter on.

Returns

The FacetQuery object with specified statuses.

Return type

FacetQuery

update_criteria(key, newlist)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

MAX_RESULTS_LIMIT = 10000

Audit and Remediation Models

class Result(*cb, initial_data*)

Bases: *UnrefreshableModel*

Represents a single result from an Audit and Remediation *Run*.

Parameters

- **id** – The result's unique ID
- **device** – The device associated with the result
- **status** – The result's status
- **time_received** – The time at which this result was received
- **device_message** – Placeholder
- **fields** – The fields returned by the backing osquery query
- **metrics** – Metrics associated with the result's host

Initialize a Result object with *initial_data*.

Device, Fields, and Metrics objects are attached using *initial_data*.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the result.

class Device(*cb, initial_data*)

Bases: *UnrefreshableModel*

Represents device information for a result.

Initialize a Device Result object with *initial_data*.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the result.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class Fields(*cb*, *initial_data*)

Bases: [UnrefreshableModel](#)

Represents the fields of a result.

Initialize a Result Fields object with *initial_data*.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the result.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class Metrics(*cb, initial_data*)

Bases: [*UnrefreshableModel*](#)

Represents the metrics of a result.

Initialize a Result Metrics object with *initial_data*.

Parameters

- **cb** ([*BaseAPI*](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the result.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

property device_

Returns the reified *Result.Device* for this result.

property fields_

Returns the reified *Result.Fields* for this result.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

property metrics_

Returns the reified *Result.Metrics* for this result.

query_device_summaries()

Returns a ResultQuery for a DeviceSummary.

This represents the search for a summary of results from a single device of a *Run*. The query may be further augmented with additional criteria prior to enumerating its results.

Returns

The query object returned by this operation.

Return type

ResultQuery

query_device_summary_facets()

Returns a ResultQuery for a DeviceSummaryFacet.

This represents the search for a summary of a single device summary of a *Run*. The query may be further augmented with additional criteria prior to enumerating its results.

Returns

The query object returned by this operation.

Return type

ResultQuery

query_result_facets()

Returns a ResultQuery for a ResultFacet.

This represents the search for a summary of results from a single field of a *Run*. The query may be further augmented with additional criteria prior to enumerating its results.

Returns

The query object returned by this operation.

Return type

ResultQuery

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The raw json Result.

Return type

dict

class ResultFacet(*cb, initial_data*)

Bases: *UnrefreshableModel*

Represents the summary of results for a single field in an Audit and Remediation *Run*.

Parameters

field – The name of the field being summarized

Initialize a ResultFacet object with initial_data.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the result.

class Values(*cb, initial_data*)

Bases: *UnrefreshableModel*

Represents the values associated with a field.

Initialize a ResultFacet Values object with initial_data.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the result.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

property values_Returns the reified *ResultFacet.Values* for this result.**class ResultQuery**(*doc_class*, *cb*)

Bases: [BaseQuery](#), [QueryBuilderSupportMixin](#), [IterableQueryMixin](#),
[CriteriaBuilderSupportMixin](#), [AsyncQueryMixin](#)

Represents a query that retrieves results from a LiveQuery run.

Initialize the ResultQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None, **kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

async_export()

Create an asynchronous job that exports the results from the run.

This is recommended if you are expecting a very large result set. Once the Job is created, wait for it to be completed, then get the results from the Job using one of the `get_output` methods on the [cbc_sdk.platform.jobs\(\)](#) object. To wait asynchronously for the results, use the Job object's `await_completion()` method.

Required Permissions:

livequery.manage(READ), jobs.status(READ)

Returns

The Job object that represents the asynchronous job.

Return type

Job

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

export_csv_as_file(filename)

Export the results from the run as CSV, writing the CSV to the named file.

Required Permissions:

livequery.manage(READ)

Parameters

filename (*str*) – Name of the file to write the results to.

export_csv_as_lines()

Export the results from the run as CSV, returning the CSV data as iterated lines.

Required Permissions:

livequery.manage(READ)

Returns

An iterable that can be used to get each line of CSV text in turn as a string.

Return type

iterable

export_csv_as_stream(output, compressed=False)

Export the results from the run as CSV, writing the CSV to the given stream.

Required Permissions:

livequery.manage(READ)

Parameters

- **output** (*RawIOBase*) – Stream to write the CSV data from the request to.
- **compressed** (*bool*) – True to download as a compressed ZIP file, False to download as CSV.

export_csv_as_string()

Export the results from the run as CSV, returning the CSV data as a string.

Required Permissions:

livequery.manage(READ)

Returns

The CSV data as one big string.

Return type

str

export_zipped_csv(filename)

Export the results from the run as a zipped CSV, writing the zip data to the named file.

Required Permissions:

livequery.manage(READ)

Parameters**filename** (*str*) – Name of the file to write the results to.**first()**

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(q=None, **kwargs)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type*Query***one()**

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(q=None, **kwargs)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type*Query***run_id**(*run_id*)

Sets the run ID to query results for.

Parameters

run_id (*str*) – The run ID to retrieve results for.

Returns

ResultQuery object with specified run_id.

Return type*ResultQuery***Example**

```
>>> cb.select(Result).run_id(my_run)
```

scroll(*rows=10000*)

Iteratively fetch results across Live Query Runs or paginate all results beyond the 10k search limits.

To fetch the next set of results repeatedly call the scroll function until *ResultQuery.num_remaining == 0* or no results are returned.

Note: You must specify either a *set_time_received* or a *set_run_ids* on the query before using scroll

Parameters

rows (*int*) – The number of rows to fetch

Returns

The list of results

Return type*list[Result]***set_device_ids**(*device_ids*)

Sets the device.id criteria filter.

Parameters

device_ids (*[int]*) – Device IDs to filter on.

Returns

The ResultQuery with specified device.id.

Return type*ResultQuery***set_device_names**(*device_names*)

Sets the device.name criteria filter.

Parameters

device_names (*[str]*) – Device names to filter on.

Returns

The ResultQuery with specified device.name.

Return type*ResultQuery*

set_device_os(*device_os*)

Sets the device.os criteria.

Parameters

device_os (*[str]*) – Device OS’s to filter on.

Returns

The ResultQuery object with specified device_os.

Return type

ResultQuery

Note: Device OS’s can be one or more of [“WINDOWS”, “MAC”, “LINUX”].

set_policy_ids(*policy_ids*)

Sets the device.policy_id criteria.

Parameters

policy_ids (*[int]*) – Device policy ID’s to filter on.

Returns

The ResultQuery object with specified policy_ids.

Return type

ResultQuery

set_policy_names(*policy_names*)

Sets the device.policy_name criteria.

Parameters

policy_names (*[str]*) – Device policy names to filter on.

Returns

The ResultQuery object with specified policy_names.

Return type

ResultQuery

set_run_ids(*run_ids*)

Sets the run IDs to query results for.

Note: Only supported for scroll

Parameters

run_ids (*list[str]*) – The run IDs to retrieve results for.

Returns

ResultQuery object with specified run_id.

Return type

ResultQuery

set_statuses(*statuses*)

Sets the status criteria.

Parameters

statuses (*[str]*) – Query statuses to filter on.

Returns

The ResultQuery object with specified statuses.

Return type

ResultQuery

set_time_received(*start=None, end=None, range=None*)

Set the time received to query results for.

Note: If you are using scroll you may only specify range, or start and end. range supports max of 24hrs

Parameters

- **start** (*str*) – Start time in ISO8601 UTC format
- **end** (*str*) – End time in ISO8601 UTC format
- **range** (*str*) – Relative time window using the following allowed time units y years, w weeks, d days, h hours, m minutes, s seconds

Returns

ResultQuery object with specified time_received.

Return type

ResultQuery

sort_by(*key, direction='ASC'*)

Sets the sorting behavior on a query's results.

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

ResultQuery object with specified sorting key and order.

Return type

ResultQuery

Example

```
>>> cb.select(Result).run_id(my_run).where(username="foobar").sort_by("uid")
```

update_criteria(*key, newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

class Run(*cb, model_unique_id=None, initial_data=None*)

Bases: *NewBaseModel*

Represents an Audit and Remediation run.

Example:

```
>>> run = cb.select(Run, run_id)
>>> print(run.name, run.sql, run.create_time)
>>> print(run.status, run.match_count)
>>> run.refresh()
```

Parameters

- **org_key** – The organization key for this run
- **name** – The name of the Audit and Remediation run
- **id** – The run's unique ID
- **sql** – The Audit and Remediation query
- **created_by** – The user or API id that created the run
- **create_time** – When this run was created
- **status_update_time** – When the status of this run was last updated
- **timeout_time** – The time at which the query will stop requesting results from any devices who have not responded
- **cancellation_time** – The time at which a user or API id cancelled the run
- **cancelled_by** – The user or API id that cancelled the run
- **notify_on_finish** – Whether or not to send an email on query completion
- **active_org_devices** – The number of devices active in the organization
- **status** – The run status

- **device_filter** – Any device filter rules associated with the run
- **last_result_time** – When the most recent result for this run was reported
- **total_results** – The number of results received
- **match_count** – The number of devices which received a match to the query
- **no_match_count** – The number of devices which did not received a match to the query
- **error_count** – The number of devices which errored
- **not_supported_count** – The number of devices which do not support a portion of the osquery
- **cancelled_count** – The number of devices which were cancelled before they ran the query
- **not_started_count** – The number of devices which have not run the query
- **success_count** – The number of devices which succeeded in running the query
- **in_progress_count** – The number of devices which were currently executing the query
- **recommended_query_id** – The id of a query from the recommended route
- **template_id** – The template that created the run

Initialize a Run object with initial_data.

Required Permissions:

livequery.manage(READ)

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the query run represented.
- **initial_data** (*dict*) – Initial data used to populate the query run.

delete()

Delete a query.

Required Permissions:

livequery.manage(DELETE)

Returns

True if the query was deleted successfully, False otherwise.

Return type

bool

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

query_device_summaries()

Create a DeviceSummary query that searches for all device summaries on this run.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all device summaries for this run.

Return type*ResultQuery***Raises**

ApiError – If the query has been deleted.

query_facets()

Create a ResultFacet query that searches for all result facets on this run.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all result facets for this run.

Return type*FacetQuery***Raises**

ApiError – If the query has been deleted.

query_results()

Create a Result query that searches for all results on this run.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all results for this run.

Return type*ResultQuery***Raises**

ApiError – If the query has been deleted.

refresh()

Reload this object from the server.

stop()

Stop a running query.

Required Permissions:

livequery.manage(UPDATE)

Returns

True if query was stopped successfully, False otherwise.

Return type

bool

Raises

ServerError – If the server response cannot be parsed as JSON.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class RunHistory(*cb, initial_data=None*)

Bases: [Run](#)

Represents a historical Audit and Remediation *Run*.

Initialize a RunHistory object with initial_data.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the history object.

delete()

Delete a query.

Required Permissions:

livequery.manage(DELETE)

Returns

True if the query was deleted successfully, False otherwise.

Return type

bool

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

query_device_summaries()

Create a DeviceSummary query that searches for all device summaries on this run.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all device summaries for this run.

Return type

[ResultQuery](#)

Raises

[ApiError](#) – If the query has been deleted.

query_facets()

Create a ResultFacet query that searches for all result facets on this run.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all result facets for this run.

Return type

FacetQuery

Raises

ApiError – If the query has been deleted.

query_results()

Create a Result query that searches for all results on this run.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all results for this run.

Return type

ResultQuery

Raises

ApiError – If the query has been deleted.

refresh()

Reload this object from the server.

stop()

Stop a running query.

Required Permissions:

livequery.manage(UPDATE)

Returns

True if query was stopped successfully, False otherwise.

Return type

bool

Raises

ServerError – If the server response cannot be parsed as JSON.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class RunHistoryQuery(doc_class, cb)

Bases: *BaseQuery*, *QueryBuilderSupportMixin*, *IterableQueryMixin*,
CriteriaBuilderSupportMixin, *AsyncQueryMixin*

Represents a query that retrieves historic LiveQuery runs.

Initialize the RunHistoryQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None, **kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(*q=None, **kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

set_template_ids(*template_ids*)

Sets the template_id criteria filter.

Parameters

template_ids (*[str]*) – Template IDs to filter on.

Returns

The RunHistoryQuery with specified template_id.

Return type

RunHistoryQuery

sort_by(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

RunHistoryQuery object with specified sorting key and order.

Return type

RunHistoryQuery

Example:

```
>>> cb.select(Result).run_id(my_run).where(username="foobar").sort_by("uid")
```

update_criteria(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q*=None, ***kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

class RunQuery(*doc_class*, *cb*)

Bases: *BaseQuery*, *AsyncQueryMixin*

Represents a query that either creates or retrieves the status of a LiveQuery run.

Initialize the RunQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.

device_ids(*device_ids*)

Restricts the devices that this Audit and Remediation run is performed on to the given IDs.

Parameters

device_ids (*[int]*) – Device IDs to perform the Run on.

Returns

The RunQuery with specified device_ids.

Return type

RunQuery

device_types(*device_types*)

Restricts the devices that this Audit and Remediation run is performed on to the given OS.

Parameters

device_types (*[str]*) – Device types to perform the Run on.

Returns

The RunQuery object with specified device_types.

Return type

RunQuery

Note: Device type can be one of ["WINDOWS", "MAC", "LINUX"].

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

name(*name*)

Sets this Audit and Remediation run's name.

If no name is explicitly set, the run is named after its SQL.

Parameters

name (*str*) – The name for this Run.

Returns

The RunQuery object with specified name.

Return type

RunQuery

notify_on_finish()

Sets the notify-on-finish flag on this Audit and Remediation run.

Returns

The RunQuery object with *notify_on_finish* set to True.

Return type*RunQuery***policy_id(policy_id)**

Restricts this Audit and Remediation run to the given policy ID.

Parameters

policy_id (*int*) or (*list[int]*) – Policy ID to perform the Run on.

Returns

The RunQuery object with specified policy_id.

Return type*RunQuery***schedule(rrule, timezone)**

Sets a schedule for the SQL Query to recur

A schedule requires an rrule and a timezone to determine the time to rerun the SQL query. rrule is defined in RFC 2445 however only a subset of the functionality is supported here. If a Run is created with a schedule then the Run will contain a template_id to the corresponding template and a new Run will be created each time the schedule is met.

Example RRule, Daily

Field	Values
BYSECOND	0
BYMINUTE	0 or 30
BYHOUR	0 to 23

Daily at 1:30PM

RRULE:FREQ=DAILY;BYHOUR=13;BYMINUTE=30;BYSECOND=0

Example RRule, Weekly

Field	Values
BYSECOND	0
BYMINUTE	0
BYHOUR	0 to 23
BYDAY	One or more: SU, MO, TU, WE, TH, FR, SA

Monday and Friday of the week at 2:30 AM

RRULE:FREQ=WEEKLY;BYDAY=MO,FR;BYHOUR=13;BYMINUTE=30;BYSECOND=0

Example RRule, Monthly

Note: Either (BYDAY and BYSETPOS) or BYMONTHDAY is required.

Field	Values
BYSECOND	0
BYMINUTE	0 or 30
BYHOUR	0 to 23
BYDAY	One or more: SU, MO, TU, WE, TH, FR, SA
BYSETPOS	-1, 1, 2, 3, 4
BYMONTHDAY	One or more: 1 to 28

Last Monday of the Month at 2:30 AM

RRULE:FREQ=MONTHLY;BYDAY=MO;BYSETPOS=-1;BYHOUR=2;BYMINUTE=30;BYSECOND=0

1st and 15th of the Month at 2:30 AM

RRULE:FREQ=DAILY;BYMONTHDAY=1,15;BYHOUR=2;BYMINUTE=30;BYSECOND=0

Parameters

- **rrule** (*string*) – A recurrence rule (RFC 2445) specifying the frequency and time at which the query will recur
- **timezone** (*string*) – The timezone database name to use as a base for the rrule

Returns

The RunQuery with a recurrence schedule.

Return type

RunQuery

submit()

Submits this Audit and Remediation run.

Returns

A new *Run* instance containing the run's status.

Return type

Run

Raises

ApiError – If the Run does not have SQL set, or if the Run has already been submitted.

where(sql)

Sets this Audit and Remediation run's underlying SQL.

Parameters

sql (*str*) – The SQL to execute for the Run.

Returns

The RunQuery object with specified sql.

Return type

RunQuery

class Template(*cb, model_unique_id=None, initial_data=None*)

Bases: *Run*

Represents an Audit and Remediation Live Query Template.

Example:

```
>>> template = cb.select(Template, template_id)
>>> print(template.name, template.sql, template.create_time)
>>> print(template.status, template.match_count, template.schedule)
>>> template.refresh()
```

Parameters

- **org_key** – The organization key for this run
- **name** – The name of the Audit and Remediation run
- **id** – The run’s unique ID
- **sql** – The Audit and Remediation query
- **created_by** – The user or API id that created the run
- **create_time** – When this run was created
- **status_update_time** – When the status of this run was last updated
- **timeout_time** – The time at which the query will stop requesting results from any devices who have not responded
- **cancellation_time** – The time at which a user or API id cancelled the run
- **cancelled_by** – The user or API id that cancelled the run
- **archive_time** – The time at which a user or API id cancelled the run
- **archived_by** – The user or API id that archived the run
- **notify_on_finish** – Whether or not to send an email on query completion
- **active_org_devices** – The number of devices active in the organization
- **status** – The run status
- **device_filter** – Any device filter rules associated with the run
- **last_result_time** – When the most recent result for this run was reported
- **total_results** – The number of results received
- **match_count** – The number of devices which received a match to the query
- **no_match_count** – The number of devices which did not received a match to the query
- **error_count** – The number of devices which errored
- **not_supported_count** – The number of devices which do not support a portion of the osquery
- **cancelled_count** – The number of devices which were cancelled before they ran the query
- **not_started_count** – The number of devices which have not run the query
- **success_count** – The number of devices which succeeded in running the query
- **in_progress_count** – The number of devices which were currently executing the query
- **recommended_query_id** – The id of a query from the recommended route
- **template_id** – The template that created the run

Initialize a Template object with `initial_data`.

Required Permissions:

livequery.manage(READ)

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the query run represented.
- **initial_data** (*dict*) – Initial data used to populate the query run.

delete()

Delete a query.

Required Permissions:

livequery.manage(DELETE)

Returns

True if the query was deleted successfully, False otherwise.

Return type

bool

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

query_device_summaries()

Create a DeviceSummary query that searches for all device summaries on this run.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all device summaries for this run.

Return type

ResultQuery

Raises

ApiError – If the query has been deleted.

query_facets()

Create a ResultFacet query that searches for all result facets on this run.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all result facets for this run.

Return type

FacetQuery

Raises

ApiError – If the query has been deleted.

query_results()

Create a Result query that searches for all results on this run.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all results for this run.

Return type

ResultQuery

Raises

ApiError – If the query has been deleted.

query_runs()

Create a RunHistory query that searches for all runs created by this template ID.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all runs based on this template.

Return type

RunHistoryQuery

refresh()

Reload this object from the server.

stop()

Stop a template.

Required Permissions:

livequery.manage(UPDATE)

Returns

True if query was stopped successfully, False otherwise.

Return type

bool

Raises

ServerError – If the server response cannot be parsed as JSON.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class TemplateHistory(cb, initial_data=None)

Bases: *Template*

Represents a historical Audit and Remediation *Template*.

Initialize a Template object with initial_data.

Required Permissions:

livequery.manage(READ)

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the query run.

delete()

Delete a query.

Required Permissions:

livequery.manage(DELETE)

Returns

True if the query was deleted successfully, False otherwise.

Return type

bool

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

query_device_summaries()

Create a DeviceSummary query that searches for all device summaries on this run.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all device summaries for this run.

Return type

[ResultQuery](#)

Raises

[ApiError](#) – If the query has been deleted.

query_facets()

Create a ResultFacet query that searches for all result facets on this run.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all result facets for this run.

Return type

[FacetQuery](#)

Raises

[ApiError](#) – If the query has been deleted.

query_results()

Create a Result query that searches for all results on this run.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all results for this run.

Return type

ResultQuery

Raises

ApiError – If the query has been deleted.

query_runs()

Create a RunHistory query that searches for all runs created by this template ID.

The query may be further augmented with additional criteria prior to enumerating its results.

Returns

A query object which will search for all runs based on this template.

Return type

RunHistoryQuery

refresh()

Reload this object from the server.

stop()

Stop a template.

Required Permissions:

livequery.manage(UPDATE)

Returns

True if query was stopped successfully, False otherwise.

Return type

bool

Raises

ServerError – If the server response cannot be parsed as JSON.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class TemplateHistoryQuery(doc_class, cb)

Bases: *BaseQuery*, *QueryBuilderSupportMixin*, *IterableQueryMixin*, *CriteriaBuilderSupportMixin*, *AsyncQueryMixin*

Represents a query that retrieves historic LiveQuery templates.

Initialize the TemplateHistoryQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.

add_criteria(*key*, *newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None*, ***kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(*q=None, **kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- ***MoreThanOneResultError*** – If the query returns more than one item
- ***ObjectNotFoundError*** – If the query returns zero items

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

sort_by(*key, direction='ASC'*)

Sets the sorting behavior on a query's results.

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

object with specified sorting key and order.

Return type

TemplateHistoryQuery

Example:

```
>>> cb.select(Result).run_id(my_run).where(username="foobar").sort_by("uid")
```

update_criteria(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None*, ***kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, *QueryBuilder*, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

4.7.2 Differential Module

Model and Query Classes for Differential Analysis

ASYNC_RATE_LIMIT = 100

Differential Analysis Models

class Differential(*cb*, *initial_data=None*)

Bases: *NewBaseModel*

Represents a Differential Analysis run.

Example:

```
>>> query = cb.select(Differential).newer_run_id(newer_run_id)
>>> run = query.submit()
>>> print(run)
>>> print(run.diff_results)
```

Parameters

- **newer_run_id** – id against which the older run id results will be compared

- **newer_run_create_time** – Timestamp of the primary run in ISO 8601 UTC format
- **older_run_id** – This can be optional. If not specified, the previous run as compared to the primary will be chosen. This can be optional if you are comparing recurring runs only.
- **older_run_create_time** – Timestamp of the older run in ISO 8601 UTC format
- **diff_processed_time** – The time it took to process the results in seconds and milliseconds
- **newer_run_not_responded_devices** – Array of device IDs that have not responded
- **older_run_not_responded_devices** – Array of device IDs that have not responded
- **diff_results** – An object containing either count of changes only or count and actual diff results

Initialize a Differential object with `initial_data`.

Required Permissions for CBC:

`livequery.manage(READ)`

Required Permissions for CSP:

`_API.Live.Query:livequery.Manage.read`

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the query run.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class DifferentialQuery(*doc_class*, *cb*)

Bases: [BaseQuery](#), [IterableQueryMixin](#), [CriteriaBuilderSupportMixin](#)

Query used to compare two Live Query runs.

Initialize the DifferentialQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.

add_criteria(*key*, *newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

async_export()

Create an asynchronous job that exports the results from the run.

This is recommended if you are expecting a very large result set. Once the Job is created, wait for it to be completed, then get the results from the Job using one of the `get_output` methods on the `cbc_sdk.platform.jobs` object. To wait for the results, use the Job object's `await_completion()` method.

Example

```
>>> # Get the differential
>>> query = cb.select(Differential).newer_run_id(newer_run_id)
>>> export = query.async_export()
>>> # wait for the export to finish
>>> export.await_completion()
>>> # write the results to a file
>>> export.get_output_as_file("example_data.json")
```

Required CBC Permissions:

`livequery.manage(READ)`, `jobs.status(READ)`

Required CSP Permissions:

`_API.Live.Query:livequery.Manage.read`, `_API.Background_Tasks.jobs.status.read`

Returns

The Job object that represents the asynchronous job.

Return type

Job

count_only(count_only)

Return only count of diff results per device or complete diff metadata result.

The default value is true, which means only the count will be returned.

Example

```
>>> query = cb.select(Differential).newer_run_id(newer_run_id).count_only(True)
>>> run = query.submit()
```

Parameters

count_only (*string*) – Boolean that indicates whether to return actual metadata or return just the count of differences

Returns

This instance.

Return type

DifferentialQuery

Raises

ApiError – If invalid values are passed in the list.

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

newer_run_id(newer_run_id)

Set the id against which the older_run_id results will be compared.

Example

```
>>> query = cb.select(Differential).newer_run_id(newer_run_id)
>>> run = query.submit()
```

Parameters

newer_run_id (*string*) – id against which the older_run_id results will be compared.

Returns

This instance.

Return type

DifferentialQuery

Raises

ApiError – If invalid values are passed.

older_run_id(*older_run_id*)

This can be optional.

If not specified, the previous run as compared to the primary will be chosen if it is a recurring one. If comparing two individual runs, this is required.

Example

```
>>> query = cb.select(Differential).newer_run_id(newer_run_id).older_run_
↳ id(older_run_id)
>>> run = query.submit()
```

Parameters

older_run_id (*string*) – id against which the newer_run_id results will be compared.

Returns

This instance.

Return type

DifferentialQuery

Raises

ApiError – If invalid values are passed.

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- ***MoreThanOneResultError*** – If the query returns more than one item
- ***ObjectNotFoundError*** – If the query returns zero items

set_device_ids(*device_ids*)

Restricts the query on to the specified devices only.

Example

```
>>> query = cb.select(Differential).newer_run_id(newer_run_id).set_device_
↳ ids([12345, 56789])
>>> run = query.submit()
```

Parameters

device_ids (*list*) – List of device id(s)

Returns

This instance.

Return type*DifferentialQuery***Raises***ApiError* – If invalid values are passed in the list.**submit()**

Submits this Differential Analysis run.

Returns

A new *Differential* instance containing the run's content.

Return type*Run***update_criteria(key, newlist)**

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

4.8 Credential Providers Package

4.8.1 Default Module

Function which gives us the default credentials handler for use by CBCloudAPI.

class DefaultProvider

Bases: object

Intermediate class defined to allow insertion of a “test point” into default_credential_provider().

get_default_provider(credential_file)

Return the default credential provider that CBCloudAPI should use.

Parameters

credential_file (*str*) – Credential file as specified to the initialization of the API.

Returns

The default credential provider that CBCloudAPI should use.

Return type*CredentialProvider***default_credential_provider**(*credential_file*)

Return the default credential provider that CBCloudAPI should use.

Parameters

credential_file (*str*) – Credential file as specified to the initialization of the API.

Returns

The default credential provider that CBCloudAPI should use.

Return type*CredentialProvider*

4.8.2 AWS SM Credential Provider Module

Credentials provider that reads the credentials from the AWS Secrets Manager

class **AWSCredentialProvider**(*secret_arn*, *region_name*='us-east-2', *profile_name*=None)

Bases: *CredentialProvider*

This credential provider reads from the AWS Secrets Manager

Initialize the AWSCredentialProvider.

Parameters

- **secret_arn** (*str*) – The name of the secret in the AWS Secrets Manager.
- **region_name** (*str*) – The region name
- **profile_name** (*str*) – The credentials profile

get_credentials(*section*=None)

Return a Credentials object containing the configured credentials.

Parameters

- **section** (*None*) – Since AWS doesn't support sections it is left
- **CredentialProvider** (*to satisfy the Signature of*) –

Returns

The credentials retrieved from that source.

Return type*Credentials*

4.8.3 Environ Credential Provider Module

Credentials provider that reads the credentials from the environment.

class **EnvironCredentialProvider**

Bases: *CredentialProvider*

The object which provides credentials based on variables in the environment.

Initializes the EnvironCredentialProvider.

get_credentials(*section=None*)

Return a Credentials object containing the configured credentials.

Parameters

section (*str*) – The credential section to retrieve (not used in this provider).

Returns

The credentials retrieved from that source.

Return type

Credentials

Raises

CredentialError – If there is any error retrieving the credentials.

4.8.4 File Credential Provider Module

Credentials provider that reads the credentials from a file.

class FileCredentialProvider(*credential_file=None*)

Bases: *CredentialProvider*

The object which provides credentials based on a credential file.

Initialize the FileCredentialProvider.

Parameters

credential_file (*object*) – A string or path-like object representing the credentials file, or a list of strings or path-like objects representing the search path for the credentials file.

get_credentials(*section=None*)

Return a Credentials object containing the configured credentials.

Parameters

section (*str*) – The credential section to retrieve.

Returns

The credentials retrieved from that source.

Return type

Credentials

Raises

CredentialError – If there is any error retrieving the credentials.

4.8.5 Keychain Credential Provider Module

Credentials provider that reads the credentials from the macOS's keychain.

class KeychainCredentialProvider(*keychain_name, keychain_username*)

Bases: *CredentialProvider*

This credential provider reads from the macOS's Keychain.

Initialize the KeychainCredentialProvider.

Parameters

- **keychain_name** (*str*) – The name of the entry in the Keychain.
- **keychain_username** (*str*) – The username which you've set in the Keychain.

Raises

CredentialError – If we attempt to instantiate this provider on a non-macOS system.

get_credentials(*section=None*)

Return a Credentials object containing the configured credentials.

Parameters

- **section** (*None*) – Since Keychain doesn't support sections it is left
- **CredentialProvider** (*to satisfy the Signature of*) –

Returns

The credentials retrieved from that source.

Return type

Credentials

Raises

CredentialError – If there is any error retrieving the credentials.

4.8.6 Registry Credential Provider Module

Credentials provider that reads the credentials from the environment.

OpenKey(*base, path*)

Stub to maintain source compatibility

QueryValueEx(*key, name*)

Stub to maintain source compatibility

class RegistryCredentialProvider(*keypath=None, userkey=True*)

Bases: *CredentialProvider*

The credentials provider that reads from the Windows Registry.

Initialize the RegistryCredentialProvider.

Parameters

- **keypath** (*str*) – Path from the selected base key to the key that will contain individual sections.
- **userkey** (*bool*) – True if the keypath starts at HKEY_CURRENT_USER, False if at HKEY_LOCAL_MACHINE.

Raises

CredentialError – If we attempt to instantiate this provider on a non-Windows system.

get_credentials(*section=None*)

Return a Credentials object containing the configured credentials.

Parameters

section (*str*) – The credential section to retrieve.

Returns

The credentials retrieved from that source.

Return type

Credentials

Raises

CredentialError – If there is any error retrieving the credentials.

4.9 Endpoint Standard Package

4.9.1 Base Module

Model and Query Classes for Endpoint Standard

class EnrichedEvent(*cb, model_unique_id=None, initial_data=None, force_init=False, full_doc=True*)

Bases: [*UnrefreshableModel*](#)

Represents an enriched event retrieved by one of the Enterprise EDR endpoints.

Initialize the EnrichedEvent object.

Parameters

- **cb** ([*CBCloudAPI*](#)) – A reference to the CBCloudAPI object.
- **model_unique_id** (*Any*) – The unique ID for this particular instance of the model object.
- **initial_data** (*dict*) – The data to use when initializing the model object.
- **force_init** (*bool*) – True to force object initialization.
- **full_doc** (*bool*) – True to mark the object as fully initialized.

approve_process_sha256(*description=""*)

Approves the application by adding the process_sha256 to the WHITE_LIST

Parameters

description – The justification for why the application was added to the WHITE_LIST

Returns

ReputationOverride object

created in the Carbon Black Cloud

Return type

[*ReputationOverride*](#) (*cbc_sdk.platform.ReputationOverride*)

ban_process_sha256(*description=""*)

Bans the application by adding the process_sha256 to the BLACK_LIST

Parameters

description – The justification for why the application was added to the BLACK_LIST

Returns

ReputationOverride object

created in the Carbon Black Cloud

Return type

[*ReputationOverride*](#) (*cbc_sdk.platform.ReputationOverride*)

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_details(*timeout=0, async_mode=False*)

Requests detailed results.

Parameters

- **timeout** (*int*) – Event details request timeout in milliseconds. This value can never be greater than the configured default timeout. If this value is 0, the configured default timeout is used.
- **async_mode** (*bool*) – True to request details in an asynchronous manner.

Note:

- When using asynchronous mode, this method returns a python future. You can call `result()` on the future object to wait for completion and get the results.
-

property process_sha256

Returns a string representation of the SHA256 hash for this process.

Returns

SHA256 hash of the process.

Return type

hash (str)

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class EnrichedEventFacet(*cb, model_unique_id, initial_data*)Bases: [*UnrefreshableModel*](#)

Represents an enriched event retrieved by one of the Enterprise EDR endpoints.

Parameters

- **job_id** – The Job ID assigned to this query
- **terms** – Contains the Enriched Event Facet search results
- **ranges** – Groupings for search result properties that are ISO 8601 timestamps or numbers
- **contacted** – The number of searchers contacted for this query
- **completed** – The number of searchers that have reported their results

Initialize the Terms object with initial data.

class `Ranges(cb, initial_data)`

Bases: `UnrefreshableModel`

Represents the range (bucketed) facet fields and values associated with an Enriched Event Facet query.

Initialize an EnrichedEventFacet Ranges object with `initial_data`.

property facets

Returns the reified `EnrichedEventFacet.Terms._facets` for this result.

property fields

Returns the ranges fields for this result.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class `Terms(cb, initial_data)`

Bases: `UnrefreshableModel`

Represents the facet fields and values associated with an Enriched Event Facet query.

Initialize an EnrichedEventFacet Terms object with `initial_data`.

property facets

Returns the terms' facets for this result.

property fields

Returns the terms facets' fields for this result.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

property ranges_

Returns the reified *EnrichedEventFacet.Ranges* for this result.

refresh()

Reload this object from the server.

property terms_

Returns the reified *EnrichedEventFacet.Terms* for this result.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class EnrichedEventQuery(doc_class, cb)

Bases: [Query](#)

Represents the query logic for an Enriched Event query.

This class specializes *Query* to handle the particulars of enriched events querying.

Initialize the EnrichedEventQuery object.

Parameters

- **doc_class** (*class*) – The class of the model this query returns.
- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.

add_criteria(key, newlist)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.

- **newlist** (*str* or *list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

add_exclusions(*key*, *newlist*)

Add to the exclusions on this query with a custom exclusions key.

Will overwrite any existing exclusion for the specified key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*str* or *list[str]*) – Value or list of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).add_exclusions("type", ["WATCHLIST"])
>>> query = api.select(Alert).add_exclusions("type", "WATCHLIST")
```

aggregation(*field*)

Performs an aggregation search where results are grouped by an aggregation field

Parameters

field (*str*) – The aggregation field, either 'process_sha256' or 'device_id'

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None*, ***kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

batch_size(*new_batch_size*)

Set the batch size of the paginated query.

Parameters

new_batch_size (*int*) – The new batch size.

Returns

A new query with the updated batch size.

Return type

PaginatedQuery

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(*q=None, **kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(***kwargs*)

or_() criteria are explicitly provided to EnrichedEvent queries.

This method overrides the base class in order to provide *or_()* functionality rather than raising an exception.

set_fields(*fields*)

Sets the fields to be returned with the response.

Parameters

fields (*str* or *list[str]*) – Field or list of fields to be returned.

set_rows(*rows*)

Sets the ‘rows’ query body parameter to the ‘start search’ API call, determining how many rows to request.

Parameters

rows (*int*) – How many rows to request.

set_start(*start*)

Sets the ‘start’ query body parameter, determining where to begin retrieving results from.

Parameters

start (*int*) – Where to start results from.

set_time_range(*start=None, end=None, window=None*)

Sets the ‘time_range’ query body parameter, determining a time window based on ‘device_timestamp’.

Parameters

- **start** (*str* in ISO 8601 timestamp) – When to start the result search.
- **end** (*str* in ISO 8601 timestamp) – When to end the result search.
- **window** (*str*) – Time window to execute the result search, ending on the current time. Should be in the form “-2w”, where y=year, w=week, d=day, h=hour, m=minute, s=second.

Note:

- *window* will take precedent over *start* and *end* if provided.
-

Examples

```
>>> query = api.select(Process).set_time_range(start="2020-10-20T20:34:07Z").
↳ where("query is required")
>>> second_query = api.select(Process).
...     set_time_range(start="2020-10-20T20:34:07Z", end="2020-10-30T20:34:07Z
↳ ").where("query is required")
>>> third_query = api.select(Process).set_time_range(window='-3d').where("query_
↳ is required")
```

sort_by(*key, direction='ASC'*)

Sets the sorting behavior on a query’s results.

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

The query with sorting parameters.

Return type

Query

Example

```
>>> cb.select(Process).where(process_name="cmd.exe").sort_by("device_timestamp")
```

timeout(*msecs*)

Sets the timeout on a event query.

Parameters

msecs (*int*) – Timeout duration, in milliseconds. This value can cever be greater than the configured default timeout. If this value is 0, the configured default timeout is used.

Returns

The Query object with new milliseconds parameter.

Return type

Query (EnrichedEventQuery)

Example

```
>>> cb.select(EnrichedEvent).where(process_name="foo.exe").timeout(5000)
```

update_criteria(*key, newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

update_exclusions(*key, newlist*)

Update the exclusion on this query with a custom exclusion key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*list*) – List of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).update_exclusions("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

class Event(*cb, model_unique_id, initial_data=None*)

Bases: object

Represents an Endpoint Standard Event.

This functionality has been decommissioned. Please use `EnrichedEvent` instead. More information may be found here: <https://community.carbonblack.com/t5/Developer-Relations/Migration-Guide-Carbon-Black-Cloud-Events-API/m-p/95915/thread-id/2519>

This functionality has been decommissioned. Do not use.

Parameters

- **cb** (*BaseAPI*) – Unused.
- **model_unique_id** (*int*) – Unused.
- **initial_data** (*dict*) – Unused.

Raises

FunctionalityDecommissioned – Always.

log = <Logger cbc_sdk.endpoint_standard.base (WARNING)>

Endpoint Standard Models

4.9.2 Standard Recommendation Module

Model and query APIs for Recommendations

class Recommendation(*cb, model_unique_id, initial_data=None*)

Bases: *NewBaseModel*

Represents a recommended proposed policy change for the organization.

Parameters

- **changed_by** – Who made the last update to the workflow

- **create_time** – The time the recommendation was created
- **ref_id** – Reference id for an accepted Recommendation which is the id of the created Reputation Override
- **status** – Status of the recommendation
- **update_time** – The last time the recommendation was updated
- **comment** – A comment added when the recommendation was updated

Initialize the Recommendation object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the recommendation represented.
- **initial_data** (*dict*) – Initial data used to populate the recommendation.

class RecommendationApplication(*cb, model_unique_id, initial_data=None*)

Bases: [UnrefreshableModel](#)

Represents the rule application of a proposed change to an organization's policies.

Parameters

- **type** – Application type
- **value** – Application value

Initialize the RecommendationApplication object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – Should be None.
- **initial_data** (*dict*) – Initial data used to populate the object.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

```
class RecommendationImpact(cb, model_unique_id, initial_data=None)
```

Bases: [UnrefreshableModel](#)

Represents metadata about a recommendation to be used in the decision to accept or reject it.

Parameters

- **event_count** – Number of alerts encountered for recommendation
- **impact_score** – Impact score
- **impacted_devices** – Number of devices impacted by the recommendation
- **org_adoption** – Priority for adoption of this recommendation
- **update_time** – The last time this impact was updated

Initialize the RecommendationImpact object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – Should be None.
- **initial_data** (*dict*) – Initial data used to populate the object.

```
get(attrname, default_val=None)
```

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

```
refresh()
```

Reload this object from the server.

```
to_json()
```

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

```
class RecommendationNewRule(cb, model_unique_id, initial_data=None)
```

Bases: [UnrefreshableModel](#)

Represents the proposed change to an organization's policies from a recommendation.

Parameters

- **action** – Rule action
- **application** – Rule application
- **certificate_authority** – Certificate authority
- **filename** – File name
- **include_child_processes** – Include child processes
- **operation** – Operation

- **override_list** – Override list
- **override_type** – Override type
- **path** – File path
- **sha256_hash** – SHA256 hash
- **signed_by** – Signed by

Initialize the RecommendationNewRule object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – Should be None.
- **initial_data** (*dict*) – Initial data used to populate the object.

property application_

Return the object representing the rule application of a proposed change to an organization's policies.

Returns

The object representing the rule application of a proposed change.

Return type

[RecommendationApplication](#)

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class RecommendationWorkflow(cb, model_unique_id, initial_data=None)

Bases: [UnrefreshableModel](#)

Represents the lifecycle state of a recommendation.

Parameters

- **changed_by** – Who made the last update to the workflow
- **create_time** – The time the recommendation was created
- **ref_id** – Reference id for an accepted Recommendation which is the id of the created Reputation Override
- **status** – Status of the recommendation

- **update_time** – The last time the recommendation was updated
- **comment** – A comment added when the recommendation was updated

Initialize the RecommendationWorkflow object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – Should be None.
- **initial_data** (*dict*) – Initial data used to populate the object.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

accept(*comment=None*)

Accept this recommendation, converting it into a reputation override.

Parameters

- **comment** (*str*) – Optional comment associated with the action.

Returns

True if we successfully refreshed this Recommendation's state, False if not.

Return type

bool

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

property impact_

Return the object representing metadata about the recommendation.

Returns

The object representing metadata about the recommendation.

Return type

RecommendationImpact

property new_rule_

Return the object representing the proposed change to an organization's policies from the recommendation.

Returns

The object representing the proposed change to an organization's policies.

Return type

RecommendationNewRule

refresh()

Reload this object from the server.

reject(comment=None)

Reject this recommendation.

Parameters

comment (*str*) – Optional comment associated with the action.

Returns

True if we successfully refreshed this Recommendation's state, False if not.

Return type

bool

reputation_override()

Returns the reputation override associated with the recommendation (if the recommendation was accepted).

Returns

The associated reputation override, or None if there is none.

Return type

ReputationOverride

reset(comment=None)

Reset the recommendation, undoing any created reputation override and setting it back to NEW state.

Parameters

comment (*str*) – Optional comment associated with the action.

Returns

True if we successfully refreshed this Recommendation's state, False if not.

Return type

bool

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

property workflow_

Returns the object representing the lifecycle state of the recommendation.

Returns

The object representing the lifecycle state of the recommendation.

Return type

RecommendationWorkflow

class RecommendationQuery(doc_class, cb)

Bases: *BaseQuery*, *CriteriaBuilderSupportMixin*, *IterableQueryMixin*, *AsyncQueryMixin*

Query used to locate Recommendation objects.

Initialize the RecommendationQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.

add_criteria(key, newlist)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

set_hashes(*hashes*)

Restricts the recommendations that this query is performed on to the specified hashes.

Parameters

hashes (*list*) – List of hashes to restrict the search to.

Returns

This instance.

Return type

RecommendationQuery

Raises

ApiError – If invalid values are passed in the list.

set_policy_types(*policy_types*)

Restricts the recommendations that this query is performed on to the specified policy types.

Parameters

policy_types (*list*) – List of policy types to restrict the search to.

Returns

This instance.

Return type

RecommendationQuery

Raises

ApiError – If invalid values are passed in the list.

set_statuses(*statuses*)

Restricts the recommendations that this query is performed on to the specified status values.

Parameters

statuses (*list*) – List of status values to restrict the search to. If no statuses are specified, the search defaults to NEW only.

Returns

This instance.

Return type*RecommendationQuery***Raises***ApiError* – If invalid values are passed in the list.**sort_by**(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Example

```
>>> cb.select(USBDevice).sort_by("product_name")
```

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

This instance.

Return type*USBDeviceQuery***update_criteria**(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

log = <Logger cbc_sdk.endpoint_standard.recommendation (WARNING)>

Recommendation models

4.9.3 USB Device Control Module

Model and Query Classes for USB Device Control

class `USBDevice`(*cb*, *model_unique_id*, *initial_data=None*)

Bases: [`NewBaseModel`](#)

Represents a USB device.

Parameters

- **created_at** – the UTC date the external USB device configuration was created in ISO 8601 format
- **device_friendly_name** – human readable name for the external USB device
- **device_name** – name of the external USB device
- **device_type** – type of external USB device
- **endpoint_count** – number of endpoints that the external USB device has connected to
- **first_seen** – first timestamp that the external USB device was seen
- **id** – the id for this external USB device
- **interface_type** – type of interface used by external USB device
- **last_endpoint_id** – ID of the last endpoint the device accessed
- **last_endpoint_name** – name of the last endpoint the device accessed
- **last_policy_id** – ID of the last policy associated with the device
- **last_seen** – last timestamp that the external USB device was seen
- **org_key** – unique org key of the organization that the external USB device was connected to
- **product_id** – product ID of the external USB device in decimal form
- **product_name** – product name of the external USB device
- **serial_number** – serial number of external device
- **status** – Calculated status of device
- **updated_at** – the UTC date the external USB device configuration was updated in ISO 8601 format
- **vendor_id** – ID of the Vendor for the external USB device in decimal form
- **vendor_name** – vendor name of the external USB device

Initialize the USBDevice object.

Parameters

- **cb** ([`BaseAPI`](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

approve(*approval_name*, *notes*)

Creates and saves an approval for this USB device, allowing it to be treated as approved from now on.

Required Permissions:

external-device.manage (CREATE)

Parameters

- **approval_name** (*str*) – The name for this new approval.
- **notes** (*str*) – Notes to be added to this approval.

Returns

The new approval.

Return type

USBDeviceApproval

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_endpoints()

Returns the information about endpoints associated with this USB device.

Required Permissions:

external-device.manage (READ)

Returns

List of information about USB endpoints, each item specified as a dict.

Return type

list

classmethod get_vendors_and_products_seen(*cb*)

Returns all vendors and products that have been seen for the organization.

Required Permissions:

external-device.manage (READ)

Parameters

cb (*BaseAPI*) – Reference to API object used to communicate with the server.

Returns

A list of vendors and products seen for the organization, each vendor being represented by a dict.

Return type

list

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class USBDeviceApproval(*cb, model_unique_id, initial_data=None*)

Bases: [*MutableBaseModel*](#)

Represents a USB device approval.

Parameters

- **approval_name** – the name of the approval
- **created_at** – the UTC date the approval was created in ISO 8601 format
- **id** – the id for this approval
- **notes** – the notes for the approval
- **product_id** – product ID of the approval's external USB device in hex form
- **product_name** – product name of the approval's external USB device
- **serial_number** – serial number of the approval's external device
- **updated_at** – the UTC date the approval was updated in ISO 8601 format
- **updated_by** – the user who updated the record last
- **vendor_id** – ID of the Vendor for the approval's external USB device in hex form
- **vendor_name** – vendor name of the approval's external USB device

Initialize the USBDeviceApproval object.

Parameters

- **cb** ([*BaseAPI*](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

classmethod bulk_create(*cb, approvals*)

Creates multiple approvals and returns the USBDeviceApproval objects. Data is supplied as a list of dicts.

Required Permissions:

external-device.manage (CREATE)

Parameters

- **cb** ([*BaseAPI*](#)) – Reference to API object used to communicate with the server.
- **approvals** (*list*) – List of dicts containing approval data to be created, formatted as shown below.

Example

```
>>> [
    {
        "approval_name": "string",
        "notes": "string",
        "product_id": "string",
        "serial_number": "string",
        "vendor_id": "string"
    }
]
```

Returns

A list of USBDeviceApproval objects representing the approvals that were created.

Return type

list

classmethod `bulk_create_csv(cb, approval_data)`

Creates multiple approvals and returns the USBDeviceApproval objects. Data is supplied as text in CSV format.

Required Permissions:

external-device.manage (CREATE)

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **approval_data** (*str*) – CSV data for the approvals to be created. Header line MUST be included as shown below.

Example

vendor_id,product_id,serial_number,approval_name,notes

string,string,string,string,string

Returns

A list of USBDeviceApproval objects representing the approvals that were created.

Return type

list

classmethod `create_from_usb_device(usb_device)`

Creates a new, unsaved approval object from a USBDeviceObject, filling in its basic fields.

Parameters

usb_device ([USBDevice](#)) – The USB device to create the approval from.

Returns

The new approval object.

Return type

USBDeviceApproval

delete()

Delete this object.

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

MutableBaseModel

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(fulltouch=False)

Force this object to be considered as changed.

validate()

Validates this object.

Returns

True if the object is validated.

Return type

bool

Raises

InvalidObjectError – If the object has missing fields.

class USBDeviceApprovalQuery(*doc_class, cb*)

Bases: *BaseQuery*, *QueryBuilderSupportMixin*, *CriteriaBuilderSupportMixin*, *IterableQueryMixin*, *AsyncQueryMixin*

Represents a query that is used to locate USBDeviceApproval objects.

Initialize the USBDeviceApprovalQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None, **kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

export(*export_format*)

Starts the process of exporting USB device approval data from the organization in a specified format.

Required Permissions:

external-device.manage (READ)

Parameters

export_format (*str*) – The format to export USB device approval data in. Must be either “CSV” or “JSON”.

Returns

The asynchronous job that will provide the export output when the server has prepared it.

Return type

Job

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(*q=None, **kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

set_device_ids(*device_ids*)

Restricts the device approvals that this query is performed on to the specified device IDs.

Parameters

device_ids (*list*) – List of string device IDs.

Returns

This instance.

Return type

USBDeviceApprovalQuery

set_product_names(*product_names*)

Restricts the device approvals that this query is performed on to the specified product names.

Parameters

product_names (*list*) – List of string product names.

Returns

This instance.

Return type

USBDeviceApprovalQuery

set_vendor_names(*vendor_names*)

Restricts the device approvals that this query is performed on to the specified vendor names.

Parameters

vendor_names (*list*) – List of string vendor names.

Returns

This instance.

Return type

USBDeviceApprovalQuery

update_criteria(*key, newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

class USBDeviceBlock(*cb, model_unique_id, initial_data=None*)

Bases: *NewBaseModel*

Represents a USB device block.

Parameters

- **created_at** – the UTC date the block was created in ISO 8601 format
- **id** – the id for this block
- **policy_id** – policy id which is blocked
- **updated_at** – the UTC date the block was updated in ISO 8601 format

Initialize the USBDeviceBlock object.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

classmethod bulk_create(*cb, policy_ids*)

Creates multiple blocks and returns the USBDeviceBlocks that were created.

Required Permissions:

org.policies (UPDATE), external-device.enforce (UPDATE)

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **policy_ids** (*list*) – List of policy IDs to have blocks created for.

Returns

A list of USBDeviceBlock objects representing the approvals that were created.

Return type

list

classmethod `create(cb, policy_id)`

Creates a USBDeviceBlock for a given policy ID.

Required Permissions:

org.policies (UPDATE), external-device.enforce (UPDATE)

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **policy_id** (*str/int*) – Policy ID to create a USBDeviceBlock for.

Returns

New USBDeviceBlock object representing the block.

Return type*USBDeviceBlock***delete()**

Delete this object.

Required Permissions:

org.policies (DELETE), external-device.enforce (UPDATE)

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class `USBDeviceBlockQuery(doc_class, cb)`Bases: [BaseQuery](#), [IterableQueryMixin](#), [AsyncQueryMixin](#)

Represents a query that is used to locate USBDeviceBlock objects.

Initialize the USBDeviceBlockQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- [MoreThanOneResultError](#) – If the query returns more than one item
- [ObjectNotFoundError](#) – If the query returns zero items

class USBDeviceQuery(*doc_class, cb*)

Bases: [BaseQuery](#), [QueryBuilderSupportMixin](#), [CriteriaBuilderSupportMixin](#), [IterableQueryMixin](#), [AsyncQueryMixin](#)

Represents a query that is used to locate USBDevice objects.

Initialize the USBDeviceQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None, **kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

export(*export_format*)

Starts the process of exporting USB device data from the organization in a specified format.

Required Permissions:

external-device.manage (READ)

Parameters

export_format (*str*) – The format to export USB device data in. Must be either “CSV” or “JSON”.

Returns

The asynchronous job that will provide the export output when the server has prepared it.

Return type*Job***facets**(*fieldlist*, *max_rows=0*)

Return information about the facets for all known USB devices, using the defined criteria.

Required Permissions:

external-device.manage (READ)

Parameters

- **fieldlist** (*list*) – List of facet field names. Valid names are “vendor_name”, “product_name”, “endpoint.endpoint_name”, and “status”.
- **max_rows** (*int*) – The maximum number of rows to return. 0 means return all rows.

Returns

A list of facet information specified as dicts.

Return type*list***first**()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type*obj***not_**(*q=None*, ***kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type*Query***one**()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type*obj***Raises**

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

set_endpoint_names(*endpoint_names*)

Restricts the devices that this query is performed on to the specified endpoint names.

Parameters

endpoint_names (*list*) – List of string endpoint names.

Returns

This instance.

Return type

USBDeviceQuery

set_max_rows(*max_rows*)

Sets the max number of usb devices to fetch in a singular query

Parameters

max_rows (*integer*) – Max number of usb devices

Returns

This instance.

Return type

USBDeviceQuery

Raises

ApiError – If rows is negative or greater than 10000

set_product_names(*product_names*)

Restricts the devices that this query is performed on to the specified product names.

Parameters

product_names (*list*) – List of string product names.

Returns

This instance.

Return type

USBDeviceQuery

set_serial_numbers(*serial_numbers*)

Restricts the devices that this query is performed on to the specified serial numbers.

Parameters

serial_numbers (*list*) – List of string serial numbers.

Returns

This instance.

Return type*USBDeviceQuery***set_statuses**(*statuses*)

Restricts the devices that this query is performed on to the specified status values.

Parameters

statuses (*list*) – List of string status values. Valid values are APPROVED and UNAPPROVED.

Returns

This instance.

Return type*USBDeviceQuery***set_vendor_names**(*vendor_names*)

Restricts the devices that this query is performed on to the specified vendor names.

Parameters

vendor_names (*list*) – List of string vendor names.

Returns

This instance.

Return type*USBDeviceQuery***sort_by**(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Example

```
>>> cb.select(USBDevice).sort_by("product_name")
```

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

This instance.

Return type*USBDeviceQuery***update_criteria**(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

log = <Logger cbc_sdk.endpoint_standard.usb_device_control (WARNING)>

USB Device Control models

4.10 Enterprise EDR Package

4.10.1 Auth Events Module

Model and Query Classes for Auth Events

class AuthEvent(*cb, model_unique_id=None, initial_data=None, force_init=False, full_doc=False*)

Bases: *NewBaseModel*

Represents an AuthEvent

Initialize the AuthEvent object.

Required RBAC Permissions:

org.search.events (CREATE, READ)

Parameters

- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.
- **model_unique_id** (*Any*) – The unique ID for this particular instance of the model object.
- **initial_data** (*dict*) – The data to use when initializing the model object.
- **force_init** (*bool*) – True to force object initialization.
- **full_doc** (*bool*) – False to mark the object as not fully initialized.

Example

```
>>> cb = CBCloudAPI(profile="example_profile")
>>> events = cb.select(AuthEvent).where("auth_username:SYSTEM")
>>> print(*events)
```

static bulk_get_details(*cb*, *alert_id=None*, *event_ids=None*, *timeout=0*)

Bulk get details

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **alert_id** (*str*) – An alert id to fetch associated events
- **event_ids** (*list*) – A list of event ids to fetch
- **timeout** (*int*) – AuthEvent details request timeout in milliseconds. This can never be greater than the configured default timeout. If this value is 0, the configured default timeout is used.

Returns

list of Auth Events

Return type

list

Example

```
>>> cb = CBCloudAPI(profile="example_profile")
>>> bulk_details = AuthEvent.bulk_get_details(cb, event_ids=['example-value'])
>>> print(bulk_details)
```

Raises

[ApiError](#) – if cb is not instance of CBCloudAPI

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

static get_auth_events_descriptions(*cb*)

Returns descriptions and status messages of Auth Events.

Parameters

cb ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.

Returns

Descriptions and status messages of Auth Events as dict objects.

Return type

dict

Raises*ApiError* – if cb is not instance of CBCloudAPI**Example**

```
>>> cb = CBCloudAPI(profile="example_profile")
>>> descriptions = AuthEvent.get_auth_events_descriptions(cb)
>>> print(descriptions)
```

get_details(timeout=0, async_mode=False)

Requests detailed results.

Parameters

- **timeout** (*int*) – AuthEvent details request timeout in milliseconds. This can never be greater than the configured default timeout. If this is 0, the configured default timeout is used.
- **async_mode** (*bool*) – True to request details in an asynchronous manner.

Returns

Auth Events object enriched with the details fields

Return type*AuthEvent*

Note:

- When using asynchronous mode, this method returns a python future. You can call result() on the future object to wait for completion and get the results.
-

Examples

```
>>> cb = CBCloudAPI(profile="example_profile")
```

```
>>> events = cb.select(AuthEvent).where(process_pid=2000)
>>> print(events[0].get_details())
```

refresh()

Reload this object from the server.

static search_suggestions(cb, query, count=None)

Returns suggestions for keys and field values that can be used in a search.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.
- **query** (*str*) – A search query to use.
- **count** (*int*) – (optional) Number of suggestions to be returned

Returns

A list of search suggestions expressed as dict objects.

Return type

list

Raises

ApiError – if cb is not instance of CBCloudAPI

Example

```
>>> cb = CBCloudAPI(profile="example_profile")
>>> suggestions = AuthEvent.search_suggestions(cb, 'auth')
>>> print(suggestions)
```

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class AuthEventFacet(cb, model_unique_id, initial_data)

Bases: *UnrefreshableModel*

Represents an AuthEvent facet retrieved.

Example:

```
>>> cb = CBCloudAPI(profile="example_profile")
>>> events_facet = cb.select(AuthEventFacet).where("auth_username:SYSTEM
↳").add_facet_field("process_name")
>>> print(events_facet.results)
```

Parameters

- **terms** – Contains the Auth Event Facet search results
- **ranges** – Groupings for search result properties that are ISO 8601 timestamps or numbers
- **contacted** – The number of searchers contacted for this query
- **completed** – The number of searchers that have reported their results

Initialize the Terms object with initial data.

class Ranges(cb, initial_data)

Bases: *UnrefreshableModel*

Represents the range (bucketed) facet fields and values associated with an AuthEvent Facet query.

Initialize an AuthEventFacet Ranges object with initial_data.

property facets

Returns the reified *AuthEventFacet.Terms._facets* for this result.

property fields

Returns the ranges fields for this result.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class Terms(*cb*, *initial_data*)

Bases: [*UnrefreshableModel*](#)

Represents the facet fields and values associated with an AuthEvent Facet query.

Initialize an AuthEventFacet Terms object with *initial_data*.

property facets

Returns the terms' facets for this result.

property fields

Returns the terms facets' fields for this result.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

property ranges_

Returns the reified *AuthEventFacet.Ranges* for this result.

refresh()

Reload this object from the server.

property terms_

Returns the reified *AuthEventFacet.Terms* for this result.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class AuthEventGroup(*cb*, *initial_data=None*)

Bases: object

Represents AuthEventGroup

Initialize AuthEventGroup object

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **initial_data** (*dict*) – The data to use when initializing the model object.

Notes

The constructed object will have the following data: - group_start_timestamp - group_end_timestamp - group_key - group_value

Example

```
>>> cb = CBCloudAPI(profile="example_profile")
>>> groups = set(cb.select(AuthEvent).where(process_pid=2000).group_results("device_
->name"))
>>> for group in groups:
>>>     print(group._info)
```

class `AuthEventQuery(doc_class, cb)`

Bases: `Query`

Represents the query logic for an AuthEvent query.

This class specializes `Query` to handle the particulars of Auth Events querying.

Initialize the AuthEventQuery object.

Parameters

- **doc_class** (`class`) – The class of the model this query returns.
- **cb** (`CBCloudAPI`) – A reference to the CBCloudAPI object.

Example

```
>>> cb = CBCloudAPI(profile="example_profile")
>>> events = cb.select(AuthEvent).where("auth_username:SYSTEM")
>>> print(*events)
```

add_criteria(`key, newlist`)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (`str`) – The key for the criteria item to be set.
- **newlist** (`str` or `list[str]`) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

add_exclusions(`key, newlist`)

Add to the exclusions on this query with a custom exclusions key.

Will overwrite any existing exclusion for the specified key.

Parameters

- **key** (`str`) – The key for the exclusion item to be set.
- **newlist** (`str` or `list[str]`) – Value or list of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).add_exclusions("type", ["WATCHLIST"])
>>> query = api.select(Alert).add_exclusions("type", "WATCHLIST")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(q=None, **kwargs)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

batch_size(new_batch_size)

Set the batch size of the paginated query.

Parameters

new_batch_size (*int*) – The new batch size.

Returns

A new query with the updated batch size.

Return type

PaginatedQuery

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

group_results(*fields*, *max_events_per_group*=None, *rows*=500, *start*=None, *range_duration*=None, *range_field*=None, *range_method*=None)

Get group results grouped by provided fields.

Parameters

- **fields** (*str* / *list*) – field or fields by which to perform the grouping
- **max_events_per_group** (*int*) – Maximum number of events in a group, if not provided all events will be returned
- **rows** (*int*) – Number of rows to request, can be paginated
- **start** (*int*) – First row to use for pagination
- **ranges** (*dict*) – dict with information about duration, field, method

Returns

grouped results

Return type

dict

Examples

```
>>> cb = CBCloudAPI(profile="example_profile")
>>> groups = set(cb.select(AuthEvent).where(process_pid=2000).group_results(
    ↪ "device_name"))
>>> for group in groups:
>>>     print(group._info)
```

not_(*q*=None, ***kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(**kwargs)

or_() criteria are explicitly provided to AuthEvent queries.

This method overrides the base class in order to provide **or_()** functionality rather than raising an exception.

Example

```
>>> cb = CCloudAPI(profile="example_profile")
>>> events = cb.select(AuthEvent).where(process_name="chrome.exe").or_(process_
↳ name="firefox.exe")
>>> print(*events)
```

set_fields(fields)

Sets the fields to be returned with the response.

Parameters

fields (str or list[str]) – Field or list of fields to be returned.

set_rows(rows)

Sets the ‘rows’ query body parameter to the ‘start search’ API call, determining how many rows to request.

Parameters

rows (int) – How many rows to request.

Returns

AuthEventQuery object

Return type

Query

Example

```
>>> cb = CCloudAPI(profile="example_profile")
>>> events = cb.select(AuthEvent).where(process_name="chrome.exe").set_rows(5)
>>> print(*events)
```

set_start(start)

Sets the ‘start’ query body parameter, determining where to begin retrieving results from.

Parameters

start (int) – Where to start results from.

set_time_range(start=None, end=None, window=None)

Sets the ‘time_range’ query body parameter, determining a time window based on ‘device_timestamp’.

Parameters

- **start** (str in ISO 8601 timestamp) – When to start the result search.
- **end** (str in ISO 8601 timestamp) – When to end the result search.
- **window** (str) – Time window to execute the result search, ending on the current time. Should be in the form “-2w”, where y=year, w=week, d=day, h=hour, m=minute, s=second.

Note:

- *window* will take precedent over *start* and *end* if provided.

Examples

```
>>> query = api.select(Process).set_time_range(start="2020-10-20T20:34:07Z").
↳ where("query is required")
>>> second_query = api.select(Process).
...     set_time_range(start="2020-10-20T20:34:07Z", end="2020-10-30T20:34:07Z
↳ ").where("query is required")
>>> third_query = api.select(Process).set_time_range(window='-3d').where("query_
↳ is required")
```

sort_by(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

The query with sorting parameters.

Return type

Query

Example

```
>>> cb.select(Process).where(process_name="cmd.exe").sort_by("device_timestamp")
```

timeout(*msecs*)

Sets the timeout on a Auth Event query.

Parameters

msecs (*int*) – Timeout duration, in milliseconds. This value can never be greater than the configured default timeout. If this value is 0, the configured default timeout is used.

Returns

The Query object with new milliseconds parameter.

Return type

Query (AuthEventQuery)

Example

```
>>> cb = CBCloudAPI(profile="example_profile")
>>> events = cb.select(AuthEvent).where(process_name="chrome.exe").timeout(5000)
>>> print(*events)
```

update_criteria(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

update_exclusions(*key, newlist*)

Update the exclusion on this query with a custom exclusion key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*list*) – List of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).update_exclusions("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

4.10.2 Threat Intelligence Module

Model Classes for Enterprise Endpoint Detection and Response

class Feed(*cb, model_unique_id=None, initial_data=None*)

Bases: [FeedModel](#)

Represents an Enterprise EDR feed's metadata.

Parameters

- **name** – A human-friendly name for this feed
- **owner** – The feed owner's connector ID
- **provider_url** – A URL supplied by the feed's provider
- **summary** – A human-friendly summary for the feed
- **category** – The feed's category
- **source_label** – The feed's source label
- **access** – The feed's access (public or private)
- **id** – The feed's unique ID

Initialize the Feed object.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **model_unique_id** (*str*) – The unique ID of the feed.
- **initial_data** (*dict*) – The initial data for the object.

class FeedBuilder(*cb, info*)

Bases: `object`

Helper class allowing Feeds to be assembled.

Creates a new FeedBuilder object.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **info** (*dict*) – The initial information for the new feed.

add_reports(*reports*)

Adds new reports to the new feed.

Parameters

reports (*list* [[Report](#)]) – New reports to be added to the feed.

Returns

This object.

Return type

[FeedBuilder](#)

build()

Builds the new Feed.

Returns

The new Feed.

Return type

[Feed](#)

set_alertable(*alertable*)

Sets the alertable for the new feed. Defaults to true if not specified.

Parameters

alertable (*bool*) – Indicator whether the feed supports alerting.

Returns

This object.

Return type

FeedBuilder

set_category(*category*)

Sets the category for the new feed.

Parameters

category (*str*) – New category for the feed.

Returns

This object.

Return type

FeedBuilder

set_name(*name*)

Sets the name for the new feed.

Parameters

name (*str*) – New name for the feed.

Returns

This object.

Return type

FeedBuilder

set_provider_url(*provider_url*)

Sets the provider URL for the new feed.

Parameters

provider_url (*str*) – New provider URL for the feed.

Returns

This object.

Return type

FeedBuilder

set_source_label(*source_label*)

Sets the source label for the new feed.

Parameters

source_label (*str*) – New source label for the feed.

Returns

This object.

Return type

FeedBuilder

set_summary(*summary*)

Sets the summary for the new feed.

Parameters

summary (*str*) – New summary for the feed.

Returns

This object.

Return type

FeedBuilder

append_reports(*reports*)

Append the given Reports to this Feed's current Reports.

Parameters

reports (*[Report]*) – List of Reports to append to Feed.

Raises

InvalidObjectError – If *id* is missing.

append_reports_rawdata(*report_data*)

Append the given report data, formatted as per the API documentation for reports, to this Feed's Reports.

Parameters

report_data (*list[dict]*) –

Raises

InvalidObjectError – If *id* is missing or validation of the data fails.

classmethod create(*cb, name, provider_url, summary, category, alertable=True*)

Begins creating a new feed by making a FeedBuilder to hold the new feed data.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.
- **name** (*str*) – Name for the new feed.
- **provider_url** (*str*) – Provider URL for the new feed.
- **summary** (*str*) – Summary for the new feed.
- **category** (*str*) – Category for the new feed.

Returns

The new FeedBuilder object to be used to create the feed.

Return type

FeedBuilder

delete()

Deletes this feed from the Enterprise EDR server.

Raises

InvalidObjectError – If *id* is missing.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

refresh()

Reload this object from the server.

replace_reports(reports)

Replace this Feed's Reports with the given Reports.

Parameters

reports (*[Report]*) – List of Reports to replace existing Reports with.

Raises

InvalidObjectError – If *id* is missing.

replace_reports_rawdata(report_data)

Replace this Feed's Reports with the given reports, specified as raw data.

Parameters

report_data (*list[dict]*) –

Raises

InvalidObjectError – If *id* is missing or validation of the data fails.

property reports

Returns a list of Reports associated with this feed.

Returns

List of Reports in this Feed.

Return type

Reports (*[Report]*)

reset()

Undo any changes made to this object's fields.

save(public=False)

Saves this feed on the Enterprise EDR server.

Parameters

public (*bool*) – Whether to make the feed publicly available.

Returns

The saved Feed.

Return type

Feed (Feed)

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(fulltouch=False)

Force this object to be considered as changed.

update(kwargs)**

Update this feed's metadata with the given arguments.

Parameters

****kwargs** (*dict(str, str)*) – The fields to update.

Raises

- ***InvalidObjectError*** – If *id* is missing or `Feed.validate()` fails.
- ***ApiError*** – If an invalid field is specified.

Example

```
>>> feed.update(access="private")
```

validate()

Checks to ensure this feed contains valid data.

Raises

InvalidObjectError – If the feed contains invalid data.

class FeedModel(*cb*, *model_unique_id=None*, *initial_data=None*, *force_init=False*, *full_doc=False*)

Bases: ***UnrefreshableModel***, ***CreatableModelMixin***, ***MutableBaseModel***

A common base class for models used by the Feed and Watchlist APIs.

Initialize the NewBaseModel object.

Parameters

- **cb** (***CBCloudAPI***) – A reference to the CBCloudAPI object.
- **model_unique_id** (*Any*) – The unique ID for this particular instance of the model object.
- **initial_data** (*dict*) – The data to use when initializing the model object.
- **force_init** (*bool*) – True to force object initialization.
- **full_doc** (*bool*) – True to mark the object as fully initialized.

delete()

Delete this object.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

MutableBaseModel

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(*fulltouch=False*)

Force this object to be considered as changed.

validate()

Validates this object.

Returns

True if the object is validated.

Return type

bool

Raises

InvalidObjectError – If the object has missing fields.

class FeedQuery(*doc_class, cb*)

Bases: *SimpleQuery*

Represents the logic for a Feed query.

```
>>> cb.select(Feed)
>>> cb.select(Feed, id)
>>> cb.select(Feed).where(include_public=True)
```

Initialize the FeedQuery object.

Parameters

- **doc_class** (*class*) – The class of the model this query returns.
- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*new_query*)

Add an additional “where” clause to this query.

Parameters

new_query (*object*) – The additional “where” clause, as a string or solrq.Q object.

Returns

A new query with the extra “where” clause specified.

Return type*SimpleQuery***first**()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

property results

Return a list of Feed objects matching self._args parameters.

sort(*new_sort*)

Set the sorting for this query.

Parameters

new_sort (*object*) – The new sort criteria for this query.

Returns

A new query with the sort parameter specified.

Return type*SimpleQuery***where**(***kwargs*)

Add kwargs to self._args dictionary.

class **IOC**(*cb*, *model_unique_id=None*, *initial_data=None*, *report_id=None*)

Bases: *FeedModel*

Represents a collection of categorized IOCs. These objects are officially deprecated and replaced by IOC_V2.

Parameters

- **md5** – A list of MD5 checksums
- **ipv4** – A list of IPv4 addresses
- **ipv6** – A list of IPv6 addresses
- **dns** – A list of domain names
- **query** – A list of dicts, each containing an IOC query

Creates a new IOC instance.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – Unique ID of this IOC.
- **initial_data** (*dict*) – Initial data used to populate the IOC.
- **report_id** (*str*) – ID of the report this IOC belongs to (if this is a watchlist IOC).

Raises

[ApiError](#) – If *initial_data* is None.

delete()

Delete this object.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

[MutableBaseModel](#)

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(*fulltouch=False*)

Force this object to be considered as changed.

validate()

Checks to ensure this IOC contains valid data.

Raises

InvalidObjectError – If the IOC contains invalid data.

class IOC_V2(*cb, model_unique_id=None, initial_data=None, report_id=None*)

Bases: ***FeedModel***

Represents a collection of IOCs of a particular type, plus matching criteria and metadata.

Parameters

- **id** – The IOC_V2’s unique ID
- **match_type** – How IOCs in this IOC_V2 are matched
- **values** – A list of IOCs
- **field** – The kind of IOCs contained in this IOC_V2
- **link** – A URL for some reference for this IOC_V2

Creates a new IOC_V2 instance.

Parameters

- **cb** (***BaseAPI***) – Reference to API object used to communicate with the server.
- **model_unique_id** (*Any*) – Unused.
- **initial_data** (*dict*) – Initial data used to populate the IOC.
- **report_id** (*str*) – ID of the report this IOC belongs to (if this is a watchlist IOC).

Raises

ApiError – If *initial_data* is None.

classmethod create_equality(*cb, iocid, field, *values*)

Creates a new “equality” IOC.

Parameters

- **cb** (***BaseAPI***) – Reference to API object used to communicate with the server.
- **iocid** (*str*) – ID for the new IOC. If this is None, a UUID will be generated for the IOC.
- **field** (*str*) – Name of the field to be matched by this IOC.
- ***values** (*List(str)*) – String values to match against the value of the specified field.

Returns

New IOC data structure.

Return type*IOC_V2***Raises***ApiError* – If there is not at least one value to match against.**classmethod** **create_query**(*cb, iocid, query*)

Creates a new “query” IOC.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **iocid** (*str*) – ID for the new IOC. If this is None, a UUID will be generated for the IOC.
- **query** (*str*) – Query to be incorporated in this IOC.

Returns

New IOC data structure.

Return type*IOC_V2***Raises***ApiError* – If the query string is not present.**classmethod** **create_regex**(*cb, iocid, field, *values*)

Creates a new “regex” IOC.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **iocid** (*str*) – ID for the new IOC. If this is None, a UUID will be generated for the IOC.
- **field** (*str*) – Name of the field to be matched by this IOC.
- ***values** (*list(str)*) – Regular expression values to match against the value of the specified field.

Returns

New IOC data structure.

Return type*IOC_V2***Raises***ApiError* – If there is not at least one regular expression to match against.**delete**()

Delete this object.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

ignore()

Sets the ignore status on this IOC.

Only watchlist IOCs have an ignore status.

Raises

InvalidObjectError – If *id* is missing or this IOC is not from a Watchlist.

property ignored

Returns whether or not this IOC is ignored.

Only watchlist IOCs have an ignore status.

Returns

True if the IOC is ignored, False otherwise.

Return type

bool

Raises

InvalidObjectError – If this IOC is missing an *id* or is not a Watchlist IOC.

Example

```
>>> if ioc.ignored:
...     ioc.unignore()
```

classmethod ipv6_equality_format(*input*)

Turns a canonically-formatted IPv6 address into a string suitable for use in an equality IOC.

Parameters

input (*str*) – The IPv6 address to be translated.

Returns

The translated form of IPv6 address.

Return type

str

Raises

ApiError – If the string is not in valid format.

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

MutableBaseModel

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(*fulltouch=False*)

Force this object to be considered as changed.

unignore()

Removes the ignore status on this IOC.

Only watchlist IOCs have an ignore status.

Raises

InvalidObjectError – If *id* is missing or this IOC is not from a Watchlist.

validate()

Checks to ensure this IOC contains valid FQDN.

Raises

InvalidObjectError – If the IOC contains invalid data.

class Report(*cb, model_unique_id=None, initial_data=None, feed_id=None, from_watchlist=False*)

Bases: *FeedModel*

Represents reports retrieved from an Enterprise EDR feed.

Parameters

- **id** – The report's unique ID
- **timestamp** – When this report was created
- **title** – A human-friendly title for this report
- **description** – A human-friendly description for this report
- **severity** – The severity of the IOCs within this report
- **link** – A URL for some reference for this report
- **tags** – A list of tags for this report
- **iocs_v2** – A list of IOC_V2 dicts associated with this report
- **visibility** – The visibility of this report

Initialize the ReportSeverity object.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.

- **model_unique_id** (*str*) – The ID of the Report (only works for Reports in Watchlists).
- **initial_data** (*dict*) – The initial data for the object.
- **feed_id** (*str*) – The ID of the feed this report is for.
- **from_watchlist** (*bool*) – If the report is in a watchlist

class ReportBuilder(*cb, report_body*)

Bases: object

Helper class allowing Reports to be assembled.

Initialize a new ReportBuilder.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **report_body** (*dict*) – Partial report body which should be filled in with all “required” fields.

add_ioc(*ioc*)

Adds an IOC to the new report.

Parameters

- **ioc** ([IOC_V2](#)) – The IOC to be added to the report.

Returns

This object.

Return type

ReportBuilder

add_tag(*tag*)

Adds a tag value to the new report.

Parameters

- **tag** (*str*) – The new tag for the object.

Returns

This object.

Return type

ReportBuilder

build()

Builds the actual Report from the internal data of the ReportBuilder.

Returns

The new Report.

Return type

Report

set_description(*description*)

Set the description for the new report.

Parameters

- **description** (*str*) – New description for the report.

Returns

This object.

Return type

ReportBuilder

set_link(*link*)

Set the link for the new report.

Parameters

- **link** (*str*) – New link for the report.

Returns

This object.

Return type

ReportBuilder

set_severity(severity)

Set the severity for the new report.

Parameters

severity (*int*) – New severity for the report.

Returns

This object.

Return type

ReportBuilder

set_timestamp(timestamp)

Set the timestamp for the new report.

Parameters

timestamp (*int*) – New timestamp for the report.

Returns

This object.

Return type

ReportBuilder

set_title(title)

Set the title for the new report.

Parameters

title (*str*) – New title for the report.

Returns

This object.

Return type

ReportBuilder

set_visibility(visibility)

Set the visibility for the new report.

Parameters

visibility (*str*) – New visibility for the report.

Returns

This object.

Return type

ReportBuilder

append_iocs(iocs)

Append a list of IOCs to this Report.

Parameters

iocs (*list* [*IOC_V2*]) – List of IOCs to be added.

classmethod create(cb, title, description, severity, timestamp=None, tags=None)

Begin creating a new Report by returning a ReportBuilder.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.
- **title** (*str*) – Title for the new report.
- **description** (*str*) – Description for the new report.
- **severity** (*int*) – Severity value for the new report.

- **timestamp** (*int*) – UNIX-epoch timestamp for the new report. If omitted, current time will be used.
- **tags** (*list[str]*) – Tags to be added to the report. If omitted, there will be none.

Returns

Reference to the ReportBuilder object.

Return type

ReportBuilder

property custom_severity

Returns the custom severity for this report.

Returns

The custom severity for this Report,
if it exists.

Return type

ReportSeverity (ReportSeverity)

Raises

InvalidObjectError – If *id* is missing or this Report is from a Watchlist.

delete()

Deletes this report from the Enterprise EDR server.

Raises

InvalidObjectError – If *id* is missing, or *feed_id* is missing and this report is a Feed Report.

Example

```
>>> report.delete()
```

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

ignore()

Sets the ignore status on this report.

Raises

InvalidObjectError – If *id* is missing or feed ID is missing.

property ignored

Returns the ignore status for this report.

Returns

True if this Report is ignored, False otherwise.

Return type

(bool)

Raises

InvalidObjectError – If *id* is missing or feed ID is missing.

Example

```
>>> if report.ignored:
...     report.unignore()
```

property iocs_

Returns a list of IOC_V2's associated with this report.

Returns

List of IOC_V2's for associated with the Report.

Return type

IOC_V2 ([*IOC_V2*])

Example

```
>>> for ioc in report.iocs_:
...     print(ioc.values)
```

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

refresh()

Reload this object from the server.

remove_iocs(iocs)

Remove a list of IOCs from this Report.

Parameters

iocs (*list* [*IOC_V2*]) – List of IOCs to be removed.

remove_iocs_by_id(ids_list)

Remove IOCs from this report by specifying their IDs.

Parameters

ids_list (*list* [*str*]) – List of IDs of the IOCs to be removed.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

MutableBaseModel

save_watchlist()

Saves this report *as a watchlist report*.

Note: This method **cannot** be used to save a feed report. To save feed reports, create them with *cb.create* and use *Feed.replace*.

This method **cannot** be used to save a report that is *already* part of a watchlist. Use the *update()* method instead.

Raises

InvalidObjectError – If Report.validate() fails.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(fulltouch=False)

Force this object to be considered as changed.

unignore()

Removes the ignore status on this report.

Raises

InvalidObjectError – If *id* is missing or feed ID is missing.

update(kwargs)**

Update this Report with the given arguments.

Parameters

****kwargs** (*dict(str, str)*) – The Report fields to update.

Returns

The updated Report.

Return type

Report (Report)

Raises

InvalidObjectError – If *id* is missing, or *feed_id* is missing and this report is a Feed Report, or Report.validate() fails.

Note: The report's timestamp is always updated, regardless of whether passed explicitly.

```
>>> report.update(title="My new report title")
```

validate()

Checks to ensure this report contains valid data.

Raises

InvalidObjectError – If the report contains invalid data.

class ReportQuery(*doc_class, cb*)

Bases: *SimpleQuery*

Represents the logic for a Report query.

Example

```
>>> cb.select(Report).where(feed_id=id)
>>> cb.select(Report, id)
>>> cb.select(Report, id, from_watchlist=True)
```

Initialize the ReportQuery object.

Parameters

- **doc_class** (*class*) – The class of the model this query returns.
- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*new_query*)

Add an additional “where” clause to this query.

Parameters

new_query (*object*) – The additional “where” clause, as a string or solrq.Q object.

Returns

A new query with the extra “where” clause specified.

Return type

SimpleQuery

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

property results

Return a list of Report objects

sort(*new_sort*)

Set the sorting for this query.

Parameters

new_sort (*object*) – The new sort criteria for this query.

Returns

A new query with the sort parameter specified.

Return type

SimpleQuery

where(kwargs)**

Add kwargs to self._args dictionary.

class ReportSeverity(*cb*, *initial_data=None*)

Bases: *FeedModel*

Represents severity information for a Watchlist Report.

Parameters

- **report_id** – The unique ID for the corresponding report
- **severity** – The severity level

Initialize the ReportSeverity object.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.
- **initial_data** (*dict*) – The initial data for the object.

delete()

Delete this object.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

MutableBaseModel

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(*fulltouch=False*)

Force this object to be considered as changed.

validate()

Validates this object.

Returns

True if the object is validated.

Return type

bool

Raises

InvalidObjectError – If the object has missing fields.

class Watchlist(*cb, model_unique_id=None, initial_data=None*)

Bases: *FeedModel*

Represents an Enterprise EDR watchlist.

Parameters

- **name** – A human-friendly name for the watchlist

- **description** – A short description of the watchlist
- **id** – The watchlist’s unique id
- **tags_enabled** – Whether tags are currently enabled
- **alerts_enabled** – Whether alerts are currently enabled
- **create_timestamp** – When this watchlist was created
- **last_update_timestamp** – Report IDs associated with this watchlist
- **report_ids** – Report IDs associated with this watchlist
- **classifier** – A key, value pair specifying an associated feed

Initialize the Watchlist object.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **model_unique_id** (*str*) – The unique ID of the watch list.
- **initial_data** (*dict*) – The initial data for the object.

class WatchlistBuilder(*cb, name*)

Bases: `object`

Helper class allowing Watchlists to be assembled.

Creates a new WatchlistBuilder object.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **name** (*str*) – Name for the new watchlist.

add_report_ids(*report_ids*)

Adds report IDs to the watchlist.

Parameters

report_ids (*list[str]*) – List of report IDs to add to the watchlist.

Returns

This object.

Return type

[WatchlistBuilder](#)

add_reports(*reports*)

Adds reports to the watchlist.

Parameters

reports (*list[Report]*) – List of reports to be added to the watchlist.

Returns

This object.

Return type

[WatchlistBuilder](#)

build()

Builds the new Watchlist using information in the builder. The new watchlist must still be saved.

Returns

The new Watchlist.

Return type

[Watchlist](#)

set_alerts_enabled(flag)

Sets whether alerts will be enabled on the new watchlist.

Parameters

flag (*bool*) – True to enable alerts, False to disable them. Default is False.

Returns

This object.

Return type

WatchlistBuilder

set_description(description)

Sets the description for the new watchlist.

Parameters

description (*str*) – New description for the watchlist.

Returns

This object.

Return type

WatchlistBuilder

set_name(name)

Sets the name for the new watchlist.

Parameters

name (*str*) – New name for the watchlist.

Returns

This object.

Return type

WatchlistBuilder

set_tags_enabled(flag)

Sets whether tags will be enabled on the new watchlist.

Parameters

flag (*bool*) – True to enable tags, False to disable them. Default is True.

Returns

This object.

Return type

WatchlistBuilder

add_report_ids(report_ids)

Adds new report IDs to the watchlist.

Parameters

report_ids (*list[str]*) – List of report IDs to be added to the watchlist.

add_reports(reports)

Adds new reports to the watchlist.

Parameters

reports (*list[Report]*) – List of reports to be added to the watchlist.

property classifier_

Returns the classifier key and value, if any, for this watchlist.

Returns

Watchlist's classifier key and value. None: If there is no classifier key and value.

Return type

tuple(str, str)

classmethod `create(cb, name)`

Starts creating a new Watchlist by returning a WatchlistBuilder that can be used to set attributes.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **name** (*str*) – Name for the new watchlist.

Returns

The builder for the new watchlist. Call `build()` to create the actual Watchlist.

Return type

[WatchlistBuilder](#)

classmethod `create_from_feed(feed, name=None, description=None, enable_alerts=False, enable_tags=True)`

Creates a new Watchlist that encapsulates a Feed.

Parameters

- **feed** ([Feed](#)) – The feed to be encapsulated by this Watchlist.
- **name** (*str*) – Name for the new watchlist. The default is to use the Feed name.
- **description** (*str*) – Description for the new watchlist. The default is to use the Feed summary.
- **enable_alerts** (*bool*) –
- **enable_tags** (*bool*) –

Returns

A new Watchlist object, which must be saved to the server.

Return type

[Watchlist](#)

delete()

Deletes this watchlist from the Enterprise EDR server.

Raises

[InvalidObjectError](#) – If *id* is missing.

disable_alerts()

Disable alerts for this watchlist.

Raises

[InvalidObjectError](#) – If *id* is missing.

disable_tags()

Disable tagging for this watchlist.

Raises

[InvalidObjectError](#) – if *id* is missing.

enable_alerts()

Enable alerts for this watchlist. Alerts are not retroactive.

Raises

[InvalidObjectError](#) – If *id* is missing.

enable_tags()

Enable tagging for this watchlist.

Raises

InvalidObjectError – If *id* is missing.

property feed

Returns the Feed linked to this Watchlist, if there is one.

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

refresh()

Reload this object from the server.

property reports

Returns a list of Report objects associated with this watchlist.

Returns

List of Reports associated with the watchlist.

Return type

Reports ([*Report*])

Note: If this Watchlist is a classifier (i.e. feed-linked) Watchlist, *reports* will be empty. To get the reports associated with the linked Feed, use feed like:

```
>>> for report in watchlist.feed.reports:
...     print(report.title)
```

reset()

Undo any changes made to this object's fields.

save()

Saves this watchlist on the Enterprise EDR server.

Returns

The saved Watchlist.

Return type*Watchlist (Watchlist)***Raises***InvalidObjectError* – If `Watchlist.validate()` fails.**to_json()**

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(*fulltouch=False*)

Force this object to be considered as changed.

update(*kwargs*)**

Updates this watchlist with the given arguments.

Parameters****kwargs** (*dict(str, str)*) – The fields to update.**Raises**

- *InvalidObjectError* – If *id* is missing or `Watchlist.validate()` fails.
- *ApiError* – If *report_ids* is given and is empty.

Example

```
>>> watchlist.update(name="New Name")
```

validate()

Checks to ensure this watchlist contains valid data.

Raises*InvalidObjectError* – If the watchlist contains invalid data.**class WatchlistQuery(*doc_class, cb*)**Bases: *SimpleQuery*

Represents the logic for a Watchlist query.

```
>>> cb.select(Watchlist)
```

Initialize the WatchlistQuery object.

Parameters

- **doc_class** (*class*) – The class of the model this query returns.
- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*new_query*)

Add an additional “where” clause to this query.

Parameters

new_query (*object*) – The additional “where” clause, as a string or solrq.Q object.

Returns

A new query with the extra “where” clause specified.

Return type*SimpleQuery***first**()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type*obj***one**()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type*obj***Raises**

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

property results

Return a list of all Watchlist objects.

sort(*new_sort*)

Set the sorting for this query.

Parameters

new_sort (*object*) – The new sort criteria for this query.

Returns

A new query with the sort parameter specified.

Return type*SimpleQuery***where**(*new_query*)

Add a “where” clause to this query.

Parameters

new_query (*object*) – The “where” clause, as a string or solrq.Q object.

Returns

A new query with the “where” clause specified.

Return type

SimpleQuery

```
log = <Logger cbc_sdk.enterprise_edr.threat_intelligence (WARNING)>
```

Models

4.10.3 UBS Module

Model Classes for Enterprise Endpoint Detection and Response

class `Binary`(*cb, model_unique_id*)

Bases: *UnrefreshableModel*

Represents a retrievable binary.

Parameters

- **sha256** – The SHA-256 hash of the file
- **md5** – The MD5 hash of the file
- **file_available** – If true, the file is available for download
- **available_file_size** – The size of the file available for download
- **file_size** – The size of the actual file (represented by the hash)
- **os_type** – The OS that this file is designed for
- **architecture** – The set of architectures that this file was compiled for
- **lang_id** – The Language ID value for the Windows VERSIONINFO resource
- **charset_id** – The Character set ID value for the Windows VERSIONINFO resource
- **internal_name** – The internal name from FileVersionInformation
- **product_name** – The product name from FileVersionInformation
- **company_name** – The company name from FileVersionInformation
- **trademark** – The trademark from FileVersionInformation
- **file_description** – The file description from FileVersionInformation
- **file_version** – The file version from FileVersionInformation
- **comments** – Comments from FileVersionInformation
- **original_filename** – The original filename from FileVersionInformation
- **product_description** – The product description from FileVersionInformation
- **product_version** – The product version from FileVersionInformation
- **private_build** – The private build from FileVersionInformation
- **special_build** – The special build from FileVersionInformation

Initialize the Binary object.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.
- **model_unique_id** (*str*) – The SHA-256 of the binary being retrieved.

class `Summary(cb, model_unique_id)`

Bases: [`UnrefreshableModel`](#)

Represents a summary of organization-specific information for a retrievable binary.

Initialize the Summary object.

Parameters

- **cb** ([`CBCloudAPI`](#)) – A reference to the CBCloudAPI object.
- **model_unique_id** (*str*) – The SHA-256 of the binary being retrieved.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

download_url(*expiration_seconds=3600*)

Returns a URL that can be used to download the file for this binary. Returns None if no download found.

Parameters

- **expiration_seconds** (*int*) – How long the download should be valid for.

Returns

A pre-signed AWS download URL. None: If no download is found.

Return type

URL (*str*)

Raises

[`InvalidObjectError`](#) – If the URL retrieval should be retried.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

property summary

Returns organization-specific information about this binary.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class Downloads(*cb, shas, expiration_seconds=3600*)

Bases: [*UnrefreshableModel*](#)

Represents download information for a list of process hashes.

Initialize the Downloads object.

Parameters

- **cb** ([*CBCloudAPI*](#)) – A reference to the CBCloudAPI object.
- **shas** (*list*) – A list of SHA hash values for binaries.
- **expiration_seconds** (*int*) – Number of seconds until this request expires.

class FoundItem(*cb, item*)

Bases: [*UnrefreshableModel*](#)

Represents the download URL and process hash for a successfully located binary.

Initialize the FoundItem object.

Parameters

- **cb** ([*CBCloudAPI*](#)) – A reference to the CBCloudAPI object.
- **item** (*dict*) – The values for a successfully-retrieved item.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

property found

Returns a list of Downloads.FoundItem, one for each binary found in the binary store.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

4.11 Platform Package

4.11.1 Base Module

Model and Query Classes for Platform

class PlatformModel(*cb*, *model_unique_id=None*, *initial_data=None*, *force_init=False*, *full_doc=False*)

Bases: [NewBaseModel](#)

Represents the base of all Platform API model classes.

Initialize the PlatformModel object.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **model_unique_id** (*Any*) – The unique ID for this particular instance of the model object.
- **initial_data** (*dict*) – The data to use when initializing the model object.
- **force_init** (*bool*) – True to force object initialization.
- **full_doc** (*bool*) – True to mark the object as fully initialized.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.

- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

log = <Logger **cbc_sdk.platform.base** (**WARNING**)>

Platform Models

4.11.2 Submodules

4.11.3 Alerts Module

Model and Query Classes for Platform Alerts and Workflows

class Alert(*cb, model_unique_id, initial_data=None*)

Bases: [*PlatformModel*](#)

Represents a basic alert.

Parameters

- **additional_events_present** – Indicator to let API and forwarder users know that they should look up other associated events related to this alert
- **alert_notes_present** – True if notes are present on the alert ID. False if notes are not present.
- **alert_url** – Link to the alerts page for this alert. Does not vary by alert type
- **attack_tactic** – A tactic from the MITRE ATT&CK framework; defines a reason for an adversary’s action, such as achieving credential access
- **attack_technique** – A technique from the MITRE ATT&CK framework; defines an action an adversary takes to accomplish a goal, such as dumping credentials to achieve credential access
- **backend_timestamp** – Timestamp when the Carbon Black Cloud processed and enabled the alert for searching. Corresponds to the Created column on the Alerts page.
- **backend_update_timestamp** – Timestamp when the Carbon Black Cloud initiated and processed an update to an alert. Corresponds to the Updated column on the Alerts page. Note that changes made by users do not change this date; those changes are reflected on *user_update_timestamp*
- **blocked_effective_reputation** – Effective reputation of the blocked file or process; applied by the sensor at the time the block occurred

- **blocked_md5** – MD5 hash of the child process binary; for any process terminated by the sensor
- **blocked_name** – Tokenized file path of the files blocked by sensor action
- **blocked_sha256** – SHA-256 hash of the child process binary; for any process terminated by the sensor
- **category** – Alert category - Monitored vs Threat
- **childproc_cmdline** – Command line for the child process
- **childproc_effective_reputation** – Effective reputation of the child process; applied by the sensor at the time the event occurred
- **childproc_guid** – Unique process identifier assigned to the child process
- **childproc_md5** – Hash of the child process' binary (Enterprise EDR)
- **childproc_name** – Filesystem path of the child process' binary
- **childproc_sha256** – Hash of the child process' binary (Endpoint Standard)
- **childproc_username** – User context in which the child process was executed
- **connection_type** – Connection Type
- **detection_timestamp** – For sensor-sent alerts, this is the time of the event on the sensor. For alerts generated on the backend, this is the time the backend system triggered the alert.
- **determination** – User-updatable determination of the alert
- **device_external_ip** – IP address of the endpoint according to the Carbon Black Cloud; can differ from `device_internal_ip` due to network proxy or NAT; either IPv4 (dotted decimal notation) or IPv6 (proprietary format)
- **device_id** – ID of devices
- **device_internal_ip** – IP address of the endpoint reported by the sensor; either IPv4 (dotted decimal notation) or IPv6 (proprietary format)
- **device_location** – Whether the device was on or off premises when the alert started, based on the current IP address and the device's registered DNS domain suffix
- **device_name** – Device name
- **device_os** – Device Operating Systems
- **device_os_version** – The operating system and version of the endpoint. Requires Windows CBC sensor version 3.5 or later.
- **device_policy** – Device policy
- **device_policy_id** – Device policy id
- **device_target_value** – Target value assigned to the device, set from the policy
- **device_uem_id** – Device correlation with WS1/EUC, required for our Workspace ONE Intelligence integration to function
- **device_username** – Logged on user during the alert. This is filled on a best-effort approach. If the user is not available it may be populated with the device owner (empty for Container Runtime alerts)
- **egress_group_id** – Unique identifier for the egress group
- **egress_group_name** – Name of the egress group

- **external_device_friendly_name** – Human-readable external device names
- **first_event_timestamp** – Timestamp when the first event in the alert occurred
- **id** – Unique IDs of alerts
- **ioc_field** – The field the indicator of comprise (IOC) hit contains
- **ioc_hit** – IOC field value or IOC query that matches
- **ioc_id** – Unique identifier of the IOC that generated the watchlist hit
- **ip_reputation** – Range of reputations to accept for the remote IP 0- unknown 1-20 high risk 21-40 suspicious 41-60 moderate 61-80 low risk 81-100 trustworthy There must be two values in this list. The first is the lower end of the range (inclusive) the second is the upper end of the range (inclusive)
- **is_updated** – Boolean that describes whether or not this is the original copy of the alert
- **k8s_cluster** – K8s Cluster name
- **k8s_kind** – K8s Workload kind
- **k8s_namespace** – K8s namespace
- **k8s_pod_name** – Name of the pod within a workload
- **k8s_policy** – Name of the K8s policy
- **k8s_policy_id** – Unique identifier for the K8s policy
- **k8s_rule** – Name of the K8s policy rule
- **k8s_rule_id** – Unique identifier for the K8s policy rule
- **k8s_workload_name** – K8s Workload Name
- **last_event_timestamp** – Timestamp when the last event in the alert occurred
- **mdr_alert** – Is Mdr alert
- **mdr_alert_notes_present** – Customer visible notes at the alert level that were added by a MDR analyst
- **mdr_determination** – Mdr updatable classification of the alert
- **mdr_threat_notes_present** – Customer visible notes at the threat level that were added by a MDR analyst
- **mdr_workflow** – MDR-updatable workflow of the alert
- **ml_classification_final_verdict** – Final verdict of the alert, based on the ML models that were used to make the prediction.
- **ml_classification_global_prevalence** – Categories (low/medium/high) used to describe the prevalence of alerts across all regional organizations.
- **ml_classification_org_prevalence** – Categories (low/medium/high) used to describe the prevalence of alerts within an organization.
- **netconn_local_ip** – IP address of the remote side of the network connection; stored as dotted decimal
- **netconn_local_ipv4** – IPv4 address of the local side of the network connection; stored as a dotted decimal. Only one of ipv4 and ipv6 fields will be populated.

- **netconn_local_ipv6** – IPv6 address of the local side of the network connection; stored as a string without octet-separating colon characters. Only one of `ipv4` and `ipv6` fields will be populated.
- **netconn_local_port** – TCP or UDP port used by the local side of the network connection
- **netconn_protocol** – Network protocol of the network connection
- **netconn_remote_domain** – Domain name (FQDN) associated with the remote end of the network connection, if available
- **netconn_remote_ip** – IP address of the local side of the network connection; stored as dotted decimal
- **netconn_remote_ipv4** – IPv4 address of the remote side of the network connection; stored as dotted decimal. Only one of `ipv4` and `ipv6` fields will be populated.
- **netconn_remote_ipv6** – IPv6 address of the remote side of the network connection; stored as a string without octet-separating colon characters. Only one of `ipv4` and `ipv6` fields will be populated.
- **netconn_remote_port** – TCP or UDP port used by the remote side of the network connection; same as `netconn_port` and `event_network_remote_port`
- **org_key** – Unique alphanumeric string that identifies your organization in the Carbon Black Cloud
- **parent_cmdline** – Command line of the parent process
- **parent_effective_reputation** – Effective reputation of the parent process; applied by the sensor when the event occurred
- **parent_guid** – Unique process identifier assigned to the parent process
- **parent_md5** – MD5 hash of the parent process binary
- **parent_name** – Filesystem path of the parent process binary
- **parent_pid** – Identifier assigned by the operating system to the parent process
- **parent_reputation** – Reputation of the parent process; applied by the Carbon Black Cloud when the event is initially processed
- **parent_sha256** – SHA-256 hash of the parent process binary
- **parent_username** – User context in which the parent process was executed
- **policy_applied** – Indicates whether or not a policy has been applied to any event associated with this alert
- **primary_event_id** – ID of the primary event in the alert
- **process_cmdline** – Command line executed by the actor process
- **process_effective_reputation** – Effective reputation of the actor hash
- **process_guid** – Guid of the process that has fired the alert (optional)
- **process_md5** – MD5 hash of the actor process binary
- **process_name** – Process names of an alert
- **process_pid** – PID of the process that has fired the alert (optional)
- **process_reputation** – Reputation of the actor process; applied when event is processed by the Carbon Black Cloud

- **process_sha256** – SHA-256 hash of the actor process binary
- **process_username** – User context in which the actor process was executed. MacOS - all users for the PID for fork() and exec() transitions. Linux - process user for exec() events, but in a future sensor release can be multi-valued due to setuid().
- **product_id** – IDs of the product that identifies USB devices
- **product_name** – Names of the product that identifies USB devices
- **reason** – A spoken language written explanation of the what and why the alert occurred and any action taken, usually consisting of 1 to 3 sentences.
- **reason_code** – A unique short-hand code or GUID identifying the particular alert reason
- **remote_is_private** – Is the remote information private
- **remote_k8s_kind** – Kind of remote workload; set if the remote side is another workload in the same cluster
- **remote_k8s_namespace** – Namespace within the remote workload's cluster; set if the remote side is another workload in the same cluster
- **remote_k8s_pod_name** – Remote workload pod name; set if the remote side is another workload in the same cluster
- **remote_k8s_workload_name** – Name of the remote workload; set if the remote side is another workload in the same cluster
- **report_description** – Description of the watchlist report associated with the alert
- **report_id** – Report IDs that contained the IOC that caused a hit
- **report_link** – Link of reports that contained the IOC that caused a hit
- **report_name** – Name of the watchlist report
- **report_tags** – Tags associated with the watchlist report
- **rule_category_id** – ID representing the category of the rule_id for certain alert types
- **rule_config_category** – Types of rule configs
- **rule_id** – ID of the rule that triggered an alert; applies to Intrusion Detection System, Host-Based Firewall, TAU Intelligence, and USB Device Control alerts
- **run_state** – Whether the threat in the alert actually ran
- **sensor_action** – Actions taken by the sensor, according to the rules of a policy
- **serial_number** – Serial numbers of the specific devices
- **severity** – integer representation of the impact of alert if true positive
- **tags** – Tags added to the threat ID of the alert
- **threat_id** – ID assigned to a group of alerts with common criteria, based on alert type
- **threat_name** – Name of the threat
- **threat_notes_present** – True if notes are present on the threat ID. False if notes are not present.
- **tms_rule_id** – Detection id
- **ttps** – Other potential malicious activities involved in a threat
- **type** – Type of alert generated

- **user_update_timestamp** – Timestamp of the last property of an alert changed by a user, such as the alert workflow or determination
- **vendor_id** – IDs of the vendors who produced the devices
- **vendor_name** – Names of the vendors who produced the devices
- **watchlists** – List of watchlists associated with an alert. Alerts are batched hourly
- **workflow** – Current workflow state of an alert. The workflow represents the flow from *OPEN* to *IN_PROGRESS* to *CLOSED* and captures who moved the alert into the current state. The history of these state transitions is available via the alert history route.

Initialize the Alert object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

class Note(*cb, alert, model_unique_id, threat_note=False, initial_data=None*)

Bases: [PlatformModel](#)

Represents a note within an alert.

Parameters

- **author** – User who created the note
- **create_timestamp** – Time the note was created
- **last_update_timestamp** – Time the note was created
- **id** – Unique ID for this note
- **note** – Note contents
- **parent_id** – ID for this note of this notes parent if is a thread

Initialize the Note object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **alert** ([Alert](#)) – The alert where the note is saved.
- **model_unique_id** (*str*) – ID of the note represented.
- **threat_note** (*bool*) – Whether the note is an Alert or Threat note
- **initial_data** (*dict*) – Initial data used to populate the note.

delete()

Deletes a note from an alert.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

add_threat_tags(tags)

Adds tags to the threat

Required Permissions:

org.alerts.tags (CREATE)

Parameters**tags** (*list[str]*) – List of tags to add to the threat**Raises*****ApiError*** – If tags is not a list of strings**Returns**

The list of current tags

Return type

(list[str])

close(closure_reason=None, determination=None, note=None)

Closes this alert.

Parameters

- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.close("RESOLVED", "FALSE_POSITIVE", "Normal behavior")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type

Job

create_note(*note*, *threat_note=False*)

Creates a new note.

Parameters

- **note** (*str*) – Note content to add
- **threat_note** (*bool*) – Whether to add the note to the Alert or Threat

delete_threat_tag(*tag*)

Delete a threat tag

Required Permissions:

org.alerts.tags (DELETE)

Parameters

tag (*str*) – The tag to delete

Returns

The list of current tags

Return type

(list[str])

deobfuscate_cmdline()

Deobfuscates the command line of the process pointed to by the alert and returns the deobfuscated result.

Required Permissions:

script.deobfuscation(EXECUTE)

Returns

A dict containing information about the obfuscated command line, including the deobfuscated result.

Return type

dict

dismiss_threat(*remediation=None*, *comment=None*)

Dismisses all future alerts assigned to the threat_id.

Parameters

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to dismiss all past and current open alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).close(
↪ ..)
```

get(item, default_val=None)

Return an attribute of this object.

Parameters

- **item** (str) – Name of the attribute to be returned.
- **default_val** (Any) – Default value to be used if the attribute is not set.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_history(threat=False)

Get the actions taken on an Alert such as Notes added and workflow state changes.

Parameters

threat (bool) – Whether to return the Alert or Threat history

Returns

The dicts of each determination, note or workflow change

Return type

list

get_observations(timeout=0)

Requests observations that are associated with the Alert.

Uses Observations bulk get details.

Returns

Observations associated with the alert

Return type

list

Note:

- When using asynchronous mode, this method returns a python future. You can call result() on the future object to wait for completion and get the results.

get_process(async_mode=False)

Gets the process corresponding with the alert.

Parameters

async_mode – True to request process in an asynchronous manner.

Returns

The process corresponding to the alert.

Return type

Process

get_threat_tags()

Gets the threat's tags

Required Permissions:

org.alerts.tags (READ)

Returns

The list of current tags

Return type

(list[str])

notes_(threat_note=False)

Retrieves all notes for an alert.

Parameters

threat_note (*bool*) – Whether to return the Alert notes or Threat notes

refresh()

Reload this object from the server.

static search_suggestions(cb, query)

Returns suggestions for keys and field values that can be used in a search.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **query** (*str*) – A search query to use.

Returns

A list of search suggestions expressed as dict objects.

Return type

list

Raises

[ApiError](#) – if cb is not instance of CBCloudAPI

to_json(version='v7')

Return a json object of the response.

Parameters

version (*str*) – version of json to return. Either v6 or v7. DEFAULT v7

Returns

The returned attribute value.

Return type

Any

update(status, closure_reason=None, determination=None, note=None)

Update the Alert with optional closure_reason, determination, note, or status.

Parameters

- **status** (*str*) – The status to set for this alert, either “OPEN”, “IN_PROGRESS”, or “CLOSED”.
- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.update("IN_PROGRESS", "NO_REASON", "NONE", "Starting_
↳Investigation")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type

Job

update_threat(*remediation=None, comment=None*)

Updates all future alerts assigned to the threat_id to the OPEN state.

Parameters

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to update all past and current alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).
↳update(...)
```

property workflow_

Returns the workflow associated with this alert.

Returns

The workflow associated with this alert.

Return type
dict

class AlertSearchQuery(*doc_class, cb*)

Bases: [BaseQuery](#), [QueryBuilderSupportMixin](#), [IterableQueryMixin](#),
[LegacyAlertSearchQueryCriterionMixin](#), [CriteriaBuilderSupportMixin](#),
[ExclusionBuilderSupportMixin](#)

Represents a query that is used to locate Alert objects.

Initialize the AlertSearchQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

add_exclusions(*key, newlist*)

Add to the exclusions on this query with a custom exclusions key.

Will overwrite any existing exclusion for the specified key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).add_exclusions("type", ["WATCHLIST"])
>>> query = api.select(Alert).add_exclusions("type", "WATCHLIST")
```

`add_time_criteria(key, **kwargs)`

Restricts the alerts that this query is performed on to the specified time range for a given key.

The time may either be specified as a start and end point or as a range.

Parameters

- **key** (*str*) – The key to use for criteria one of `create_time`, `first_event_time`, `last_event_time`, `backend_update_timestamp`, or `last_update_time`
- ****kwargs** (*dict*) – Used to specify:
 - `start=` for start time
 - `end=` for end time
 - `range=` for range
 - `excludes=` to set this as an exclusion rather than criteria. Defaults to `False`.

Returns

This instance.

Return type

AlertSearchQuery

Examples

```
>>> query = api.select(Alert).
...     add_time_criteria("detection_timestamp", start="2020-10-20T20:34:07Z",
↪ end="2020-10-30T20:34:07Z")
>>> second_query = api.select(Alert).add_time_criteria("detection_timestamp",
↪ range='-3d')
>>> third_query_legacy = api.select(Alert).set_time_range("create_time", range=
↪ '-3d')
>>> exclusions_query = api.add_time_criteria("detection_timestamp", range="-2h",
↪ exclude=True)
```

`all()`

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

`and_(q=None, **kwargs)`

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

close(*closure_reason=None, determination=None, note=None*)

Close all alerts matching the given query. The alerts will be left in a CLOSED state after this request.

Parameters

- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Returns

The Job object for the bulk workflow action.

Return type

Job

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.

Example

```
>>> alert_query = cb.select(Alert).add_criteria("threat_id", [
    ↪ "19261158DBBF00775959F8AA7F7551A1"])
>>> job = alert_query.close("RESOLVED", "FALSE_POSITIVE", "Normal behavior")
>>> completed_job = job.await_completion().result()
```

facets(*fieldlist, max_rows=0*)

Return information about the facets for this alert by search, using the defined criteria.

Parameters

- **fieldlist** (*list*) – List of facet field names.
- **max_rows** (*int*) – The maximum number of rows to return. 0 means return all rows.

Returns

A list of facet information specified as dicts. error: invalid enum

Return type

list

Raises

- **FunctionalityDecommissioned** – If the requested attribute is no longer available.
- **ApiError** – If the facet field is not valid

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(*q=None, **kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- ***MoreThanOneResultError*** – If the query returns more than one item
- ***ObjectNotFoundError*** – If the query returns zero items

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

set_alert_ids(*alert_ids*)

Restricts the alerts that this query is performed on to the specified alert IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

alert_ids (*list*) – List of string alert IDs.

Returns

This instance.

Return type

AlertSearchQuery

set_alert_notes_present(*is_present*, *exclude=False*)

Restricts the alerts that this query is performed on to those with or without notes.

Parameters

- **is_present** (*bool*) – If true, returns alerts that have a note attached
- **exclude** (*bool*) – If true, will set is_present in the exclusions. Otherwise adds to criteria

Returns

This instance.

Return type

AlertSearchQuery

set_blocked_threat_categories(*categories*)

The field *blocked_threat_category* was deprecated and not included in v7. This method has been removed.

See [Developer Network Alerts v6 Migration](#) for more details.

Args: *categories* (list): List of threat categories to look for.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_categories(*categories*)

The field *categories* was deprecated and not included in v7. This method has been removed.

In Alerts v7, only records with the type THREAT are returned. Records that in v6 had the category MONITORED (Observed) are now Observations See [Developer Network Alerts v6 Migration](#) for more details.

Parameters

categories (*list*) – List of categories to be restricted to.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_cluster_names(*names*)

Restricts the alerts that this query is performed on to the specified Kubernetes cluster names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

names (*list*) – List of Kubernetes cluster names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_create_time(*args, **kwargs)

Restricts the alerts that this query is performed on to the specified creation time.

The time may either be specified as a start and end point or as a range. In SDK 1.5.0 to align with Alerts v7 API, create_time is set as time_range outside of criteria.

Deprecated:

Use *add_time_criteria(field_name, start, end, range)* instead.

Parameters

- ***args** (*list*) – Not used.
- ****kwargs** (*dict*) – Used to specify start= for start time, end= for end time, and range= for range.

Returns

This instance.

Return type

AlertSearchQuery

set_device_ids(*device_ids*)

Restricts the alerts that this query is performed on to the specified device IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

device_ids (*list*) – List of integer device IDs.

Returns

This instance.

Return type

AlertSearchQuery

set_device_locations(*locations*)

Restricts the alerts that this query is performed on to the specified device locations.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

locations (*list*) – List of device locations to look for. Valid values are “ONSITE”, “OFF-SITE”, and “UNKNOWN”.

Returns

This instance.

Return type

CBAnalyticsAlertSearchQuery

set_device_names(*device_names*)

Restricts the alerts that this query is performed on to the specified device names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

device_names (*list*) – List of string device names.

Returns

This instance.

Return type

AlertSearchQuery

set_device_os(device_os)

Restricts the alerts that this query is performed on to the specified device operating systems.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

device_os (*list*) – List of string operating systems. Valid values are “WINDOWS”, “ANDROID”, “MAC”, “IOS”, “LINUX”, and “OTHER.”

Returns

This instance.

Return type

AlertSearchQuery

set_device_os_versions(device_os_versions)

Restricts the alerts that this query is performed on to the specified device operating system versions.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

device_os_versions (*list*) – List of string operating system versions.

Returns

This instance.

Return type

AlertSearchQuery

set_device_username(users)

Restricts the alerts that this query is performed on to the specified user names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

users (*list*) – List of string user names.

Returns

This instance.

Return type

AlertSearchQuery

set_egress_group_ids(ids)

Restricts the alerts that this query is performed on to the specified egress group IDs.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

ids (*list*) – List of egress group IDs to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_egress_group_names(*names*)

Restricts the alerts that this query is performed on to the specified egress group names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

names (*list*) – List of egress group names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_external_device_friendly_names(*names*)

Restricts the alerts that this query is performed on to the specified external device friendly names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

names (*list*) – List of external device friendly names to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_external_device_ids(*ids*)

Restricts the alerts that this query is performed on to the specified external device IDs.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

ids (*list*) – List of external device IDs to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_group_by(*field*)

Converts the AlertSearchQuery to a GroupAlertSearchQuery grouped by the argument

Parameters

field (*string*) – The field to group by, defaults to “threat_id”

Returns

AlertSearchQuery

Note: Does not preserve sort criterion

set_group_results(*do_group*)

The field *group_results* was deprecated and not included in v7. This method has been removed.

It previously specified whether to group the results of the query. Use the [Grouped Alerts Operations](#) (#grouped-alerts-operations) instead. See [Developer Network Alerts v6 Migration](#) for more details.

Parameters

do_group (*bool*) – True to group the results, False to not do so.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_ip_reputations(*reputations*)

Restricts the alerts that this query is performed on to the specified IP reputation values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

reputations (*list*) – List of IP reputation values to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_kill_chain_statuses(*statuses*)

The field *kill_chain_status* was deprecated and not included in v7. This method has been removed.

See [Developer Network Alerts v6 Migration](#) for more details.

Args: statuses (list): List of kill chain statuses to look for.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_legacy_alert_ids(*alert_ids*)

Restricts the alerts that this query is performed on to the specified legacy alert IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

alert_ids (*list*) – List of string legacy alert IDs.

Returns

This instance.

Return type*AlertSearchQuery***set_minimum_severity**(*severity*)

Restricts the alerts that this query is performed on to the specified minimum severity level.

Parameters

severity (*int*) – The minimum severity level for alerts.

Returns

This instance.

Return type*AlertSearchQuery***set_namespaces**(*namespaces*)

Restricts the alerts that this query is performed on to the specified Kubernetes namespaces.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

namespaces (*list*) – List of Kubernetes namespaces to look for.

Returns

This instance.

Return type*ContainerRuntimeAlertSearchQuery***set_not_blocked_threat_categories**(*categories*)

The field *not_blocked_threat_category* was deprecated and not included in v7. This method has been removed.

See [Developer Network Alerts v6 Migration](#) for more details.

Args: categories (list): List of threat categories to look for.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_policy_applied(*applied_statuses*)

Restricts the alerts that this query is performed on to the specified policy status values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

applied_statuses (*list*) – List of status values to look for. Valid values are “APPLIED” and “NOT_APPLIED”.

Returns

This instance.

Return type*CBAnalyticsAlertSearchQuery*

set_policy_ids(*policy_ids*)

Restricts the alerts that this query is performed on to the specified policy IDs.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

policy_ids (*list*) – List of integer policy IDs.

Returns

This instance.

Return type

AlertSearchQuery

set_policy_names(*policy_names*)

Restricts the alerts that this query is performed on to the specified policy names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

policy_names (*list*) – List of string policy names.

Returns

This instance.

Return type

AlertSearchQuery

set_ports(*ports*)

Restricts the alerts that this query is performed on to the specified netconn_local_ports.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Note that in SDK 1.5.0, to align with Alerts API v7, the search field was updated from *port* to *netconn_local_port*. It is possible to search on either *netconn_local_port* or *netconn_remote_port* using the `add_criteria(fieldname, [field values])` method.

Parameters

ports (*list*) – List of netconn_local_ports to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_process_names(*process_names*)

Restricts the alerts that this query is performed on to the specified process names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

process_names (*list*) – List of string process names.

Returns

This instance.

Return type

AlertSearchQuery

set_process_sha256(*shas*)

Restricts the alerts that this query is performed on to the specified process SHA-256 hash values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

shas (*list*) – List of string process SHA-256 hash values.

Returns

This instance.

Return type

AlertSearchQuery

set_product_ids(*ids*)

Restricts the alerts that this query is performed on to the specified product IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

ids (*list*) – List of product IDs to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_product_names(*names*)

Restricts the alerts that this query is performed on to the specified product names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

names (*list*) – List of product names to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_protocols(*protocols*)

Restricts the alerts that this query is performed on to the specified protocols.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

protocols (*list*) – List of protocols to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_reason_code(*reason*)

Restricts the alerts that this query is performed on to the specified reason codes (enum values).

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

reason (*list*) – List of string reason codes to look for.

Returns

This instance.

Return type

CBAalyticsAlertSearchQuery

set_remote_domains(*domains*)

Restricts the alerts that this query is performed on to the specified remote domains.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

domains (*list*) – List of remote domains to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_remote_ips(*addrs*)

Restricts the alerts that this query is performed on to the specified remote IP addresses.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

addrs (*list*) – List of remote IP addresses to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_remote_is_private(*is_private*, *exclude=False*)

Restricts the alerts that this query is performed on based on matching the remote_is_private field.

This field is only present on CONTAINER_RUNTIME alerts and so filtering will be ignored on other alert types.

Parameters

- **is_private** (*boolean*) – Whether the remote information is private: true or false
- **exclude** (*bool*) – If true, will set is_present in the exclusions. Otherwise adds to criteria

Returns

This instance.

Return type

AlertSearchQuery

set_replica_ids(ids)

Restricts the alerts that this query is performed on to the specified pod names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

ids (*list*) – List of pod names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_reputations(reps)

Restricts the alerts that this query is performed on to the specified reputation values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

reps (*list*) – List of string reputation values. Valid values are “KNOWN_MALWARE”, “SUSPECT_MALWARE”, “PUP”, “NOT_LISTED”, “ADAPTIVE_WHITE_LIST”, “COMMON_WHITE_LIST”, “TRUSTED_WHITE_LIST”, and “COMPANY_BLACK_LIST”.

Returns

This instance.

Return type

AlertSearchQuery

set_rows(rows)

Sets the ‘rows’ query body parameter, determining how many rows of results to request.

Parameters

rows (*int*) – How many rows to request.

set_rule_ids(ids)

Restricts the alerts that this query is performed on to the specified Kubernetes policy rule IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

In SDK prior to 1.5.0 this was only supported for Container Runtime Alerts so will convert to `k8s_rule_id` in criteria. In SDK 1.5.0 and later, aligned to Alert v7 API, use `add_criteria()` should be used for both `k8s_rule_id` and for other alert types, `rule_id`.

Parameters

ids (*list*) – List of Kubernetes policy rule IDs to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_rule_names(*names*)

Restricts the alerts that this query is performed on to the specified Kubernetes policy rule names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

names (*list*) – List of Kubernetes policy rule names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_run_states(*states*)

Restricts the alerts that this query is performed on to the specified run states.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

states (*list*) – List of run states to look for. Valid values are “DID_NOT_RUN”, “RAN”, and “UNKNOWN”.

Returns

This instance.

Return type

CBAalyticsAlertSearchQuery

set_sensor_actions(*actions*)

Restricts the alerts that this query is performed on to the specified sensor actions.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

actions (*list*) – List of sensor actions to look for. Valid values are “POLICY_NOT_APPLIED”, “ALLOW”, “ALLOW_AND_LOG”, “TERMINATE”, and “DENY”.

Returns

This instance.

Return type

CBAalyticsAlertSearchQuery

set_serial_numbers(*serial_numbers*)

Restricts the alerts that this query is performed on to the specified serial numbers.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

serial_numbers (*list*) – List of serial numbers to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_tags(tags)

Restricts the alerts that this query is performed on to the specified tag values.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

tags (*list*) – List of string tag values.

Returns

This instance.

Return type

AlertSearchQuery

set_target_priorities(priorities)

Restricts the alerts that this query is performed on to the specified target priority values.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

priorities (*list*) – List of string target priority values. Valid values are “LOW”, “MEDIUM”, “HIGH”, and “MISSION_CRITICAL”.

Returns

This instance.

Return type

AlertSearchQuery

set_threat_cause_vectors(vectors)

The field `threat_cause_vector` was deprecated and not included in v7. This method has been removed.

See [Developer Network Alerts v6 Migration](#) for more details.

Parameters

vectors (*list*) – List of threat cause vectors to look for.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_threat_ids(threats)

Restricts the alerts that this query is performed on to the specified threat ID values.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

threats (*list*) – List of string threat ID values.

Returns

This instance.

Return type*AlertSearchQuery***set_threat_notes_present**(*is_present*, *exclude=False*)

Restricts the alerts that this query is performed on to those with or without threat_notes.

Parameters

- **is_present** (*bool*) – If true, returns alerts that have a note attached to the threat_id
- **exclude** (*bool*) – If true, will set is_present in the exclusions. Otherwise adds to criteria

Returns

This instance.

Return type*AlertSearchQuery***set_time_range**(*args, **kwargs)

For v7 Alerts:

Sets the 'time_range' query body parameter, determining a time range based on 'backend_timestamp'.

Parameters

- ***args** – not used
 - ****kwargs** (*dict*) – Used to specify the period to search within
 - start= either timestamp ISO 8601 strings or datetime objects
 - end= either timestamp ISO 8601 strings or datetime objects
 - range= the period on which to execute the result search, ending on the current time.
- Range must be in the format “-<quantity><units>” where quantity is an integer, and units is one of:
- M: month(s)
 - w: week(s)
 - d: day(s)
 - h: hour(s)
 - m: minute(s)
 - s: second(s)

For v6 Alerts (backwards compatibility):

Restricts the alerts that this query is performed on to the specified time range for a given key. Will set the 'time_range' as in the v7 usage if key is create_time and set a criteria value for any other valid key.

Parameters

- **key** (*str*) – The key to use for criteria one of create_time, first_event_time, last_event_time or last_update_time. i.e. legacy field names from the Alert v6 API.
- ****kwargs** (*dict*) – Used to specify the period to search within
 - start= either timestamp ISO 8601 strings or datetime objects
 - end= either timestamp ISO 8601 strings or datetime objects
 - range= the period on which to execute the result search, ending on the current time.

Returns

This instance.

Return type

AlertSearchQuery

Examples

```
>>> query_specify_start_and_end = api.select(Alert).  
...     set_time_range(start="2020-10-20T20:34:07Z", end="2020-10-30T20:34:07Z")  
>>> query_specify_range = api.select(Alert).set_time_range(range='-3d')  
>>> query_legacy_use = api.select(Alert).set_time_range("create_time", range='-  
↪3d')
```

set_types(alerttypes)

Restricts the alerts that this query is performed on to the specified alert type values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

alerttypes (*list*) – List of string alert type values. Valid values are “CB_ANALYTICS”, “WATCHLIST”, “DEVICE_CONTROL”, and “CONTAINER_RUNTIME”. In SDK 1.5.0, to align with Alert API v7, more alert types are available but the *add_criteria* method must be used.

Returns

This instance.

Return type

AlertSearchQuery

Note: - When filtering by fields that take a list parameter, an empty list will be treated as a wildcard and match everything.

set_vendor_ids(ids)

Restricts the alerts that this query is performed on to the specified vendor IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

ids (*list*) – List of vendor IDs to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_vendor_names(names)

Restricts the alerts that this query is performed on to the specified vendor names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

names (*list*) – List of vendor names to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_watchlist_ids(ids)

Restricts the alerts that this query is performed on to the specified watchlist ID values.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

ids (*list*) – List of string watchlist ID values.

Returns

This instance.

Return type

WatchlistAlertSearchQuery

set_watchlist_names(names)

Restricts the alerts that this query is performed on to the specified watchlist name values.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

names (*list*) – List of string watchlist name values.

Returns

This instance.

Return type

WatchlistAlertSearchQuery

set_workflows(workflow_vals)

Restricts the alerts that this query is performed on to the specified workflow status values.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

workflow_vals (*list*) – List of string alert type values. Valid values are “OPEN” and “DISMISSED”.

Returns

This instance.

Return type

AlertSearchQuery

set_workload_ids(*ids*)

The field *workload_id* was deprecated and not included in v7. This method has been removed.

Use *workload_name* instead. See [Developer Network Alerts v6 Migration](#) for more details.

Parameters

ids (*list*) – List of workload IDs to look for.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_workload_kinds(*kinds*)

Restricts the alerts that this query is performed on to the specified workload types.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

kinds (*list*) – List of workload types to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_workload_names(*names*)

Restricts the alerts that this query is performed on to the specified workload names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

names (*list*) – List of workload names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

sort_by(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Example

```
>>> cb.select(Alert).sort_by("name")
```

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

This instance.

Return type

AlertSearchQuery

update(*status*, *closure_reason*=None, *determination*=None, *note*=None)

Update all alerts matching the given query.

Parameters

- **status** (*str*) – The status to set for this alert, either “OPEN”, “IN_PROGRESS”, or “CLOSED”.
- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Returns

The Job object for the bulk workflow action.

Return type

Job

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert_query = cb.select(Alert).add_criteria("threat_id", [
↳ "19261158DBBF00775959F8AA7F7551A1"])
>>> job = alert_query.update("IN_PROGRESS", "NO_REASON", "NONE", "Starting_
↳ Investigation")
>>> completed_job = job.await_completion().result()
```

update_criteria(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

`update_exclusions(key, newlist)`

Update the exclusion on this query with a custom exclusion key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*list*) – List of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).update_exclusions("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

`where(q=None, **kwargs)`

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, `QueryBuilder`, or `solrq.Q` object
- ****kwargs** (*dict*) – Arguments to construct a `solrq.Q` with

Returns

This Query object.

Return type

Query

```
class CBAnalyticsAlert(cb, model_unique_id, initial_data=None)
```

Bases: *Alert*

Represents CB Analytics alerts.

Parameters

- **additional_events_present** – Indicator to let API and forwarder users know that they should look up other associated events related to this alert
- **alert_notes_present** – True if notes are present on the alert ID. False if notes are not present.
- **alert_url** – Link to the alerts page for this alert. Does not vary by alert type

- **attack_tactic** – A tactic from the MITRE ATT&CK framework; defines a reason for an adversary’s action, such as achieving credential access
- **attack_technique** – A technique from the MITRE ATT&CK framework; defines an action an adversary takes to accomplish a goal, such as dumping credentials to achieve credential access
- **backend_timestamp** – Timestamp when the Carbon Black Cloud processed and enabled the alert for searching. Corresponds to the Created column on the Alerts page.
- **backend_update_timestamp** – Timestamp when the Carbon Black Cloud initiated and processed an update to an alert. Corresponds to the Updated column on the Alerts page. Note that changes made by users do not change this date; those changes are reflected on *user_update_timestamp*
- **blocked_effective_reputation** – Effective reputation of the blocked file or process; applied by the sensor at the time the block occurred
- **blocked_md5** – MD5 hash of the child process binary; for any process terminated by the sensor
- **blocked_name** – Tokenized file path of the files blocked by sensor action
- **blocked_sha256** – SHA-256 hash of the child process binary; for any process terminated by the sensor
- **category** – Alert category - Monitored vs Threat
- **childproc_cmdline** – Command line for the child process
- **childproc_effective_reputation** – Effective reputation of the child process; applied by the sensor at the time the event occurred
- **childproc_guid** – Unique process identifier assigned to the child process
- **childproc_md5** – Hash of the child process’ binary (Enterprise EDR)
- **childproc_name** – Filesystem path of the child process’ binary
- **childproc_sha256** – Hash of the child process’ binary (Endpoint Standard)
- **childproc_username** – User context in which the child process was executed
- **detection_timestamp** – For sensor-sent alerts, this is the time of the event on the sensor. For alerts generated on the backend, this is the time the backend system triggered the alert.
- **determination** – User-updatable determination of the alert
- **device_external_ip** – IP address of the endpoint according to the Carbon Black Cloud; can differ from *device_internal_ip* due to network proxy or NAT; either IPv4 (dotted decimal notation) or IPv6 (proprietary format)
- **device_id** – ID of devices
- **device_internal_ip** – IP address of the endpoint reported by the sensor; either IPv4 (dotted decimal notation) or IPv6 (proprietary format)
- **device_location** – Whether the device was on or off premises when the alert started, based on the current IP address and the device’s registered DNS domain suffix
- **device_name** – Device name
- **device_os** – Device Operating Systems
- **device_os_version** – The operating system and version of the endpoint. Requires Windows CBC sensor version 3.5 or later.

- **device_policy** – Device policy
- **device_policy_id** – Device policy id
- **device_target_value** – Target value assigned to the device, set from the policy
- **device_uem_id** – Device correlation with WS1/EUC, required for our Workspace ONE Intelligence integration to function
- **device_username** – Logged on user during the alert. This is filled on a best-effort approach. If the user is not available it may be populated with the device owner (empty for Container Runtime alerts)
- **first_event_timestamp** – Timestamp when the first event in the alert occurred
- **id** – Unique IDs of alerts
- **is_updated** – Boolean that describes whether or not this is the original copy of the alert
- **last_event_timestamp** – Timestamp when the last event in the alert occurred
- **netconn_local_ip** – IP address of the remote side of the network connection; stored as dotted decimal
- **netconn_local_ipv4** – IPv4 address of the local side of the network connection; stored as a dotted decimal. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_local_ipv6** – IPv6 address of the local side of the network connection; stored as a string without octet-separating colon characters. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_local_port** – TCP or UDP port used by the local side of the network connection
- **netconn_protocol** – Network protocol of the network connection
- **netconn_remote_domain** – Domain name (FQDN) associated with the remote end of the network connection, if available
- **netconn_remote_ip** – IP address of the local side of the network connection; stored as dotted decimal
- **netconn_remote_ipv4** – IPv4 address of the remote side of the network connection; stored as dotted decimal. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_remote_ipv6** – IPv6 address of the remote side of the network connection; stored as a string without octet-separating colon characters. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_remote_port** – TCP or UDP port used by the remote side of the network connection; same as netconn_port and event_network_remote_port
- **org_key** – Unique alphanumeric string that identifies your organization in the Carbon Black Cloud
- **parent_cmdline** – Command line of the parent process
- **parent_effective_reputation** – Effective reputation of the parent process; applied by the sensor when the event occurred
- **parent_guid** – Unique process identifier assigned to the parent process
- **parent_md5** – MD5 hash of the parent process binary
- **parent_name** – Filesystem path of the parent process binary
- **parent_pid** – Identifier assigned by the operating system to the parent process

- **parent_reputation** – Reputation of the parent process; applied by the Carbon Black Cloud when the event is initially processed
- **parent_sha256** – SHA-256 hash of the parent process binary
- **parent_username** – User context in which the parent process was executed
- **policy_applied** – Indicates whether or not a policy has been applied to any event associated with this alert
- **primary_event_id** – ID of the primary event in the alert
- **process_cmdline** – Command line executed by the actor process
- **process_effective_reputation** – Effective reputation of the actor hash
- **process_guid** – Guid of the process that has fired the alert (optional)
- **process_md5** – MD5 hash of the actor process binary
- **process_name** – Process names of an alert
- **process_pid** – PID of the process that has fired the alert (optional)
- **process_reputation** – Reputation of the actor process; applied when event is processed by the Carbon Black Cloud
- **process_sha256** – SHA-256 hash of the actor process binary
- **process_username** – User context in which the actor process was executed. MacOS - all users for the PID for fork() and exec() transitions. Linux - process user for exec() events, but in a future sensor release can be multi-valued due to setuid().
- **reason** – A spoken language written explanation of the what and why the alert occurred and any action taken, usually consisting of 1 to 3 sentences.
- **reason_code** – A unique short-hand code or GUID identifying the particular alert reason
- **rule_category_id** – ID representing the category of the rule_id for certain alert types
- **rule_id** – ID of the rule that triggered an alert; applies to Intrusion Detection System, Host-Based Firewall, TAU Intelligence, and USB Device Control alerts
- **run_state** – Whether the threat in the alert actually ran
- **sensor_action** – Actions taken by the sensor, according to the rules of a policy
- **severity** – integer representation of the impact of alert if true positive
- **tags** – Tags added to the threat ID of the alert
- **threat_id** – ID assigned to a group of alerts with common criteria, based on alert type
- **threat_notes_present** – True if notes are present on the threat ID. False if notes are not present.
- **ttps** – Other potential malicious activities involved in a threat
- **type** – Type of alert generated
- **user_update_timestamp** – Timestamp of the last property of an alert changed by a user, such as the alert workflow or determination
- **workflow** – Current workflow state of an alert. The workflow represents the flow from *OPEN* to *IN_PROGRESS* to *CLOSED* and captures who moved the alert into the current state. The history of these state transitions is available via the alert history route.

Initialize the Alert object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

class Note(*cb, alert, model_unique_id, threat_note=False, initial_data=None*)

Bases: [PlatformModel](#)

Represents a note within an alert.

Parameters

- **author** – User who created the note
- **create_timestamp** – Time the note was created
- **last_update_timestamp** – Time the note was created
- **id** – Unique ID for this note
- **note** – Note contents
- **parent_id** – ID for this note of this notes parent if is a thread

Initialize the Note object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **alert** ([Alert](#)) – The alert where the note is saved.
- **model_unique_id** (*str*) – ID of the note represented.
- **threat_note** (*bool*) – Whether the note is an Alert or Threat note
- **initial_data** (*dict*) – Initial data used to populate the note.

delete()

Deletes a note from an alert.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

add_threat_tags(*tags*)

Adds tags to the threat

Required Permissions:

org.alerts.tags (CREATE)

Parameters

tags (*list[str]*) – List of tags to add to the threat

Raises

ApiError – If tags is not a list of strings

Returns

The list of current tags

Return type

(list[str])

close(*closure_reason=None, determination=None, note=None*)

Closes this alert.

Parameters

- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.close("RESOLVED", "FALSE_POSITIVE", "Normal behavior")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type

Job

create_note(*note, threat_note=False*)

Creates a new note.

Parameters

- **note** (*str*) – Note content to add
- **threat_note** (*bool*) – Whether to add the note to the Alert or Threat

delete_threat_tag(*tag*)

Delete a threat tag

Required Permissions:

org.alerts.tags (DELETE)

Parameters

tag (*str*) – The tag to delete

Returns

The list of current tags

Return type

(list[*str*])

deobfuscate_cmdline()

Deobfuscates the command line of the process pointed to by the alert and returns the deobfuscated result.

Required Permissions:

script.deobfuscation(EXECUTE)

Returns

A dict containing information about the obfuscated command line, including the deobfuscated result.

Return type

dict

dismiss_threat(*remediation=None, comment=None*)

Dismisses all future alerts assigned to the threat_id.

Parameters

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to dismiss all past and current open alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).close().  
↪ ..)
```

get(*item, default_val=None*)

Return an attribute of this object.

Parameters

- **item** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_events(*timeout=0, async_mode=False*)

Removed in CBC SDK 1.5.0 because Enriched Events are deprecated.

Previously requested enriched events detailed results. Update to use `get_observations()` instead. See [Developer Network Observations Migration](#) for more details.

Parameters

- **timeout** (*int*) – Event details request timeout in milliseconds.
- **async_mode** (*bool*) – True to request details in an asynchronous manner.

Returns

EnrichedEvents matching the legacy_alert_id

Return type

list

Note:

- When using asynchronous mode, this method returns a python future. You can call `result()` on the future object to wait for completion and get the results.
-

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

get_history(*threat=False*)

Get the actions taken on an Alert such as Notes added and workflow state changes.

Parameters

threat (*bool*) – Whether to return the Alert or Threat history

Returns

The dicts of each determination, note or workflow change

Return type

list

get_observations(*timeout=0*)

Requests observations that are associated with the Alert.

Uses Observations bulk get details.

Returns

Observations associated with the alert

Return type

list

Note:

- When using asynchronous mode, this method returns a python future. You can call `result()` on the future object to wait for completion and get the results.

get_process(*async_mode=False*)

Gets the process corresponding with the alert.

Parameters

async_mode – True to request process in an asynchronous manner.

Returns

The process corresponding to the alert.

Return type

Process

get_threat_tags()

Gets the threat's tags

Required Permissions:

org.alerts.tags (READ)

Returns

The list of current tags

Return type

(list[str])

notes_(*threat_note=False*)

Retrieves all notes for an alert.

Parameters

threat_note (*bool*) – Whether to return the Alert notes or Threat notes

refresh()

Reload this object from the server.

static search_suggestions(*cb, query*)

Returns suggestions for keys and field values that can be used in a search.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.
- **query** (*str*) – A search query to use.

Returns

A list of search suggestions expressed as dict objects.

Return type

list

Raises

ApiError – if cb is not instance of CBCloudAPI

to_json(*version='v7'*)

Return a json object of the response.

Parameters

version (*str*) – version of json to return. Either v6 or v7. DEFAULT v7

Returns

The returned attribute value.

Return type

Any

update(*status*, *closure_reason*=None, *determination*=None, *note*=None)Update the Alert with optional *closure_reason*, *determination*, *note*, or *status*.**Parameters**

- **status** (*str*) – The status to set for this alert, either “OPEN”, “IN_PROGRESS”, or “CLOSED”.
- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.update("IN_PROGESS", "NO_REASON", "NONE", "Starting_
↳Investigation")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type*Job***update_threat**(*remediation*=None, *comment*=None)Updates all future alerts assigned to the *threat_id* to the OPEN state.**Parameters**

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to update all past and current alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).
↳update(...)
```

property workflow_

Returns the workflow associated with this alert.

Returns

The workflow associated with this alert.

Return type

dict

class ContainerRuntimeAlert(*cb, model_unique_id, initial_data=None*)

Bases: [*Alert*](#)

Represents Container Runtime alerts.

Parameters

- **alert_notes_present** – True if notes are present on the alert ID. False if notes are not present.
- **alert_url** – Link to the alerts page for this alert. Does not vary by alert type
- **backend_timestamp** – Timestamp when the Carbon Black Cloud processed and enabled the alert for searching. Corresponds to the Created column on the Alerts page.
- **backend_update_timestamp** – Timestamp when the Carbon Black Cloud initiated and processed an update to an alert. Corresponds to the Updated column on the Alerts page. Note that changes made by users do not change this date; those changes are reflected on *user_update_timestamp*
- **connection_type** – Connection Type
- **detection_timestamp** – For sensor-sent alerts, this is the time of the event on the sensor. For alerts generated on the backend, this is the time the backend system triggered the alert.
- **determination** – User-updatable determination of the alert
- **egress_group_id** – Unique identifier for the egress group
- **egress_group_name** – Name of the egress group
- **first_event_timestamp** – Timestamp when the first event in the alert occurred
- **id** – Unique IDs of alerts
- **ip_reputation** – Range of reputations to accept for the remote IP 0- unknown 1-20 high risk 21-40 suspicious 41-60 moderate 61-80 low risk 81-100 trustworthy There must be two values in this list. The first is the lower end of the range (inclusive) the second is the upper end of the range (inclusive)
- **is_updated** – Boolean that describes whether or not this is the original copy of the alert
- **k8s_cluster** – K8s Cluster name
- **k8s_kind** – K8s Workload kind
- **k8s_namespace** – K8s namespace
- **k8s_pod_name** – Name of the pod within a workload
- **k8s_policy** – Name of the K8s policy
- **k8s_policy_id** – Unique identifier for the K8s policy
- **k8s_rule** – Name of the K8s policy rule
- **k8s_rule_id** – Unique identifier for the K8s policy rule

- **k8s_workload_name** – K8s Workload Name
- **last_event_timestamp** – Timestamp when the last event in the alert occurred
- **netconn_local_ip** – IP address of the remote side of the network connection; stored as dotted decimal
- **netconn_local_ipv4** – IPv4 address of the local side of the network connection; stored as a dotted decimal. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_local_ipv6** – IPv6 address of the local side of the network connection; stored as a string without octet-separating colon characters. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_local_port** – TCP or UDP port used by the local side of the network connection
- **netconn_protocol** – Network protocol of the network connection
- **netconn_remote_domain** – Domain name (FQDN) associated with the remote end of the network connection, if available
- **netconn_remote_ip** – IP address of the local side of the network connection; stored as dotted decimal
- **netconn_remote_ipv4** – IPv4 address of the remote side of the network connection; stored as dotted decimal. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_remote_ipv6** – IPv6 address of the remote side of the network connection; stored as a string without octet-separating colon characters. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_remote_port** – TCP or UDP port used by the remote side of the network connection; same as netconn_port and event_network_remote_port
- **org_key** – Unique alphanumeric string that identifies your organization in the Carbon Black Cloud
- **policy_applied** – Indicates whether or not a policy has been applied to any event associated with this alert
- **primary_event_id** – ID of the primary event in the alert
- **reason** – A spoken language written explanation of the what and why the alert occurred and any action taken, usually consisting of 1 to 3 sentences.
- **reason_code** – A unique short-hand code or GUID identifying the particular alert reason
- **remote_is_private** – Is the remote information private
- **remote_k8s_kind** – Kind of remote workload; set if the remote side is another workload in the same cluster
- **remote_k8s_namespace** – Namespace within the remote workload's cluster; set if the remote side is another workload in the same cluster
- **remote_k8s_pod_name** – Remote workload pod name; set if the remote side is another workload in the same cluster
- **remote_k8s_workload_name** – Name of the remote workload; set if the remote side is another workload in the same cluster
- **run_state** – Whether the threat in the alert actually ran
- **sensor_action** – Actions taken by the sensor, according to the rules of a policy
- **severity** – integer representation of the impact of alert if true positive

- **tags** – Tags added to the threat ID of the alert
- **threat_id** – ID assigned to a group of alerts with common criteria, based on alert type
- **threat_notes_present** – True if notes are present on the threat ID. False if notes are not present.
- **type** – Type of alert generated
- **user_update_timestamp** – Timestamp of the last property of an alert changed by a user, such as the alert workflow or determination
- **workflow** – Current workflow state of an alert. The workflow represents the flow from *OPEN* to *IN_PROGRESS* to *CLOSED* and captures who moved the alert into the current state. The history of these state transitions is available via the alert history route.

Initialize the Alert object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

class Note(*cb, alert, model_unique_id, threat_note=False, initial_data=None*)

Bases: [PlatformModel](#)

Represents a note within an alert.

Parameters

- **author** – User who created the note
- **create_timestamp** – Time the note was created
- **last_update_timestamp** – Time the note was created
- **id** – Unique ID for this note
- **note** – Note contents
- **parent_id** – ID for this note of this notes parent if is a thread

Initialize the Note object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **alert** ([Alert](#)) – The alert where the note is saved.
- **model_unique_id** (*str*) – ID of the note represented.
- **threat_note** (*bool*) – Whether the note is an Alert or Threat note
- **initial_data** (*dict*) – Initial data used to populate the note.

delete()

Deletes a note from an alert.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

add_threat_tags(tags)

Adds tags to the threat

Required Permissions:

org.alerts.tags (CREATE)

Parameters

tags (*list[str]*) – List of tags to add to the threat

Raises

ApiError – If tags is not a list of strings

Returns

The list of current tags

Return type

(list[str])

close(closure_reason=None, determination=None, note=None)

Closes this alert.

Parameters

- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.close("RESOLVED", "FALSE_POSITIVE", "Normal behavior")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type

Job

create_note(*note*, *threat_note=False*)

Creates a new note.

Parameters

- **note** (*str*) – Note content to add
- **threat_note** (*bool*) – Whether to add the note to the Alert or Threat

delete_threat_tag(*tag*)

Delete a threat tag

Required Permissions:

org.alerts.tags (DELETE)

Parameters

tag (*str*) – The tag to delete

Returns

The list of current tags

Return type

(list[*str*])

deobfuscate_cmdline()

Deobfuscates the command line of the process pointed to by the alert and returns the deobfuscated result.

Required Permissions:

script.deobfuscation(EXECUTE)

Returns

A dict containing information about the obfuscated command line, including the deobfuscated result.

Return type

dict

dismiss_threat(*remediation=None*, *comment=None*)

Dismisses all future alerts assigned to the threat_id.

Parameters

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to dismiss all past and current open alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).close(
↪ ..)
```

get(item, default_val=None)

Return an attribute of this object.

Parameters

- **item** (str) – Name of the attribute to be returned.
- **default_val** (Any) – Default value to be used if the attribute is not set.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_history(threat=False)

Get the actions taken on an Alert such as Notes added and workflow state changes.

Parameters

threat (bool) – Whether to return the Alert or Threat history

Returns

The dicts of each determination, note or workflow change

Return type

list

get_observations(timeout=0)

Requests observations that are associated with the Alert.

Uses Observations bulk get details.

Returns

Observations associated with the alert

Return type

list

Note:

- When using asynchronous mode, this method returns a python future. You can call result() on the future object to wait for completion and get the results.

get_process(async_mode=False)

Gets the process corresponding with the alert.

Parameters

async_mode – True to request process in an asynchronous manner.

Returns

The process corresponding to the alert.

Return type

Process

get_threat_tags()

Gets the threat's tags

Required Permissions:

org.alerts.tags (READ)

Returns

The list of current tags

Return type

(list[str])

notes_(threat_note=False)

Retrieves all notes for an alert.

Parameters

threat_note (*bool*) – Whether to return the Alert notes or Threat notes

refresh()

Reload this object from the server.

static search_suggestions(cb, query)

Returns suggestions for keys and field values that can be used in a search.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **query** (*str*) – A search query to use.

Returns

A list of search suggestions expressed as dict objects.

Return type

list

Raises

[ApiError](#) – if cb is not instance of CBCloudAPI

to_json(version='v7')

Return a json object of the response.

Parameters

version (*str*) – version of json to return. Either v6 or v7. DEFAULT v7

Returns

The returned attribute value.

Return type

Any

update(status, closure_reason=None, determination=None, note=None)

Update the Alert with optional closure_reason, determination, note, or status.

Parameters

- **status** (*str*) – The status to set for this alert, either “OPEN”, “IN_PROGRESS”, or “CLOSED”.
- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.update("IN_PROGESS", "NO_REASON", "NONE", "Starting_
↳Investigation")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type

Job

update_threat(*remediation=None, comment=None*)

Updates all future alerts assigned to the threat_id to the OPEN state.

Parameters

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to update all past and current alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).
↳update(...)
```

property workflow_

Returns the workflow associated with this alert.

Returns

The workflow associated with this alert.

Return type

dict

class DeviceControlAlert(*cb, model_unique_id, initial_data=None*)Bases: [Alert](#)

Represents Device Control alerts.

Parameters

- **alert_notes_present** – True if notes are present on the alert ID. False if notes are not present.
- **alert_url** – Link to the alerts page for this alert. Does not vary by alert type
- **backend_timestamp** – Timestamp when the Carbon Black Cloud processed and enabled the alert for searching. Corresponds to the Created column on the Alerts page.
- **backend_update_timestamp** – Timestamp when the Carbon Black Cloud initiated and processed an update to an alert. Corresponds to the Updated column on the Alerts page. Note that changes made by users do not change this date; those changes are reflected on *user_update_timestamp*
- **detection_timestamp** – For sensor-sent alerts, this is the time of the event on the sensor. For alerts generated on the backend, this is the time the backend system triggered the alert.
- **determination** – User-updatable determination of the alert
- **device_external_ip** – IP address of the endpoint according to the Carbon Black Cloud; can differ from *device_internal_ip* due to network proxy or NAT; either IPv4 (dotted decimal notation) or IPv6 (proprietary format)
- **device_id** – ID of devices
- **device_internal_ip** – IP address of the endpoint reported by the sensor; either IPv4 (dotted decimal notation) or IPv6 (proprietary format)
- **device_location** – Whether the device was on or off premises when the alert started, based on the current IP address and the device’s registered DNS domain suffix
- **device_name** – Device name
- **device_os** – Device Operating Systems
- **device_os_version** – The operating system and version of the endpoint. Requires Windows CBC sensor version 3.5 or later.
- **device_policy** – Device policy
- **device_policy_id** – Device policy id
- **device_target_value** – Target value assigned to the device, set from the policy
- **device_uem_id** – Device correlation with WS1/EUC, required for our Workspace ONE Intelligence integration to function
- **device_username** – Logged on user during the alert. This is filled on a best-effort approach. If the user is not available it may be populated with the device owner (empty for Container Runtime alerts)
- **external_device_friendly_name** – Human-readable external device names
- **first_event_timestamp** – Timestamp when the first event in the alert occurred
- **id** – Unique IDs of alerts

- **is_updated** – Boolean that describes whether or not this is the original copy of the alert
- **last_event_timestamp** – Timestamp when the last event in the alert occurred
- **org_key** – Unique alphanumeric string that identifies your organization in the Carbon Black Cloud
- **policy_applied** – Indicates whether or not a policy has been applied to any event associated with this alert
- **primary_event_id** – ID of the primary event in the alert
- **product_id** – IDs of the product that identifies USB devices
- **product_name** – Names of the product that identifies USB devices
- **reason** – A spoken language written explanation of the what and why the alert occurred and any action taken, usually consisting of 1 to 3 sentences.
- **reason_code** – A unique short-hand code or GUID identifying the particular alert reason
- **run_state** – Whether the threat in the alert actually ran
- **sensor_action** – Actions taken by the sensor, according to the rules of a policy
- **serial_number** – Serial numbers of the specific devices
- **severity** – integer representation of the impact of alert if true positive
- **tags** – Tags added to the threat ID of the alert
- **threat_id** – ID assigned to a group of alerts with common criteria, based on alert type
- **threat_notes_present** – True if notes are present on the threat ID. False if notes are not present.
- **type** – Type of alert generated
- **user_update_timestamp** – Timestamp of the last property of an alert changed by a user, such as the alert workflow or determination
- **vendor_id** – IDs of the vendors who produced the devices
- **vendor_name** – Names of the vendors who produced the devices
- **workflow** – Current workflow state of an alert. The workflow represents the flow from *OPEN* to *IN_PROGRESS* to *CLOSED* and captures who moved the alert into the current state. The history of these state transitions is available via the alert history route.

Initialize the Alert object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

class Note(*cb, alert, model_unique_id, threat_note=False, initial_data=None*)

Bases: [PlatformModel](#)

Represents a note within an alert.

Parameters

- **author** – User who created the note
- **create_timestamp** – Time the note was created

- **last_update_timestamp** – Time the note was created
- **id** – Unique ID for this note
- **note** – Note contents
- **parent_id** – ID for this note of this notes parent if is a thread

Initialize the Note object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **alert** ([Alert](#)) – The alert where the note is saved.
- **model_unique_id** (*str*) – ID of the note represented.
- **threat_note** (*bool*) – Whether the note is an Alert or Threat note
- **initial_data** (*dict*) – Initial data used to populate the note.

delete()

Deletes a note from an alert.

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

add_threat_tags(tags)

Adds tags to the threat

Required Permissions:

org.alerts.tags (CREATE)

Parameters

tags (*list[str]*) – List of tags to add to the threat

Raises

[ApiError](#) – If tags is not a list of strings

Returns

The list of current tags

Return type

(*list[str]*)

close(*closure_reason=None, determination=None, note=None*)

Closes this alert.

Parameters

- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.close("RESOLVED", "FALSE_POSITIVE", "Normal behavior")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type

Job

create_note(*note, threat_note=False*)

Creates a new note.

Parameters

- **note** (*str*) – Note content to add
- **threat_note** (*bool*) – Whether to add the note to the Alert or Threat

delete_threat_tag(*tag*)

Delete a threat tag

Required Permissions:

org.alerts.tags (DELETE)

Parameters

tag (*str*) – The tag to delete

Returns

The list of current tags

Return type

(list[str])

deobfuscate_cmdline()

Deobfuscates the command line of the process pointed to by the alert and returns the deobfuscated result.

Required Permissions:

script.deobfuscation(EXECUTE)

Returns

A dict containing information about the obfuscated command line, including the deobfuscated result.

Return type

dict

dismiss_threat(*remediation=None, comment=None*)

Dismisses all future alerts assigned to the threat_id.

Parameters

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to dismiss all past and current open alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).close(
    ↪ ..)
```

get(*item, default_val=None*)

Return an attribute of this object.

Parameters

- **item** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_history(*threat=False*)

Get the actions taken on an Alert such as Notes added and workflow state changes.

Parameters

threat (*bool*) – Whether to return the Alert or Threat history

Returns

The dicts of each determination, note or workflow change

Return type

list

get_observations(*timeout=0*)

Requests observations that are associated with the Alert.

Uses Observations bulk get details.

Returns

Observations associated with the alert

Return type

list

Note:

- When using asynchronous mode, this method returns a python future. You can call `result()` on the future object to wait for completion and get the results.
-

get_process(*async_mode=False*)

Gets the process corresponding with the alert.

Parameters

async_mode – True to request process in an asynchronous manner.

Returns

The process corresponding to the alert.

Return type

Process

get_threat_tags()

Gets the threat's tags

Required Permissions:

org.alerts.tags (READ)

Returns

The list of current tags

Return type

(list[str])

notes(*threat_note=False*)

Retrieves all notes for an alert.

Parameters

threat_note (*bool*) – Whether to return the Alert notes or Threat notes

refresh()

Reload this object from the server.

static search_suggestions(*cb, query*)

Returns suggestions for keys and field values that can be used in a search.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.
- **query** (*str*) – A search query to use.

Returns

A list of search suggestions expressed as dict objects.

Return type

list

Raises

ApiError – if cb is not instance of CBCloudAPI

to_json(*version*='v7')

Return a json object of the response.

Parameters

version (*str*) – version of json to return. Either v6 or v7. DEFAULT v7

Returns

The returned attribute value.

Return type

Any

update(*status*, *closure_reason*=None, *determination*=None, *note*=None)

Update the Alert with optional closure_reason, determination, note, or status.

Parameters

- **status** (*str*) – The status to set for this alert, either “OPEN”, “IN_PROGRESS”, or “CLOSED”.
- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.update("IN_PROGESS", "NO_REASON", "NONE", "Starting_
↳Investigation")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type

Job

update_threat(*remediation=None, comment=None*)

Updates all future alerts assigned to the threat_id to the OPEN state.

Parameters

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to update all past and current alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).
    ↪ update(...)
```

property workflow_

Returns the workflow associated with this alert.

Returns

The workflow associated with this alert.

Return type

dict

class GroupedAlert(*cb, model_unique_id, initial_data=None*)

Bases: [PlatformModel](#)

Represents Grouped alerts.

Parameters

- **count** – Count of individual alerts that are a part of the group
- **determination_values** – Map of determination (TRUE_POSITIVE, FALSE_POSITIVE, NONE) to the number of individual alerts in the group with that determination. Determinations with no alerts are omitted.
- **ml_classification_final_verdicts** – Map of ML classification (ANOMALOUS, NOT_ANOMALOUS, NO_PREDICTION) to the number of individual alerts in the group with that classification. Classifications with no alerts are omitted.
- **workflow_states** – Map of workflow state (OPEN, IN_PROGRESS, CLOSED) to the number of individual alerts in the group in that state. States with no alerts are omitted.
- **device_count** – Count of unique devices where this alert can be found
- **first_alert_timestamp** – Timestamp of the first (oldest) alert in the group
- **highest_severity** – Highest severity score of all alerts in the group
- **last_alert_timestamp** – Timestamp of the last (newest) alert in the group
- **most_recent_alert** – The most recent alert in the group. Follows the Alerts Schema and returns an [Alert](#) object. Specific fields vary between alert instances
- **policy_applied** – APPLIED, when any of the alerts in the group had actions blocked by the sensor due to a policy. NOT_APPLIED otherwise.
- **tags** – List of tags that have been applied to the threat ID
- **threat_notes_present** – Whether there are threat-level notes available on this threat ID

- **workload_count** – Count of unique Kubernetes workloads where this alert can be found

Initialize the Grouped Alert object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_alert_search_query()

Returns the Alert Search Query needed to pull all alerts for a given Group Alert.

Returns

for all alerts associated with the calling group alert.

Return type

[AlertSearchQuery](#)

Note: Does not preserve sort criterion

get_alerts()

Returns the all alerts for a given Group Alert.

Returns

alerts associated with the calling group alert.

Return type

list

property most_recent_alert_

Returns the most recent alert for a given group alert.

Returns

the most recent alert in the Group Alert.

Return type

[Alert](#)

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class `GroupedAlertSearchQuery(*args, **kwargs)`

Bases: `AlertSearchQuery`

Represents a query that is used to group Alert objects by a given field.

Initialize the GroupAlertSearchQuery.

add_criteria(*key*, *newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

add_exclusions(*key*, *newlist*)

Add to the exclusions on this query with a custom exclusions key.

Will overwrite any existing exclusion for the specified key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).add_exclusions("type", ["WATCHLIST"])
>>> query = api.select(Alert).add_exclusions("type", "WATCHLIST")
```

add_time_criteria(*key*, ***kwargs*)

Restricts the alerts that this query is performed on to the specified time range for a given key.

The time may either be specified as a start and end point or as a range.

Parameters

- **key** (*str*) – The key to use for criteria one of `create_time`, `first_event_time`, `last_event_time`, `backend_update_timestamp`, or `last_update_time`

- ****kwargs** (*dict*) – Used to specify:
 - start= for start time
 - end= for end time
 - range= for range
 - excludes= to set this as an exclusion rather than criteria. Defaults to False.

Returns

This instance.

Return type

AlertSearchQuery

Examples

```
>>> query = api.select(Alert).  
...     add_time_criteria("detection_timestamp", start="2020-10-20T20:34:07Z",  
↪ end="2020-10-30T20:34:07Z")  
>>> second_query = api.select(Alert).add_time_criteria("detection_timestamp",  
↪ range='-3d')  
>>> third_query_legacy = api.select(Alert).set_time_range("create_time", range=  
↪ '-3d')  
>>> exclusions_query = api.add_time_criteria("detection_timestamp", range="-2h",  
↪ exclude=True)
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(q=None, **kwargs)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

close(closure_reason=None, determination=None, note=None)

Closing all alerts matching a grouped alert query is not implemented.

Note:

- Closing all alerts in all groups returned by a *GroupedAlertSearchQuery* can be done by getting the *AlertSearchQuery* and using *close()* on it as shown in the following example.
-

Example

```
>>> alert_query = grouped_alert_query.get_alert_search_query()
>>> alert_query.close(closure_reason, determination, note)
```

facets(*fieldlist*, *max_rows=0*, *filter_values=False*)

Return information about the facets for this alert by search, using the defined criteria.

Parameters

- **fieldlist** (*list*) – List of facet field names.
- **max_rows** (*int*) – The maximum number of rows to return. 0 means return all rows.
- **filter_values** (*boolean*) – A flag to indicate whether any filters on a term should be applied to facet
- **false** (*calculation. When*) –

Returns

A list of facet information specified as dicts. error: invalid enum

Return type

list

Raises

- **FunctionalityDecommissioned** – If the requested attribute is no longer available.
- **ApiError** – If the facet field is not valid

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

get_alert_search_query()

Converts the GroupedAlertSearchQuery into a nongrouped AlertSearchQuery

Returns: AlertSearchQuery

Note: Does not preserve sort criterion

not_(*q=None*, ***kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- *q* (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

set_alert_ids(*alert_ids*)

Restricts the alerts that this query is performed on to the specified alert IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

alert_ids (*list*) – List of string alert IDs.

Returns

This instance.

Return type

AlertSearchQuery

set_alert_notes_present(*is_present, exclude=False*)

Restricts the alerts that this query is performed on to those with or without notes.

Parameters

- **is_present** (*bool*) – If true, returns alerts that have a note attached
- **exclude** (*bool*) – If true, will set *is_present* in the exclusions. Otherwise adds to criteria

Returns

This instance.

Return type

AlertSearchQuery

set_blocked_threat_categories(*categories*)

The field *blocked_threat_category* was deprecated and not included in v7. This method has been removed.

See [Developer Network Alerts v6 Migration](#) for more details.

Args: *categories* (list): List of threat categories to look for.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_categories(*categories*)

The field *categories* was deprecated and not included in v7. This method has been removed.

In Alerts v7, only records with the type THREAT are returned. Records that in v6 had the category MONITORED (Observed) are now Observations See [Developer Network Alerts v6 Migration](#) for more details.

Parameters

categories (list) – List of categories to be restricted to.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_cluster_names(*names*)

Restricts the alerts that this query is performed on to the specified Kubernetes cluster names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

names (list) – List of Kubernetes cluster names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_create_time(*args, **kwargs)

Restricts the alerts that this query is performed on to the specified creation time.

The time may either be specified as a start and end point or as a range. In SDK 1.5.0 to align with Alerts v7 API, *create_time* is set as *time_range* outside of criteria.

Deprecated:

Use *add_time_criteria(field_name, start, end, range)* instead.

Parameters

- ***args** (list) – Not used.
- ****kwargs** (dict) – Used to specify *start=* for start time, *end=* for end time, and *range=* for range.

Returns

This instance.

Return type

AlertSearchQuery

set_device_ids(*device_ids*)

Restricts the alerts that this query is performed on to the specified device IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

device_ids (*list*) – List of integer device IDs.

Returns

This instance.

Return type

AlertSearchQuery

set_device_locations(*locations*)

Restricts the alerts that this query is performed on to the specified device locations.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

locations (*list*) – List of device locations to look for. Valid values are “ONSITE”, “OFF-SITE”, and “UNKNOWN”.

Returns

This instance.

Return type

CBAanalyticsAlertSearchQuery

set_device_names(*device_names*)

Restricts the alerts that this query is performed on to the specified device names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

device_names (*list*) – List of string device names.

Returns

This instance.

Return type

AlertSearchQuery

set_device_os(*device_os*)

Restricts the alerts that this query is performed on to the specified device operating systems.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

device_os (*list*) – List of string operating systems. Valid values are “WINDOWS”, “ANDROID”, “MAC”, “IOS”, “LINUX”, and “OTHER.”

Returns

This instance.

Return type*AlertSearchQuery***set_device_os_versions**(*device_os_versions*)

Restricts the alerts that this query is performed on to the specified device operating system versions.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

device_os_versions (*list*) – List of string operating system versions.

Returns

This instance.

Return type*AlertSearchQuery***set_device_username**(*users*)

Restricts the alerts that this query is performed on to the specified user names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

users (*list*) – List of string user names.

Returns

This instance.

Return type*AlertSearchQuery***set_egress_group_ids**(*ids*)

Restricts the alerts that this query is performed on to the specified egress group IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

ids (*list*) – List of egress group IDs to look for.

Returns

This instance.

Return type*ContainerRuntimeAlertSearchQuery***set_egress_group_names**(*names*)

Restricts the alerts that this query is performed on to the specified egress group names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

names (*list*) – List of egress group names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_external_device_friendly_names(*names*)

Restricts the alerts that this query is performed on to the specified external device friendly names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

names (*list*) – List of external device friendly names to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_external_device_ids(*ids*)

Restricts the alerts that this query is performed on to the specified external device IDs.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

ids (*list*) – List of external device IDs to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_group_by(*field*)

Sets the ‘group_by’ query body parameter, determining which field to group the alerts by.

Parameters

field (*string*) – The field to group by

set_group_results(*do_group*)

The field `group_results` was deprecated and not included in v7. This method has been removed.

It previously specified whether to group the results of the query. Use the [Grouped Alerts Operations #grouped-alerts-operations](#) instead. See [Developer Network Alerts v6 Migration](#) for more details.

Parameters

do_group (*bool*) – True to group the results, False to not do so.

Raises

[FunctionalityDecommissioned](#) – If the requested attribute is no longer available.

set_ip_reputations(*reputations*)

Restricts the alerts that this query is performed on to the specified IP reputation values.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

reputations (*list*) – List of IP reputation values to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_kill_chain_statuses(*statuses*)

The field *kill_chain_status* was deprecated and not included in v7. This method has been removed.

See [Developer Network Alerts v6 Migration](#) for more details.

Args: statuses (list): List of kill chain statuses to look for.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_legacy_alert_ids(*alert_ids*)

Restricts the alerts that this query is performed on to the specified legacy alert IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

alert_ids (*list*) – List of string legacy alert IDs.

Returns

This instance.

Return type

AlertSearchQuery

set_minimum_severity(*severity*)

Restricts the alerts that this query is performed on to the specified minimum severity level.

Parameters

severity (*int*) – The minimum severity level for alerts.

Returns

This instance.

Return type

AlertSearchQuery

set_namespaces(*namespaces*)

Restricts the alerts that this query is performed on to the specified Kubernetes namespaces.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

namespaces (*list*) – List of Kubernetes namespaces to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_not_blocked_threat_categories(*categories*)

The field *not_blocked_threat_category* was deprecated and not included in v7. This method has been removed.

See [Developer Network Alerts v6 Migration](#) for more details.

Args: *categories* (list): List of threat categories to look for.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_policy_applied(*applied_statuses*)

Restricts the alerts that this query is performed on to the specified policy status values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

applied_statuses (*list*) – List of status values to look for. Valid values are “APPLIED” and “NOT_APPLIED”.

Returns

This instance.

Return type

CBAnalyticsAlertSearchQuery

set_policy_ids(*policy_ids*)

Restricts the alerts that this query is performed on to the specified policy IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

policy_ids (*list*) – List of integer policy IDs.

Returns

This instance.

Return type

AlertSearchQuery

set_policy_names(*policy_names*)

Restricts the alerts that this query is performed on to the specified policy names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

policy_names (*list*) – List of string policy names.

Returns

This instance.

Return type

AlertSearchQuery

set_ports(*ports*)

Restricts the alerts that this query is performed on to the specified `netconn_local_ports`.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Note that in SDK 1.5.0, to align with Alerts API v7, the search field was updated from `port` to `netconn_local_port`. It is possible to search on either `netconn_local_port` or `netconn_remote_port` using the `add_criteria(fieldname, [field values])` method.

Parameters

ports (*list*) – List of `netconn_local_ports` to look for.

Returns

This instance.

Return type

`ContainerRuntimeAlertSearchQuery`

set_process_names(*process_names*)

Restricts the alerts that this query is performed on to the specified process names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

process_names (*list*) – List of string process names.

Returns

This instance.

Return type

AlertSearchQuery

set_process_sha256(*shas*)

Restricts the alerts that this query is performed on to the specified process SHA-256 hash values.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

shas (*list*) – List of string process SHA-256 hash values.

Returns

This instance.

Return type

AlertSearchQuery

set_product_ids(*ids*)

Restricts the alerts that this query is performed on to the specified product IDs.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

ids (*list*) – List of product IDs to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_product_names(*names*)

Restricts the alerts that this query is performed on to the specified product names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

names (*list*) – List of product names to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_protocols(*protocols*)

Restricts the alerts that this query is performed on to the specified protocols.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

protocols (*list*) – List of protocols to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_reason_code(*reason*)

Restricts the alerts that this query is performed on to the specified reason codes (enum values).

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

reason (*list*) – List of string reason codes to look for.

Returns

This instance.

Return type

CBAalyticsAlertSearchQuery

set_remote_domains(*domains*)

Restricts the alerts that this query is performed on to the specified remote domains.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

domains (*list*) – List of remote domains to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_remote_ips(*addrs*)

Restricts the alerts that this query is performed on to the specified remote IP addresses.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

addrs (*list*) – List of remote IP addresses to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_remote_is_private(*is_private*, *exclude=False*)

Restricts the alerts that this query is performed on based on matching the remote_is_private field.

This field is only present on CONTAINER_RUNTIME alerts and so filtering will be ignored on other alert types.

Parameters

- **is_private** (*boolean*) – Whether the remote information is private: true or false
- **exclude** (*bool*) – If true, will set is_present in the exclusions. Otherwise adds to criteria

Returns

This instance.

Return type*AlertSearchQuery***set_replica_ids**(*ids*)

Restricts the alerts that this query is performed on to the specified pod names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

ids (*list*) – List of pod names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_reputations(*reps*)

Restricts the alerts that this query is performed on to the specified reputation values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

reps (*list*) – List of string reputation values. Valid values are “KNOWN_MALWARE”, “SUSPECT_MALWARE”, “PUP”, “NOT_LISTED”, “ADAPTIVE_WHITE_LIST”,

“COMMON_WHITE_LIST”, “TRUSTED_WHITE_LIST”, and “COM-PANY_BLACK_LIST”.

Returns

This instance.

Return type

AlertSearchQuery

set_rows(rows)

Sets the ‘rows’ query body parameter, determining how many rows of results to request.

Parameters

rows (*int*) – How many rows to request.

set_rule_ids(ids)

Restricts the alerts that this query is performed on to the specified Kubernetes policy rule IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

In SDK prior to 1.5.0 this was only supported for Container Runtime Alerts so will convert to `k8s_rule_id` in criteria. In SDK 1.5.0 and later, aligned to Alert v7 API, use *add_criteria()* should be used for both `k8s_rule_id` and for other alert types, `rule_id`.

Parameters

ids (*list*) – List of Kubernetes policy rule IDs to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_rule_names(names)

Restricts the alerts that this query is performed on to the specified Kubernetes policy rule names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

names (*list*) – List of Kubernetes policy rule names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_run_states(states)

Restricts the alerts that this query is performed on to the specified run states.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

states (*list*) – List of run states to look for. Valid values are “DID_NOT_RUN”, “RAN”, and “UNKNOWN”.

Returns

This instance.

Return type

CBAalyticsAlertSearchQuery

set_sensor_actions(*actions*)

Restricts the alerts that this query is performed on to the specified sensor actions.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

actions (*list*) – List of sensor actions to look for. Valid values are “POLICY_NOT_APPLIED”, “ALLOW”, “ALLOW_AND_LOG”, “TERMINATE”, and “DENY”.

Returns

This instance.

Return type

CBAalyticsAlertSearchQuery

set_serial_numbers(*serial_numbers*)

Restricts the alerts that this query is performed on to the specified serial numbers.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

serial_numbers (*list*) – List of serial numbers to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_tags(*tags*)

Restricts the alerts that this query is performed on to the specified tag values.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

tags (*list*) – List of string tag values.

Returns

This instance.

Return type*AlertSearchQuery***set_target_priorities**(*priorities*)

Restricts the alerts that this query is performed on to the specified target priority values.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

priorities (*list*) – List of string target priority values. Valid values are “LOW”, “MEDIUM”, “HIGH”, and “MISSION_CRITICAL”.

Returns

This instance.

Return type

AlertSearchQuery

set_threat_cause_vectors(*vectors*)

The field *threat_cause_vector* was deprecated and not included in v7. This method has been removed.

See [Developer Network Alerts v6 Migration](#) for more details.

Parameters

vectors (*list*) – List of threat cause vectors to look for.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_threat_ids(*threats*)

Restricts the alerts that this query is performed on to the specified threat ID values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

threats (*list*) – List of string threat ID values.

Returns

This instance.

Return type

AlertSearchQuery

set_threat_notes_present(*is_present*, *exclude=False*)

Restricts the alerts that this query is performed on to those with or without threat_notes.

Parameters

- **is_present** (*bool*) – If true, returns alerts that have a note attached to the threat_id
- **exclude** (*bool*) – If true, will set is_present in the exclusions. Otherwise adds to criteria

Returns

This instance.

Return type

AlertSearchQuery

set_time_range(**args*, ***kwargs*)

For v7 Alerts:

Sets the ‘time_range’ query body parameter, determining a time range based on ‘backend_timestamp’.

Parameters

- ***args** – not used
- ****kwargs** (*dict*) – Used to specify the period to search within
 - start= either timestamp ISO 8601 strings or datetime objects

- end= either timestamp ISO 8601 strings or datetime objects
 - range= the period on which to execute the result search, ending on the current time.
- Range must be in the format “-<quantity><units>” where quantity is an integer, and units is one of:
- M: month(s)
 - w: week(s)
 - d: day(s)
 - h: hour(s)
 - m: minute(s)
 - s: second(s)

For v6 Alerts (backwards compatibility):

Restricts the alerts that this query is performed on to the specified time range for a given key. Will set the ‘time_range’ as in the v7 usage if key is create_time and set a criteria value for any other valid key.

Parameters

- **key** (*str*) – The key to use for criteria one of create_time, first_event_time, last_event_time or last_update_time. i.e. legacy field names from the Alert v6 API.
- ****kwargs** (*dict*) – Used to specify the period to search within
 - start= either timestamp ISO 8601 strings or datetime objects
 - end= either timestamp ISO 8601 strings or datetime objects
 - range= the period on which to execute the result search, ending on the current time.

Returns

This instance.

Return type

AlertSearchQuery

Examples

```
>>> query_specify_start_and_end = api.select(Alert).
...     set_time_range(start="2020-10-20T20:34:07Z", end="2020-10-30T20:34:07Z")
>>> query_specify_range = api.select(Alert).set_time_range(range='-3d')
>>> query_legacy_use = api.select(Alert).set_time_range("create_time", range='-
↪3d')
```

set_types(alerttypes)

Restricts the alerts that this query is performed on to the specified alert type values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

alerttypes (*list*) – List of string alert type values. Valid values are “CB_ANALYTICS”, “WATCHLIST”, “DEVICE_CONTROL”, and “CONTAINER_RUNTIME”. In SDK 1.5.0,

to align with Alert API v7, more alert types are available but the *add_criteria* method must be used.

Returns

This instance.

Return type

AlertSearchQuery

Note: - When filtering by fields that take a list parameter, an empty list will be treated as a wildcard and match everything.

set_vendor_ids(ids)

Restricts the alerts that this query is performed on to the specified vendor IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

ids (*list*) – List of vendor IDs to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_vendor_names(names)

Restricts the alerts that this query is performed on to the specified vendor names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

names (*list*) – List of vendor names to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_watchlist_ids(ids)

Restricts the alerts that this query is performed on to the specified watchlist ID values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

ids (*list*) – List of string watchlist ID values.

Returns

This instance.

Return type

WatchlistAlertSearchQuery

set_watchlist_names(*names*)

Restricts the alerts that this query is performed on to the specified watchlist name values.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

names (*list*) – List of string watchlist name values.

Returns

This instance.

Return type

WatchlistAlertSearchQuery

set_workflows(*workflow_vals*)

Restricts the alerts that this query is performed on to the specified workflow status values.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

workflow_vals (*list*) – List of string alert type values. Valid values are “OPEN” and “DISMISSED”.

Returns

This instance.

Return type

AlertSearchQuery

set_workload_ids(*ids*)

The field `workload_id` was deprecated and not included in v7. This method has been removed.

Use `workload_name` instead. See [Developer Network Alerts v6 Migration](#) for more details.

Parameters

ids (*list*) – List of workload IDs to look for.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_workload_kinds(*kinds*)

Restricts the alerts that this query is performed on to the specified workload types.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

kinds (*list*) – List of workload types to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_workload_names(*names*)

Restricts the alerts that this query is performed on to the specified workload names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

names (*list*) – List of workload names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

sort_by(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Example

```
>>> cb.select(Alert).sort_by("name")
```

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

This instance.

Return type

AlertSearchQuery

update(*status*, *closure_reason*=None, *determination*=None, *note*=None)

Updating all alerts matching a grouped alert query is not implemented.

Note:

- Updating all alerts in all groups returned by a `GroupedAlertSearchQuery` can be done by getting the `AlertSearchQuery` and using `update()` on it as shown in the following example.
-

Example

```
>>> alert_query = grouped_alert_query.get_alert_search_query()
>>> job = alert_query.update("IN_PROGRESS", "NO_REASON", "NONE", "Starting_
↪Investigation")
>>> completed_job = job.await_completion().result()
```

update_criteria(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

update_exclusions(*key, newlist*)

Update the exclusion on this query with a custom exclusion key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*list*) – List of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).update_exclusions("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

```
class HostBasedFirewallAlert(cb, model_unique_id, initial_data=None)
```

Bases: *Alert*

Represents Host Based Firewall alerts.

Parameters

- **additional_events_present** – Indicator to let API and forwarder users know that they should look up other associated events related to this alert
- **alert_notes_present** – True if notes are present on the alert ID. False if notes are not present.
- **alert_url** – Link to the alerts page for this alert. Does not vary by alert type
- **backend_timestamp** – Timestamp when the Carbon Black Cloud processed and enabled the alert for searching. Corresponds to the Created column on the Alerts page.
- **backend_update_timestamp** – Timestamp when the Carbon Black Cloud initiated and processed an update to an alert. Corresponds to the Updated column on the Alerts page. Note that changes made by users do not change this date; those changes are reflected on *user_update_timestamp*
- **blocked_effective_reputation** – Effective reputation of the blocked file or process; applied by the sensor at the time the block occurred
- **blocked_md5** – MD5 hash of the child process binary; for any process terminated by the sensor
- **blocked_name** – Tokenized file path of the files blocked by sensor action
- **blocked_sha256** – SHA-256 hash of the child process binary; for any process terminated by the sensor
- **category** – Alert category - Monitored vs Threat
- **childproc_cmdline** – Command line for the child process
- **childproc_effective_reputation** – Effective reputation of the child process; applied by the sensor at the time the event occurred
- **childproc_guid** – Unique process identifier assigned to the child process
- **childproc_md5** – Hash of the child process' binary (Enterprise EDR)
- **childproc_name** – Filesystem path of the child process' binary
- **childproc_sha256** – Hash of the child process' binary (Endpoint Standard)
- **childproc_username** – User context in which the child process was executed
- **detection_timestamp** – For sensor-sent alerts, this is the time of the event on the sensor. For alerts generated on the backend, this is the time the backend system triggered the alert.
- **determination** – User-updatable determination of the alert
- **device_external_ip** – IP address of the endpoint according to the Carbon Black Cloud; can differ from *device_internal_ip* due to network proxy or NAT; either IPv4 (dotted decimal notation) or IPv6 (proprietary format)
- **device_id** – ID of devices
- **device_internal_ip** – IP address of the endpoint reported by the sensor; either IPv4 (dotted decimal notation) or IPv6 (proprietary format)
- **device_location** – Whether the device was on or off premises when the alert started, based on the current IP address and the device's registered DNS domain suffix
- **device_name** – Device name
- **device_os** – Device Operating Systems

- **device_os_version** – The operating system and version of the endpoint. Requires Windows CBC sensor version 3.5 or later.
- **device_policy** – Device policy
- **device_policy_id** – Device policy id
- **device_target_value** – Target value assigned to the device, set from the policy
- **device_uem_id** – Device correlation with WS1/EUC, required for our Workspace ONE Intelligence integration to function
- **device_username** – Logged on user during the alert. This is filled on a best-effort approach. If the user is not available it may be populated with the device owner (empty for Container Runtime alerts)
- **first_event_timestamp** – Timestamp when the first event in the alert occurred
- **id** – Unique IDs of alerts
- **is_updated** – Boolean that describes whether or not this is the original copy of the alert
- **last_event_timestamp** – Timestamp when the last event in the alert occurred
- **netconn_local_ip** – IP address of the remote side of the network connection; stored as dotted decimal
- **netconn_local_ipv4** – IPv4 address of the local side of the network connection; stored as a dotted decimal. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_local_ipv6** – IPv6 address of the local side of the network connection; stored as a string without octet-separating colon characters. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_local_port** – TCP or UDP port used by the local side of the network connection
- **netconn_protocol** – Network protocol of the network connection
- **netconn_remote_domain** – Domain name (FQDN) associated with the remote end of the network connection, if available
- **netconn_remote_ip** – IP address of the local side of the network connection; stored as dotted decimal
- **netconn_remote_ipv4** – IPv4 address of the remote side of the network connection; stored as dotted decimal. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_remote_ipv6** – IPv6 address of the remote side of the network connection; stored as a string without octet-separating colon characters. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_remote_port** – TCP or UDP port used by the remote side of the network connection; same as netconn_port and event_network_remote_port
- **org_key** – Unique alphanumeric string that identifies your organization in the Carbon Black Cloud
- **parent_cmdline** – Command line of the parent process
- **parent_effective_reputation** – Effective reputation of the parent process; applied by the sensor when the event occurred
- **parent_guid** – Unique process identifier assigned to the parent process
- **parent_md5** – MD5 hash of the parent process binary

- **parent_name** – Filesystem path of the parent process binary
- **parent_pid** – Identifier assigned by the operating system to the parent process
- **parent_reputation** – Reputation of the parent process; applied by the Carbon Black Cloud when the event is initially processed
- **parent_sha256** – SHA-256 hash of the parent process binary
- **parent_username** – User context in which the parent process was executed
- **policy_applied** – Indicates whether or not a policy has been applied to any event associated with this alert
- **primary_event_id** – ID of the primary event in the alert
- **process_cmdline** – Command line executed by the actor process
- **process_effective_reputation** – Effective reputation of the actor hash
- **process_guid** – Guid of the process that has fired the alert (optional)
- **process_md5** – MD5 hash of the actor process binary
- **process_name** – Process names of an alert
- **process_pid** – PID of the process that has fired the alert (optional)
- **process_reputation** – Reputation of the actor process; applied when event is processed by the Carbon Black Cloud
- **process_sha256** – SHA-256 hash of the actor process binary
- **process_username** – User context in which the actor process was executed. MacOS - all users for the PID for fork() and exec() transitions. Linux - process user for exec() events, but in a future sensor release can be multi-valued due to setuid().
- **reason** – A spoken language written explanation of the what and why the alert occurred and any action taken, usually consisting of 1 to 3 sentences.
- **reason_code** – A unique short-hand code or GUID identifying the particular alert reason
- **rule_category_id** – ID representing the category of the rule_id for certain alert types
- **rule_id** – ID of the rule that triggered an alert; applies to Intrusion Detection System, Host-Based Firewall, TAU Intelligence, and USB Device Control alerts
- **run_state** – Whether the threat in the alert actually ran
- **sensor_action** – Actions taken by the sensor, according to the rules of a policy
- **severity** – integer representation of the impact of alert if true positive
- **tags** – Tags added to the threat ID of the alert
- **threat_id** – ID assigned to a group of alerts with common criteria, based on alert type
- **threat_notes_present** – True if notes are present on the threat ID. False if notes are not present.
- **type** – Type of alert generated
- **user_update_timestamp** – Timestamp of the last property of an alert changed by a user, such as the alert workflow or determination
- **workflow** – Current workflow state of an alert. The workflow represents the flow from *OPEN* to *IN_PROGRESS* to *CLOSED* and captures who moved the alert into the current state. The history of these state transitions is available via the alert history route.

Initialize the Alert object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

class Note(*cb, alert, model_unique_id, threat_note=False, initial_data=None*)

Bases: [PlatformModel](#)

Represents a note within an alert.

Parameters

- **author** – User who created the note
- **create_timestamp** – Time the note was created
- **last_update_timestamp** – Time the note was created
- **id** – Unique ID for this note
- **note** – Note contents
- **parent_id** – ID for this note of this notes parent if is a thread

Initialize the Note object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **alert** ([Alert](#)) – The alert where the note is saved.
- **model_unique_id** (*str*) – ID of the note represented.
- **threat_note** (*bool*) – Whether the note is an Alert or Threat note
- **initial_data** (*dict*) – Initial data used to populate the note.

delete()

Deletes a note from an alert.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

add_threat_tags(*tags*)

Adds tags to the threat

Required Permissions:

org.alerts.tags (CREATE)

Parameters

tags (*list[str]*) – List of tags to add to the threat

Raises

ApiError – If tags is not a list of strings

Returns

The list of current tags

Return type

(list[str])

close(*closure_reason=None, determination=None, note=None*)

Closes this alert.

Parameters

- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.close("RESOLVED", "FALSE_POSITIVE", "Normal behavior")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type

Job

create_note(*note, threat_note=False*)

Creates a new note.

Parameters

- **note** (*str*) – Note content to add
- **threat_note** (*bool*) – Whether to add the note to the Alert or Threat

delete_threat_tag(*tag*)

Delete a threat tag

Required Permissions:

org.alerts.tags (DELETE)

Parameters

tag (*str*) – The tag to delete

Returns

The list of current tags

Return type

(list[*str*])

deobfuscate_cmdline()

Deobfuscates the command line of the process pointed to by the alert and returns the deobfuscated result.

Required Permissions:

script.deobfuscation(EXECUTE)

Returns

A dict containing information about the obfuscated command line, including the deobfuscated result.

Return type

dict

dismiss_threat(*remediation=None, comment=None*)

Dismisses all future alerts assigned to the threat_id.

Parameters

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to dismiss all past and current open alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).close().  
↪ ..)
```

get(*item, default_val=None*)

Return an attribute of this object.

Parameters

- **item** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_history(*threat=False*)

Get the actions taken on an Alert such as Notes added and workflow state changes.

Parameters

threat (*bool*) – Whether to return the Alert or Threat history

Returns

The dicts of each determination, note or workflow change

Return type

list

get_observations(*timeout=0*)

Requests observations that are associated with the Alert.

Uses Observations bulk get details.

Returns

Observations associated with the alert

Return type

list

Note:

- When using asynchronous mode, this method returns a python future. You can call result() on the future object to wait for completion and get the results.
-

get_process(*async_mode=False*)

Gets the process corresponding with the alert.

Parameters

async_mode – True to request process in an asynchronous manner.

Returns

The process corresponding to the alert.

Return type

Process

get_threat_tags()

Gets the threat's tags

Required Permissions:

org.alerts.tags (READ)

Returns

The list of current tags

Return type

(list[str])

notes_(*threat_note=False*)

Retrieves all notes for an alert.

Parameters

threat_note (*bool*) – Whether to return the Alert notes or Threat notes

refresh()

Reload this object from the server.

static search_suggestions(*cb, query*)

Returns suggestions for keys and field values that can be used in a search.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **query** (*str*) – A search query to use.

Returns

A list of search suggestions expressed as dict objects.

Return type

list

Raises

[ApiError](#) – if cb is not instance of CBCloudAPI

to_json(*version='v7'*)

Return a json object of the response.

Parameters

version (*str*) – version of json to return. Either v6 or v7. DEFAULT v7

Returns

The returned attribute value.

Return type

Any

update(*status, closure_reason=None, determination=None, note=None*)

Update the Alert with optional closure_reason, determination, note, or status.

Parameters

- **status** (*str*) – The status to set for this alert, either “OPEN”, “IN_PROGRESS”, or “CLOSED”.
- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.update("IN_PROGESS", "NO_REASON", "NONE", "Starting_
↳Investigation")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type

Job

update_threat(*remediation=None, comment=None*)

Updates all future alerts assigned to the threat_id to the OPEN state.

Parameters

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to update all past and current alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).
↳update(...)
```

property workflow_

Returns the workflow associated with this alert.

Returns

The workflow associated with this alert.

Return type

dict

class IntrusionDetectionSystemAlert(*cb, model_unique_id, initial_data=None*)

Bases: *Alert*

Represents Intrusion Detection System alerts.

Parameters

- **additional_events_present** – Indicator to let API and forwarder users know that they should look up other associated events related to this alert
- **alert_notes_present** – True if notes are present on the alert ID. False if notes are not present.
- **alert_url** – Link to the alerts page for this alert. Does not vary by alert type
- **attack_tactic** – A tactic from the MITRE ATT&CK framework; defines a reason for an adversary's action, such as achieving credential access

- **attack_technique** – A technique from the MITRE ATT&CK framework; defines an action an adversary takes to accomplish a goal, such as dumping credentials to achieve credential access
- **backend_timestamp** – Timestamp when the Carbon Black Cloud processed and enabled the alert for searching. Corresponds to the Created column on the Alerts page.
- **backend_update_timestamp** – Timestamp when the Carbon Black Cloud initiated and processed an update to an alert. Corresponds to the Updated column on the Alerts page. Note that changes made by users do not change this date; those changes are reflected on *user_update_timestamp*
- **blocked_effective_reputation** – Effective reputation of the blocked file or process; applied by the sensor at the time the block occurred
- **blocked_md5** – MD5 hash of the child process binary; for any process terminated by the sensor
- **blocked_name** – Tokenized file path of the files blocked by sensor action
- **blocked_sha256** – SHA-256 hash of the child process binary; for any process terminated by the sensor
- **category** – Alert category - Monitored vs Threat
- **childproc_cmdline** – Command line for the child process
- **childproc_effective_reputation** – Effective reputation of the child process; applied by the sensor at the time the event occurred
- **childproc_guid** – Unique process identifier assigned to the child process
- **childproc_md5** – Hash of the child process' binary (Enterprise EDR)
- **childproc_name** – Filesystem path of the child process' binary
- **childproc_sha256** – Hash of the child process' binary (Endpoint Standard)
- **childproc_username** – User context in which the child process was executed
- **detection_timestamp** – For sensor-sent alerts, this is the time of the event on the sensor. For alerts generated on the backend, this is the time the backend system triggered the alert.
- **determination** – User-updatable determination of the alert
- **device_external_ip** – IP address of the endpoint according to the Carbon Black Cloud; can differ from *device_internal_ip* due to network proxy or NAT; either IPv4 (dotted decimal notation) or IPv6 (proprietary format)
- **device_id** – ID of devices
- **device_internal_ip** – IP address of the endpoint reported by the sensor; either IPv4 (dotted decimal notation) or IPv6 (proprietary format)
- **device_location** – Whether the device was on or off premises when the alert started, based on the current IP address and the device's registered DNS domain suffix
- **device_name** – Device name
- **device_os** – Device Operating Systems
- **device_os_version** – The operating system and version of the endpoint. Requires Windows CBC sensor version 3.5 or later.
- **device_policy** – Device policy

- **device_policy_id** – Device policy id
- **device_target_value** – Target value assigned to the device, set from the policy
- **device_uem_id** – Device correlation with WS1/EUC, required for our Workspace ONE Intelligence integration to function
- **device_username** – Logged on user during the alert. This is filled on a best-effort approach. If the user is not available it may be populated with the device owner (empty for Container Runtime alerts)
- **first_event_timestamp** – Timestamp when the first event in the alert occurred
- **id** – Unique IDs of alerts
- **is_updated** – Boolean that describes whether or not this is the original copy of the alert
- **last_event_timestamp** – Timestamp when the last event in the alert occurred
- **netconn_local_ip** – IP address of the remote side of the network connection; stored as dotted decimal
- **netconn_local_ipv4** – IPv4 address of the local side of the network connection; stored as a dotted decimal. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_local_ipv6** – IPv6 address of the local side of the network connection; stored as a string without octet-separating colon characters. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_local_port** – TCP or UDP port used by the local side of the network connection
- **netconn_protocol** – Network protocol of the network connection
- **netconn_remote_domain** – Domain name (FQDN) associated with the remote end of the network connection, if available
- **netconn_remote_ip** – IP address of the local side of the network connection; stored as dotted decimal
- **netconn_remote_ipv4** – IPv4 address of the remote side of the network connection; stored as dotted decimal. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_remote_ipv6** – IPv6 address of the remote side of the network connection; stored as a string without octet-separating colon characters. Only one of ipv4 and ipv6 fields will be populated.
- **netconn_remote_port** – TCP or UDP port used by the remote side of the network connection; same as netconn_port and event_network_remote_port
- **org_key** – Unique alphanumeric string that identifies your organization in the Carbon Black Cloud
- **parent_cmdline** – Command line of the parent process
- **parent_effective_reputation** – Effective reputation of the parent process; applied by the sensor when the event occurred
- **parent_guid** – Unique process identifier assigned to the parent process
- **parent_md5** – MD5 hash of the parent process binary
- **parent_name** – Filesystem path of the parent process binary
- **parent_pid** – Identifier assigned by the operating system to the parent process

- **parent_reputation** – Reputation of the parent process; applied by the Carbon Black Cloud when the event is initially processed
- **parent_sha256** – SHA-256 hash of the parent process binary
- **parent_username** – User context in which the parent process was executed
- **policy_applied** – Indicates whether or not a policy has been applied to any event associated with this alert
- **primary_event_id** – ID of the primary event in the alert
- **process_cmdline** – Command line executed by the actor process
- **process_effective_reputation** – Effective reputation of the actor hash
- **process_guid** – Guid of the process that has fired the alert (optional)
- **process_md5** – MD5 hash of the actor process binary
- **process_name** – Process names of an alert
- **process_pid** – PID of the process that has fired the alert (optional)
- **process_reputation** – Reputation of the actor process; applied when event is processed by the Carbon Black Cloud
- **process_sha256** – SHA-256 hash of the actor process binary
- **process_username** – User context in which the actor process was executed. MacOS - all users for the PID for fork() and exec() transitions. Linux - process user for exec() events, but in a future sensor release can be multi-valued due to setuid().
- **reason** – A spoken language written explanation of the what and why the alert occurred and any action taken, usually consisting of 1 to 3 sentences.
- **reason_code** – A unique short-hand code or GUID identifying the particular alert reason
- **rule_category_id** – ID representing the category of the rule_id for certain alert types
- **rule_id** – ID of the rule that triggered an alert; applies to Intrusion Detection System, Host-Based Firewall, TAU Intelligence, and USB Device Control alerts
- **run_state** – Whether the threat in the alert actually ran
- **sensor_action** – Actions taken by the sensor, according to the rules of a policy
- **severity** – integer representation of the impact of alert if true positive
- **tags** – Tags added to the threat ID of the alert
- **threat_id** – ID assigned to a group of alerts with common criteria, based on alert type
- **threat_name** – Name of the threat
- **threat_notes_present** – True if notes are present on the threat ID. False if notes are not present.
- **tms_rule_id** – Detection id
- **ttps** – Other potential malicious activities involved in a threat
- **type** – Type of alert generated
- **user_update_timestamp** – Timestamp of the last property of an alert changed by a user, such as the alert workflow or determination

- **workflow** – Current workflow state of an alert. The workflow represents the flow from *OPEN* to *IN_PROGRESS* to *CLOSED* and captures who moved the alert into the current state. The history of these state transitions is available via the alert history route.

Initialize the Alert object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

class Note(*cb, alert, model_unique_id, threat_note=False, initial_data=None*)

Bases: [PlatformModel](#)

Represents a note within an alert.

Parameters

- **author** – User who created the note
- **create_timestamp** – Time the note was created
- **last_update_timestamp** – Time the note was created
- **id** – Unique ID for this note
- **note** – Note contents
- **parent_id** – ID for this note of this notes parent if is a thread

Initialize the Note object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **alert** ([Alert](#)) – The alert where the note is saved.
- **model_unique_id** (*str*) – ID of the note represented.
- **threat_note** (*bool*) – Whether the note is an Alert or Threat note
- **initial_data** (*dict*) – Initial data used to populate the note.

delete()

Deletes a note from an alert.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

add_threat_tags(tags)

Adds tags to the threat

Required Permissions:

org.alerts.tags (CREATE)

Parameters

tags (*list[str]*) – List of tags to add to the threat

Raises

ApiError – If tags is not a list of strings

Returns

The list of current tags

Return type

(list[str])

close(closure_reason=None, determination=None, note=None)

Closes this alert.

Parameters

- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.close("RESOLVED", "FALSE_POSITIVE", "Normal behavior")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type*Job***create_note**(*note*, *threat_note=False*)

Creates a new note.

Parameters

- **note** (*str*) – Note content to add
- **threat_note** (*bool*) – Whether to add the note to the Alert or Threat

delete_threat_tag(*tag*)

Delete a threat tag

Required Permissions:

org.alerts.tags (DELETE)

Parameters**tag** (*str*) – The tag to delete**Returns**

The list of current tags

Return type(list[*str*])**deobfuscate_cmdline**()

Deobfuscates the command line of the process pointed to by the alert and returns the deobfuscated result.

Required Permissions:

script.deobfuscation(EXECUTE)

Returns

A dict containing information about the obfuscated command line, including the deobfuscated result.

Return type

dict

dismiss_threat(*remediation=None*, *comment=None*)

Dismisses all future alerts assigned to the threat_id.

Parameters

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to dismiss all past and current open alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).close(
    ← ...)
```


get(*item*, *default_val=None*)

Return an attribute of this object.

Parameters

- **item** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_history(*threat=False*)

Get the actions taken on an Alert such as Notes added and workflow state changes.

Parameters

threat (*bool*) – Whether to return the Alert or Threat history

Returns

The dicts of each determination, note or workflow change

Return type

list

get_network_threat_metadata()

The NetworkThreatMetadata associated with this IDS alert if it exists.

Example

```
>>> alert_threat_metadata = ids_alert.get_network_threat_metadata()
```

Returns

The NetworkThreatMetadata associated with this IDS alert.

Return type

NetworkThreatMetadata

get_observations(*timeout=0*)

Requests observations that are associated with the Alert.

Uses Observations bulk get details.

Returns

Observations associated with the alert

Return type

list

Note:

- When using asynchronous mode, this method returns a python future. You can call `result()` on the future object to wait for completion and get the results.

get_process(*async_mode=False*)

Gets the process corresponding with the alert.

Parameters

async_mode – True to request process in an asynchronous manner.

Returns

The process corresponding to the alert.

Return type

Process

get_threat_tags()

Gets the threat's tags

Required Permissions:

org.alerts.tags (READ)

Returns

The list of current tags

Return type

(list[str])

notes_(*threat_note=False*)

Retrieves all notes for an alert.

Parameters

threat_note (*bool*) – Whether to return the Alert notes or Threat notes

refresh()

Reload this object from the server.

static search_suggestions(*cb, query*)

Returns suggestions for keys and field values that can be used in a search.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.
- **query** (*str*) – A search query to use.

Returns

A list of search suggestions expressed as dict objects.

Return type

list

Raises

ApiError – if cb is not instance of CBCloudAPI

to_json(*version='v7'*)

Return a json object of the response.

Parameters

version (*str*) – version of json to return. Either v6 or v7. DEFAULT v7

Returns

The returned attribute value.

Return type

Any

update(*status*, *closure_reason*=None, *determination*=None, *note*=None)Update the Alert with optional *closure_reason*, *determination*, *note*, or *status*.**Parameters**

- **status** (*str*) – The status to set for this alert, either “OPEN”, “IN_PROGRESS”, or “CLOSED”.
- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.update("IN_PROGESS", "NO_REASON", "NONE", "Starting_
↳Investigation")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type*Job***update_threat**(*remediation*=None, *comment*=None)Updates all future alerts assigned to the *threat_id* to the OPEN state.**Parameters**

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to update all past and current alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).
↳update(...)
```

property workflow_

Returns the workflow associated with this alert.

Returns

The workflow associated with this alert.

Return type

dict

class WatchlistAlert(*cb, model_unique_id, initial_data=None*)

Bases: [*Alert*](#)

Represents watch list alerts.

Parameters

- **additional_events_present** – Indicator to let API and forwarder users know that they should look up other associated events related to this alert
- **alert_notes_present** – True if notes are present on the alert ID. False if notes are not present.
- **alert_url** – Link to the alerts page for this alert. Does not vary by alert type
- **attack_tactic** – A tactic from the MITRE ATT&CK framework; defines a reason for an adversary’s action, such as achieving credential access
- **attack_technique** – A technique from the MITRE ATT&CK framework; defines an action an adversary takes to accomplish a goal, such as dumping credentials to achieve credential access
- **backend_timestamp** – Timestamp when the Carbon Black Cloud processed and enabled the alert for searching. Corresponds to the Created column on the Alerts page.
- **backend_update_timestamp** – Timestamp when the Carbon Black Cloud initiated and processed an update to an alert. Corresponds to the Updated column on the Alerts page. Note that changes made by users do not change this date; those changes are reflected on *user_update_timestamp*
- **blocked_effective_reputation** – Effective reputation of the blocked file or process; applied by the sensor at the time the block occurred
- **blocked_md5** – MD5 hash of the child process binary; for any process terminated by the sensor
- **blocked_name** – Tokenized file path of the files blocked by sensor action
- **blocked_sha256** – SHA-256 hash of the child process binary; for any process terminated by the sensor
- **category** – Alert category - Monitored vs Threat
- **childproc_cmdline** – Command line for the child process
- **childproc_effective_reputation** – Effective reputation of the child process; applied by the sensor at the time the event occurred
- **childproc_guid** – Unique process identifier assigned to the child process
- **childproc_md5** – Hash of the child process’ binary (Enterprise EDR)
- **childproc_name** – Filesystem path of the child process’ binary
- **childproc_sha256** – Hash of the child process’ binary (Endpoint Standard)

- **childproc_username** – User context in which the child process was executed
- **detection_timestamp** – For sensor-sent alerts, this is the time of the event on the sensor. For alerts generated on the backend, this is the time the backend system triggered the alert.
- **determination** – User-updatable determination of the alert
- **device_external_ip** – IP address of the endpoint according to the Carbon Black Cloud; can differ from `device_internal_ip` due to network proxy or NAT; either IPv4 (dotted decimal notation) or IPv6 (proprietary format)
- **device_id** – ID of devices
- **device_internal_ip** – IP address of the endpoint reported by the sensor; either IPv4 (dotted decimal notation) or IPv6 (proprietary format)
- **device_location** – Whether the device was on or off premises when the alert started, based on the current IP address and the device’s registered DNS domain suffix
- **device_name** – Device name
- **device_os** – Device Operating Systems
- **device_os_version** – The operating system and version of the endpoint. Requires Windows CBC sensor version 3.5 or later.
- **device_policy** – Device policy
- **device_policy_id** – Device policy id
- **device_target_value** – Target value assigned to the device, set from the policy
- **device_uem_id** – Device correlation with WS1/EUC, required for our Workspace ONE Intelligence integration to function
- **device_username** – Logged on user during the alert. This is filled on a best-effort approach. If the user is not available it may be populated with the device owner (empty for Container Runtime alerts)
- **first_event_timestamp** – Timestamp when the first event in the alert occurred
- **id** – Unique IDs of alerts
- **ioc_field** – The field the indicator of comprise (IOC) hit contains
- **ioc_hit** – IOC field value or IOC query that matches
- **ioc_id** – Unique identifier of the IOC that generated the watchlist hit
- **is_updated** – Boolean that describes whether or not this is the original copy of the alert
- **last_event_timestamp** – Timestamp when the last event in the alert occurred
- **ml_classification_final_verdict** – Final verdict of the alert, based on the ML models that were used to make the prediction.
- **ml_classification_global_prevalence** – Categories (low/medium/high) used to describe the prevalence of alerts across all regional organizations.
- **ml_classification_org_prevalence** – Categories (low/medium/high) used to describe the prevalence of alerts within an organization.
- **netconn_local_ip** – IP address of the remote side of the network connection; stored as dotted decimal
- **netconn_local_ipv4** – IPv4 address of the local side of the network connection; stored as a dotted decimal. Only one of `ipv4` and `ipv6` fields will be populated.

- **netconn_local_ipv6** – IPv6 address of the local side of the network connection; stored as a string without octet-separating colon characters. Only one of `ipv4` and `ipv6` fields will be populated.
- **netconn_local_port** – TCP or UDP port used by the local side of the network connection
- **netconn_protocol** – Network protocol of the network connection
- **netconn_remote_domain** – Domain name (FQDN) associated with the remote end of the network connection, if available
- **netconn_remote_ip** – IP address of the local side of the network connection; stored as dotted decimal
- **netconn_remote_ipv4** – IPv4 address of the remote side of the network connection; stored as dotted decimal. Only one of `ipv4` and `ipv6` fields will be populated.
- **netconn_remote_ipv6** – IPv6 address of the remote side of the network connection; stored as a string without octet-separating colon characters. Only one of `ipv4` and `ipv6` fields will be populated.
- **netconn_remote_port** – TCP or UDP port used by the remote side of the network connection; same as `netconn_port` and `event_network_remote_port`
- **org_key** – Unique alphanumeric string that identifies your organization in the Carbon Black Cloud
- **parent_cmdline** – Command line of the parent process
- **parent_effective_reputation** – Effective reputation of the parent process; applied by the sensor when the event occurred
- **parent_guid** – Unique process identifier assigned to the parent process
- **parent_md5** – MD5 hash of the parent process binary
- **parent_name** – Filesystem path of the parent process binary
- **parent_pid** – Identifier assigned by the operating system to the parent process
- **parent_reputation** – Reputation of the parent process; applied by the Carbon Black Cloud when the event is initially processed
- **parent_sha256** – SHA-256 hash of the parent process binary
- **parent_username** – User context in which the parent process was executed
- **policy_applied** – Indicates whether or not a policy has been applied to any event associated with this alert
- **primary_event_id** – ID of the primary event in the alert
- **process_cmdline** – Command line executed by the actor process
- **process_effective_reputation** – Effective reputation of the actor hash
- **process_guid** – Guid of the process that has fired the alert (optional)
- **process_md5** – MD5 hash of the actor process binary
- **process_name** – Process names of an alert
- **process_pid** – PID of the process that has fired the alert (optional)
- **process_reputation** – Reputation of the actor process; applied when event is processed by the Carbon Black Cloud

- **process_sha256** – SHA-256 hash of the actor process binary
- **process_username** – User context in which the actor process was executed. MacOS - all users for the PID for fork() and exec() transitions. Linux - process user for exec() events, but in a future sensor release can be multi-valued due to setuid().
- **reason** – A spoken language written explanation of the what and why the alert occurred and any action taken, usually consisting of 1 to 3 sentences.
- **reason_code** – A unique short-hand code or GUID identifying the particular alert reason
- **report_description** – Description of the watchlist report associated with the alert
- **report_id** – Report IDs that contained the IOC that caused a hit
- **report_link** – Link of reports that contained the IOC that caused a hit
- **report_name** – Name of the watchlist report
- **report_tags** – Tags associated with the watchlist report
- **run_state** – Whether the threat in the alert actually ran
- **sensor_action** – Actions taken by the sensor, according to the rules of a policy
- **severity** – integer representation of the impact of alert if true positive
- **tags** – Tags added to the threat ID of the alert
- **threat_id** – ID assigned to a group of alerts with common criteria, based on alert type
- **threat_notes_present** – True if notes are present on the threat ID. False if notes are not present.
- **ttps** – Other potential malicious activities involved in a threat
- **type** – Type of alert generated
- **user_update_timestamp** – Timestamp of the last property of an alert changed by a user, such as the alert workflow or determination
- **watchlists** – List of watchlists associated with an alert. Alerts are batched hourly
- **workflow** – Current workflow state of an alert. The workflow represents the flow from *OPEN* to *IN_PROGRESS* to *CLOSED* and captures who moved the alert into the current state. The history of these state transitions is available via the alert history route.

Initialize the Alert object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

class Note(*cb, alert, model_unique_id, threat_note=False, initial_data=None*)

Bases: [PlatformModel](#)

Represents a note within an alert.

Parameters

- **author** – User who created the note
- **create_timestamp** – Time the note was created
- **last_update_timestamp** – Time the note was created

- **id** – Unique ID for this note
- **note** – Note contents
- **parent_id** – ID for this note of this notes parent if is a thread

Initialize the Note object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **alert** ([Alert](#)) – The alert where the note is saved.
- **model_unique_id** (*str*) – ID of the note represented.
- **threat_note** (*bool*) – Whether the note is an Alert or Threat note
- **initial_data** (*dict*) – Initial data used to populate the note.

delete()

Deletes a note from an alert.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

add_threat_tags(*tags*)

Adds tags to the threat

Required Permissions:

org.alerts.tags (CREATE)

Parameters

tags (*list[str]*) – List of tags to add to the threat

Raises

[ApiError](#) – If tags is not a list of strings

Returns

The list of current tags

Return type

(*list[str]*)

close(*closure_reason=None, determination=None, note=None*)

Closes this alert.

Parameters

- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.close("RESOLVED", "FALSE_POSITIVE", "Normal behavior")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type

Job

create_note(*note, threat_note=False*)

Creates a new note.

Parameters

- **note** (*str*) – Note content to add
- **threat_note** (*bool*) – Whether to add the note to the Alert or Threat

delete_threat_tag(*tag*)

Delete a threat tag

Required Permissions:

org.alerts.tags (DELETE)

Parameters

tag (*str*) – The tag to delete

Returns

The list of current tags

Return type

(list[str])

deobfuscate_cmdline()

Deobfuscates the command line of the process pointed to by the alert and returns the deobfuscated result.

Required Permissions:

script.deobfuscation(EXECUTE)

Returns

A dict containing information about the obfuscated command line, including the deobfuscated result.

Return type

dict

dismiss_threat(*remediation=None, comment=None*)

Dismisses all future alerts assigned to the threat_id.

Parameters

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to dismiss all past and current open alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).close(
    ↪ ..)
```

get(*item, default_val=None*)

Return an attribute of this object.

Parameters

- **item** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_history(*threat=False*)

Get the actions taken on an Alert such as Notes added and workflow state changes.

Parameters

threat (*bool*) – Whether to return the Alert or Threat history

Returns

The dicts of each determination, note or workflow change

Return type

list

get_observations(*timeout=0*)

Requests observations that are associated with the Alert.

Uses Observations bulk get details.

Returns

Observations associated with the alert

Return type

list

Note:

- When using asynchronous mode, this method returns a python future. You can call `result()` on the future object to wait for completion and get the results.
-

get_process(*async_mode=False*)

Gets the process corresponding with the alert.

Parameters

async_mode – True to request process in an asynchronous manner.

Returns

The process corresponding to the alert.

Return type

Process

get_threat_tags()

Gets the threat's tags

Required Permissions:

org.alerts.tags (READ)

Returns

The list of current tags

Return type

(list[str])

get_watchlist_objects()

Returns the list of associated watchlist objects for the associated watchlist alert.

Example

```
>>> watchlist_alert = cb.select(Alert, "f643d11f-59ab-478f-92c3-4198ca9b8230")
>>> watchlist_objects = watchlist_alert.get_watchlist_objects()
```

Returns

A list of Watchlist objects.

Return type

list[*Watchlist*]

notes_(*threat_note=False*)

Retrieves all notes for an alert.

Parameters

threat_note (*bool*) – Whether to return the Alert notes or Threat notes

refresh()

Reload this object from the server.

static search_suggestions(*cb, query*)

Returns suggestions for keys and field values that can be used in a search.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **query** (*str*) – A search query to use.

Returns

A list of search suggestions expressed as dict objects.

Return type

list

Raises

[ApiError](#) – if cb is not instance of CBCloudAPI

to_json(*version='v7'*)

Return a json object of the response.

Parameters

version (*str*) – version of json to return. Either v6 or v7. DEFAULT v7

Returns

The returned attribute value.

Return type

Any

update(*status, closure_reason=None, determination=None, note=None*)

Update the Alert with optional closure_reason, determination, note, or status.

Parameters

- **status** (*str*) – The status to set for this alert, either “OPEN”, “IN_PROGRESS”, or “CLOSED”.
- **closure_reason** (*str*) – the closure reason for this alert, either “NO_REASON”, “RESOLVED”, “RESOLVED_BENIGN_KNOWN_GOOD”, “DUPLICATE_CLEANUP”, “OTHER”
- **determination** (*str*) – The determination status to set for the alert, either “TRUE_POSITIVE”, “FALSE_POSITIVE”, or “NONE”
- **note** (*str*) – The comment to set for the alert.

Note:

- This is an asynchronous call that returns a Job. If you want to wait and block on the results you can call `await_completion()` to get a Future then `result()` on the future object to wait for completion and get the results.
-

Example

```
>>> alert = cb.select(Alert, "708d7dbf-2020-42d4-9cbc-0cddd0ffa31a")
>>> job = alert.update("IN_PROGRESS", "NO_REASON", "NONE", "Starting_
↳Investigation")
>>> completed_job = job.await_completion().result()
>>> alert.refresh()
```

Returns

The Job object for the alert workflow action.

Return type

Job

update_threat(*remediation=None, comment=None*)

Updates all future alerts assigned to the threat_id to the OPEN state.

Parameters

- **remediation** (*str*) – The remediation status to set for the alert.
- **comment** (*str*) – The comment to set for the alert.

Note:

- If you want to update all past and current alerts associated to the threat use the following:

```
>>> cb.select(Alert).add_criteria("threat_id", [alert.threat_id]).
↳update(...)
```

property workflow_

Returns the workflow associated with this alert.

Returns

The workflow associated with this alert.

Return type

dict

4.11.4 Asset Groups Module

The model and query classes for referencing asset groups.

An *asset group* represents a group of devices (endpoints, VM workloads, and/or VDIs) that can have a single policy applied to it so the protections of all similar assets are synchronized with one another. Policies carry a “position” value as one of their attributes, so that, between the policy attached directly to the device, and the policies attached to any asset groups the device is a member of, the one with the highest “position” is the one that applies to that device. Devices may be added to an asset group either explicitly, or implicitly by specifying a query on the asset group, such that all devices matching that search criteria are considered part of the asset group.

Typical usage example:

```
# assume "cb" is an instance of CBCloudAPI
query = cb.select(AssetGroup).where('name:"HQ Devices"')
group = query.first()
```

class `AssetGroup`(*cb*, *model_unique_id=None*, *initial_data=None*, *force_init=False*, *full_doc=False*)

Bases: [`MutableBaseModel`](#)

Represents an asset group within the current organization in the Carbon Black Cloud.

AssetGroup objects are typically located via a search (using `AssetGroupQuery`) before they can be operated

on. They may also be created on the Carbon Black Cloud by using the `create_group()` class method.

Parameters

- **id** – The asset group identifier.
- **name** – The asset group name.
- **description** – The asset group description.
- **org_key** – The organization key of the owning organization.
- **status** – Status of the group.
- **member_type** – The type of objects this asset group contains.
- **discovered** – Whether this group has been discovered.
- **create_time** – Date and time the group was created.
- **update_time** – Date and time the group was last updated.
- **member_count** – Number of members in this group.
- **policy_id** – ID of the policy associated with this group.
- **policy_name** – Name of the policy associated with this group.
- **query** – Search query used to determine which assets are included in the group membership.

Initialize the `AssetGroup` object.

Required Permissions:

group-management(READ)

Parameters

- **cb** ([`BaseAPI`](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*int*) – ID of the policy.
- **initial_data** (*dict*) – Initial data used to populate the policy.
- **force_init** (*bool*) – If True, forces the object to be refreshed after constructing. Default False.
- **full_doc** (*bool*) – If True, object is considered “fully” initialized. Default False.

add_members(*members*)

Adds additional members to this asset group.

Required Permissions:

group-management(CREATE)

Parameters

members (*int*, [`Device`](#), or *list*) – The members to be added to the group. This may be an integer device ID, a `Device` object, or a list of either integers or `Device` objects.

classmethod `create_group(cb, name, description=None, policy_id=None, query=None)`

Create a new asset group.

Required Permissions:

group-management(CREATE)

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **name** (*str*) – Name for the new asset group.
- **description** (*str*) – Description for the new asset group. Default is None.
- **policy_id** (*int*) – ID of the policy to be associated with this asset group. Default is None.
- **query** (*str*) – Query string to be used to dynamically populate this group. Default is None, which means devices `_must_` be manually assigned to the group.

Returns

The new asset group.

Return type

[AssetGroup](#)

delete()

Delete this object.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

classmethod `get_all_groups(cb)`

Retrieve all asset groups in the organization.

Required Permissions:

group-management(READ)

Parameters

cb ([BaseAPI](#)) – Reference to API object used to communicate with the server.

Returns

List of [AssetGroup](#) objects corresponding to the asset groups in the organization.

Return type

list[[AssetGroup](#)]

get_statistics()

For this group, return statistics about its group membership.

The statistics include how many of the group's members belong to other groups, and how many members belong to groups without policy association.

See [this page](#) for more details on the structure of the return value from this method.

Required Permissions:

group-management(READ)

Returns

A dict with two elements. The “intersections” element contains elements detailing which groups share members with this group, and which members they are. The “unassigned_properties” element contains elements showing which members belong to groups without policy association.

Return type

dict

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

list_member_ids(rows=20, start=0)

Gets a list of all member IDs in the group, optionally constrained by membership type.

Required Permissions:

group-management(READ)

Parameters

- **rows** (*int*) – Maximum number of rows to retrieve from the server. The function may return fewer member IDs if filtering is applied to the output. Default is 20.
- **start** (*int*) – Starting row to retrieve from the server; used to implement pagination. Default is 0.

Returns

List of dictionaries that contain the integer element `external_member_id` for the device ID, the boolean element `dynamic` which is True if the group member is there due to the group's dynamic query, and the boolean element `manual` which is True if the group member was manually added. (It is possible for both `dynamic` and `manual` to be True.)

Return type

list[dict]

list_members(rows=20, start=0, membership='ALL')

Gets a list of all member devices in the group, optionally constrained by membership type.

Required Permissions:

group-management(READ), devices(READ)

Parameters

- **rows** (*int*) – Maximum number of rows to retrieve from the server. The function may return fewer member IDs if filtering is applied to the output. Default is 20.
- **start** (*int*) – Starting row to retrieve from the server; used to implement pagination. Default is 0.
- **membership** (*str*) – Can restrict the types of members that are returned by this method. Values are “ALL” to return all members, “DYNAMIC” to return only members that were added via the asset group query, or “MANUAL” to return only manually-added members. Default is “ALL”.

Returns

List of Device objects comprising the membership of the group.

Return type

list[*Device*]

preview_add_members(*devices*)

Previews changes to the effective policies for devices which result from adding them to this asset group.

Required Permissions:

org.policies (READ)

Parameters

devices (*list*) – The devices which will be added to this asset group. Each entry in this list is either an integer device ID or a Device object.

Returns

A list of DevicePolicyChangePreview objects representing the assets that change which policy is effective as the result of this operation.

Return type

list[*DevicePolicyChangePreview*]

classmethod preview_add_members_to_groups(*cb, members, groups*)

Previews changes to the effective policies for devices which result from adding them to asset groups.

Required Permissions:

org.policies (READ)

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **members** (*list*) – The devices which will be added to new asset groups. Each entry in this list is either an integer device ID or a Device object.
- **groups** (*list*) – The asset groups to which the devices will be added. Each entry in this list is either a string asset group ID or an AssetGroup object.

Returns

A list of DevicePolicyChangePreview objects representing the assets that change which policy is effective as the result of this operation.

Return type

list[*DevicePolicyChangePreview*]

classmethod `preview_create_asset_group(cb, policy_id, query)`

Previews changes to the effective policies for devices which result from creating a new asset group.

Required Permissions:

org.policies (READ)

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **policy_id** (*int*) – The ID of the policy to be added to the new asset group.
- **query** (*str*) – The query string to be used for the new asset group.

Returns

A list of **DevicePolicyChangePreview** objects representing the assets that change which policy is effective as the result of this operation.

Return type

list[[DevicePolicyChangePreview](#)]

preview_delete()

Previews changes to the effective policies for devices which result from this asset group being deleted.

Required Permissions:

org.policies (READ)

Returns

A list of **DevicePolicyChangePreview** objects representing the assets that change which policy is effective as the result of this operation.

Return type

list[[DevicePolicyChangePreview](#)]

classmethod `preview_delete_asset_groups(cb, groups)`

Previews changes to the effective policies for devices which result from deleting asset groups.

Required Permissions:

org.policies (READ)

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **groups** (*list*) – The asset groups which will be deleted. Each entry in this list is either a string asset group ID or an `AssetGroup` object.

Returns

A list of **DevicePolicyChangePreview** objects representing the assets that change which policy is effective as the result of this operation.

Return type

list[[DevicePolicyChangePreview](#)]

preview_remove_members(devices)

Previews changes to the effective policies for devices which result from removing them from this asset group.

Required Permissions:

org.policies (READ)

Parameters

devices (*list*) – The devices which will be removed from this asset group. Each entry in this list is either an integer device ID or a Device object.

Returns

A list of **DevicePolicyChangePreview** objects representing the assets that change which policy is effective as the result of this operation.

Return type

list[[*DevicePolicyChangePreview*](#)]

classmethod `preview_remove_members_from_groups`(*cb, members, groups*)

Previews changes to the effective policies for devices which result from removing them from asset groups.

Required Permissions:

org.policies (READ)

Parameters

- **cb** ([*BaseAPI*](#)) – Reference to API object used to communicate with the server.
- **members** (*list*) – The devices which will be removed from asset groups. Each entry in this list is either an integer device ID or a Device object.
- **groups** (*list*) – The asset groups from which the devices will be removed. Each entry in this list is either a string asset group ID or an AssetGroup object.

Returns

A list of **DevicePolicyChangePreview** objects representing the assets that change which policy is effective as the result of this operation.

Return type

list[[*DevicePolicyChangePreview*](#)]

preview_save()

Previews changes to the effective policies for devices which result from unsaved changes to this asset group.

Required Permissions:

org.policies (READ)

Returns

A list of **DevicePolicyChangePreview** objects representing the assets that change which policy is effective as the result of this operation.

Return type

list[[*DevicePolicyChangePreview*](#)]

classmethod `preview_update_asset_groups`(*cb, groups, policy_id=None, query=None, remove_policy_id=False, remove_query=False*)

Previews changes to the effective policies for devices which result from changes to asset groups.

Required Permissions:

org.policies (READ)

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **groups** (*list*) – The asset groups which will be updated. Each entry in this list is either a string asset group ID or an `AssetGroup` object.
- **policy_id** (*int*) – If this is not `None` and `remove_policy_id` is `False`, contains the ID of the policy to be assigned to the specified groups. Default is `None`.
- **query** (*str*) – If this is not `None` and `remove_query` is `False`, contains the new query string to be assigned to the specified groups. Default is `None`.
- **remove_policy_id** (*bool*) – If this is `True`, indicates that the specified groups will have their policy ID removed entirely. Default is `False`.
- **remove_query** (*bool*) – If this is `True`, indicates that the specified groups will have their query strings removed entirely. Default is `False`.

Returns

A list of `DevicePolicyChangePreview` objects representing the assets that change which policy is effective as the result of this operation.

Return type

`list[DevicePolicyChangePreview]`

refresh()

Reload this object from the server.

remove_members(*members*)

Removes members from this asset group.

Required Permissions:

group-management(DELETE)

Parameters

members (*int*, [Device](#), or *list*) – The members to be removed from the group. This may be an integer device ID, a `Device` object, or a list of either integers or `Device` objects.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

[MutableBaseModel](#)

swagger_meta_file

The valid values for the 'filter' parameter to `list_members()`.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(*fulltouch=False*)

Force this object to be considered as changed.

validate()

Validates this object.

Returns

True if the object is validated.

Return type

bool

Raises*InvalidObjectError* – If the object has missing fields.**class AssetGroupQuery**(*doc_class, cb*)

Bases: *BaseQuery*, *QueryBuilderSupportMixin*, *IterableQueryMixin*,
CriteriaBuilderSupportMixin, *AsyncQueryMixin*

Query object that is used to locate AssetGroup objects.

The AssetGroupQuery is constructed via SDK functions like the `select()` method on CbCloudAPI. The user would then add a query and/or criteria to it before iterating over the results.

The following criteria are supported on AssetGroupQuery via the standard `add_criteria()` method:

- **discovered**: bool - Whether the asset group has been discovered or not.
- **name**: str - The asset group name to be matched.
- **policy_id**: int - The policy ID to be matched, expressed as an integer.
- **group_id**: str - The asset group ID to be matched, expressed as a GUID.

Initialize the AssetGroupQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None, **kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(*q=None, **kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- *q* (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

set_rows(*rows*)

Sets the number of query rows to fetch in each batch from the server.

Parameters

- **rows** (*int*) – The number of rows to be fetched from the server at a time. Default is 100.

Returns

This instance.

Return type

AssetGroupQuery

sort_by(*key, direction='ASC'*)

Sets the sorting behavior on a query's results.

Example

```
>>> cb.select(AssetGroup).sort_by("name")
```

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

This instance.

Return type

AssetGroupQuery

update_criteria(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None*, ***kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, `QueryBuilder`, or `solrq.Q` object
- ****kwargs** (*dict*) – Arguments to construct a `solrq.Q` with

Returns

This Query object.

Return type

Query

4.11.5 Audit Module

Model and Query Classes for Platform Auditing

class AuditLog(*cb*, *model_unique_id*, *initial_data=None*)

Bases: [*UnrefreshableModel*](#)

Model class which represents audit log events. Mostly for future implementation.

Creation of AuditLog objects is not yet implemented.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

static `get_auditlogs(cb)`

Retrieve queued audit logs from the Carbon Black Cloud server.

Required Permissions:

org.audits (READ)

Parameters

cb ([BaseAPI](#)) – Reference to API object used to communicate with the server.

Returns

List of dictionary objects representing the audit logs, or an empty list if none available.

Return type

list[dict]

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

4.11.6 Devices Module

The model and query classes for referencing platform devices.

A *platform device* represents an endpoint registered with the Carbon Black Cloud that runs a sensor, which communicates with Carbon Black analytics and the console. Using these classes, you can search for devices using a wide variety of filterable fields, such as policy ID, status, or operating system. You can also perform actions on individual devices such as quarantining/unquarantining them, enabling or disabling bypass, or upgrading them to a new sensor version.

Typical usage example:

```
# assume "cb" is an instance of CBCloudAPI
query = cb.select(Device).where(os="WINDOWS").set_policy_ids([142857])
for device in query:
    device.quarantine(True)
```

class `Device(cb, model_unique_id, initial_data=None)`

Bases: [PlatformModel](#)

Represents a device (endpoint) within the Carbon Black Cloud.

Device objects are generally located through a search (using `DeviceSearchQuery`) before they can be operated on.

Parameters

- **activation_code** – Device activation code
- **activation_code_expiry_time** – When the expiration code expires and cannot be used to register a device
- **ad_group_id** – Device's AD group

- **asset_group** – The asset groups that this device is a member of.
- **av_ave_version** – AVE version (part of AV Version)
- **av_engine** – Current AV version
- **av_last_scan_time** – Last AV scan time
- **av_master** – Whether the device is an AV Master (?)
- **av_pack_version** – Pack version (part of AV Version)
- **av_product_version** – AV Product version (part of AV Version)
- **av_status** – AV Statuses
- **av_update_servers** – Device’s AV servers
- **av_vdf_version** – VDF version (part of AV Version)
- **current_sensor_policy_name** – Current MSM policy name
- **deregistered_time** – When the device was deregistered with the PSC backend
- **device_id** – ID of the device
- **device_meta_data_item_list** – MSM Device metadata
- **device_owner_id** – ID of the user who owns the device
- **email** – Email of the user who owns the device
- **encoded_activation_code** – Encoded device activation code
- **first_name** – First name of the user who owns the device
- **id** – ID of the device
- **last_contact_time** – Time the device last checked into the PSC backend
- **last_device_policy_changed_time** – Last time the device’s policy was changed
- **last_device_policy_requested_time** – Last time the device requested policy updates
- **last_external_ip_address** – Device’s external IP
- **last_internal_ip_address** – Device’s internal IP
- **last_location** – Location of the device (on/off-premises)
- **last_name** – Last name of the user who owns the device
- **last_policy_updated_time** – Last time the device was MSM processed
- **last_reported_time** – Time when device last reported an event to PSC backend
- **last_reset_time** – When the sensor was last reset
- **last_shutdown_time** – When the device last shut down
- **linux_kernel_version** – Linux kernel version
- **login_user_name** – Last active logged in username
- **mac_address** – Device’s hardware MAC address
- **middle_name** – Middle name of the user who owns the device
- **name** – Device Hostname
- **organization_id** – Org ID to which the device belongs

- **organization_name** – Name of the org that owns this device
- **os** – Device type
- **os_version** – Version of the OS
- **passive_mode** – Whether the device is in passive mode (bypass?)
- **policy_id** – ID of the policy this device is using
- **policy_name** – Name of the policy this device is using
- **policy_override** – Manually assigned policy (overrides mass sensor management)
- **quarantined** – Whether the device is quarantined
- **registered_time** – When the device was registered with the PSC backend
- **scan_last_action_time** – Not used. Intended for when the background scan was last active
- **scan_last_complete_time** – Not Used. Intended for when the background scan was last completed
- **scan_status** – Not Used. Intended for Background scan status
- **sensor_out_of_date** – Whether the device is out of date
- **sensor_states** – Active sensor states
- **sensor_version** – Version of the PSC sensor
- **status** – Device status
- **target_priority_type** – Priority of the device
- **uninstall_code** – Code to enter to uninstall this device
- **vdi_base_device** – VDI Base device
- **virtual_machine** – Whether this device is a Virtual Machine (VMware AppDefense integration)
- **virtualization_provider** – VM Virtualization Provider
- **windows_platform** – Type of windows platform (client/server, x86/x64)
- **deployment_type** – Classification determined by the device lifecycle management policy

Initialize the Device object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the device represented.
- **initial_data** (*dict*) – Initial data used to populate the device.

add_to_groups(*groups*)

Given a list of asset groups, adds this device to each one as a member.

Parameters

- **groups** (*list*[[AssetGroup](#)]) – The list of groups to add this device to.

add_to_groups_by_id(*group_ids*)

Given a list of asset group IDs, adds this device to each one as a member.

Parameters

group_ids (*list[str]*) – The list of group IDs to add this device to.

background_scan(*flag*)

Set the background scan option for this device.

Required Permissions:

device.bg-scan(EXECUTE)

Parameters

flag (*bool*) – True to turn background scan on, False to turn it off.

Returns

The JSON output from the request.

Return type

str

bypass(*flag*)

Set the bypass option for this device.

Required Permissions:

device.bypass(EXECUTE)

Parameters

flag (*bool*) – True to enable bypass, False to disable it.

Returns

The JSON output from the request.

Return type

str

delete_sensor()

Delete this sensor device.

Required Permissions:

device.deregistered(DELETE)

Returns

The JSON output from the request.

Return type

str

property deviceId

Warn user that Platform Devices use ‘id’, not ‘device_id’.

Platform Device APIs return ‘id’ in API responses, where Endpoint Standard APIs return ‘deviceId’.

Raises

AttributeError – In all cases.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_asset_group_ids(*membership='ALL'*)

Finds the list of asset group IDs that this device is a member of.

Parameters

membership (*str*) – Can restrict the types of group membership returned by this method. Values are “ALL” to return all groups, “DYNAMIC” to return only groups that each member belongs to via the asset group query, or “MANUAL” to return only groups that the members were manually added to. Default is “ALL”.

Returns

A list of asset group IDs this device belongs to.

Return type

list[str]

get_asset_groups(*membership='ALL'*)

Finds the list of asset groups that this device is a member of.

Required Permissions:

group-management(READ)

Parameters

membership (*str*) – Can restrict the types of group membership returned by this method. Values are “ALL” to return all groups, “DYNAMIC” to return only groups that each member belongs to via the asset group query, or “MANUAL” to return only groups that the members were manually added to. Default is “ALL”.

Returns

A list of asset groups this device belongs to.

Return type

list[*AssetGroup*]

classmethod get_asset_groups_for_devices(*cb*, *devices*, *membership='ALL'*)

Given a list of devices, returns lists of asset groups that they are members of.

Required Permissions:

group-management(READ)

Parameters

- **cls** (*class*) – Class associated with the Device object.
- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **devices** (*int*, *Device*, or *list*) – The devices to find the group membership of. This may be an integer device ID, a Device object, or a list of either integers or Device objects.

- **membership** (*str*) – Can restrict the types of group membership returned by this method. Values are “ALL” to return all groups, “DYNAMIC” to return only groups that each member belongs to via the asset group query, or “MANUAL” to return only groups that the members were manually added to. Default is “ALL”.

Returns

A dict containing member IDs as keys, and lists of group IDs as values.

Return type

dict

get_vulnerability_summary(*category=None*)

Get the vulnerabilities associated with this device.

Required Permissions:

vulnerabilityAssessment.data(READ)

Parameters

category (*string*) – (optional) Vulnerability category (OS, APP).

Returns

Summary of the vulnerabilities for this device.

Return type

dict

get_vulnerabilities()

Return a query to get an operating system or application vulnerability list for this device.

Returns

Query for searching for vulnerabilities on this device.

Return type

VulnerabilityQuery

lr_session(*async_mode=False*)

Retrieve a Live Response session object for this Device.

Required Permissions:

org.liveresponse.session(CREATE)

Returns

Live Response session for the Device.

Return type

LiveResponseSession

Raises

ApiError – If there is an error establishing a Live Response session for this Device.

property nsx_available

Returns whether NSX actions are available on this device.

Returns

True if NSX actions are available, False if not.

Return type

bool

nsx_remediation(*tag*, *set_tag=True*)

Start an NSX Remediation job on this device to change the tag.

Required Permissions:

appliances.nsx.remediation(EXECUTE)

Parameters

- **tag** (*str*) – The NSX tag to apply to this device. Valid values are “CB-NSX-Quarantine”, “CB-NSX-Isolate”, and “CB-NSX-Custom”.
- **set_tag** (*bool*) – True to toggle the specified tag on, False to toggle it off. Default True.

Returns

The object representing all running jobs. None if the operation is a no-op.

Return type

NSXRemediationJob

classmethod preview_add_policy_override_for_devices(*cb*, *policy_id*, *devices*)

Previews changes to the effective policies for devices which result from setting a policy override on them.

Required Permissions:

org.policies (READ)

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **policy_id** (*int*) – The ID of the policy to be added to the devices as an override.
- **devices** (*list*) – The devices which will have their policies overridden. Each entry in this list is either an integer device ID or a Device object.

Returns

A list of **DevicePolicyChangePreview** objects representing the assets that change which policy is effective as the result of this operation.

Return type

list[*DevicePolicyChangePreview*]

preview_remove_policy_override()

Previews changes to this device’s effective policy which result from removing its policy override.

Required Permissions:

org.policies (READ)

Returns

A list of **DevicePolicyChangePreview** objects representing the assets that change which policy is effective as the result of this operation.

Return type

list[*DevicePolicyChangePreview*]

classmethod preview_remove_policy_override_for_devices(*cb*, *devices*)

Previews changes to the effective policies for devices which result from removing their policy override.

Required Permissions:

org.policies (READ)

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **devices** (*list*) – The devices which will have their policy overrides removed. Each entry in this list is either an integer device ID or a [Device](#) object.

Returns

A list of [DevicePolicyChangePreview](#) objects representing the assets that change which policy is effective as the result of this operation.

Return type

list[[DevicePolicyChangePreview](#)]

quarantine(flag)

Set the quarantine option for this device.

Required Permissions:

device.quarantine(EXECUTE)

Parameters

flag (*bool*) – True to enable quarantine, False to disable it.

Returns

The JSON output from the request.

Return type

str

refresh()

Reload this object from the server.

remove_from_groups(groups)

Given a list of asset groups, removes this device from each one as a member.

Parameters

groups (*list*[[AssetGroup](#)]) – The list of groups to remove this device from.

remove_from_groups_by_id(group_ids)

Given a list of asset group IDs, removes this device from each one as a member.

Parameters

group_ids (*list*[*str*]) – The list of group IDs to remove this device from.

swagger_meta_file

The valid values for the ‘filter’ parameter to [get_asset_groups_for_devices\(\)](#).

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

uninstall_sensor()

Uninstall this sensor device.

Required Permissions:

device.uninstall(EXECUTE)

Returns

The JSON output from the request.

Return type

str

update_policy(policy_id)

Set the current policy for this device.

Required Permissions:

device.policy(UPDATE)

Parameters

policy_id (int) – ID of the policy to set for the device.

Returns

The JSON output from the request.

Return type

str

update_sensor_version(sensor_version)

Update the sensor version for this device.

Required Permissions:

org.kits(EXECUTE)

Parameters

sensor_version (dict) – New version properties for the sensor.

Returns

The JSON output from the request.

Return type

str

vulnerability_refresh()

Refresh vulnerability information for the device.

Required Permissions:

vulnerabilityAssessment.data(EXECUTE)

class DeviceFacet(cb, model_unique_id, initial_data=None)

Bases: [UnrefreshableModel](#)

Represents a device field in a facet search.

Faceting is a search technique that categorizes search results according to common attributes. This allows users to explore and discover information within a dataset, in this case, the set of devices.

Example:

```
>>> facets = api.select(Device).facets(['policy_id'])
>>> for value in facets[0].values_:
...     print(f"Policy ID {value.id}: {value.total} device(s)")
```

Parameters

- **field** – Name of the field being faceted
- **values** – The values of the faceted field.

Initialize the DeviceFacet object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – Not used.
- **initial_data** (*dict*) – Initial data used to populate the facet.

class DeviceFacetValue(*cb, outer, model_unique_id, initial_data*)

Bases: [UnrefreshableModel](#)

Represents a value of a particular faceted field.

Faceting is a search technique that categorizes search results according to common attributes. This allows users to explore and discover information within a dataset, in this case, the set of devices.

Initialize the DeviceFacetValue object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **outer** ([DeviceFacet](#)) – Reference to outer facet object.
- **model_unique_id** (*str*) – Value ID.
- **initial_data** (*dict*) – Initial data used to populate the facet value.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

query_devices()

Set up a device query to find all devices that match this facet value.

Example

```
>>> facets = api.select(Device).facets(['policy_id'])
>>> for value in facets[0].values_:
...     print(f"Policy ID = {value.id}:")
...     for dev in value.query_devices():
...         print(f"    {dev.name} ({dev.last_external_ip_address})")
```

Returns

A new `DeviceQuery` set with the criteria, which may have additional criteria added to it.

Return type

`DeviceQuery`

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

property values_

Returns the list of facet values for this facet.

class DeviceSearchQuery(doc_class, cb)

Bases: `BaseQuery`, `QueryBuilderSupportMixin`, `CriteriaBuilderSupportMixin`, `IterableQueryMixin`, `AsyncQueryMixin`

Query object that is used to locate Device objects.

The `DeviceSearchQuery` is constructed via SDK functions like the `select()` method on `CBCloudAPI`. The user would then add a query and/or criteria to it before iterating over the results.

Initialize the `DeviceSearchQuery`.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.

`add_criteria(key, newlist)`

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

`all()`

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

`and_(q=None, **kwargs)`

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

`background_scan(scan)`

Set the background scan option for the specified devices.

Required Permissions:

device.bg-scan(EXECUTE)

Parameters

scan (*bool*) – True to turn background scan on, False to turn it off.

Returns

The JSON output from the request.

Return type

str

bypass(*enable*)

Set the bypass option for the specified devices.

Required Permissions:

device.bypass(EXECUTE)

Parameters

enable (*bool*) – True to enable bypass, False to disable it.

Returns

The JSON output from the request.

Return type

str

delete_sensor()

Delete the specified sensor devices.

Required Permissions:

device.deregistered(DELETE)

Returns

The JSON output from the request.

Return type

str

download()

Uses the query parameters that have been set to download all device listings in CSV format.

Deprecated:

Use DeviceSearchQuery.export for increased export capabilities and limits

Example

```
>>> cb.select(Device).set_status(["ALL"]).download()
```

Required Permissions:

device(READ)

Returns

The CSV raw data as returned from the server.

Return type

str

Raises

[*ApiError*](#) – If status values have not been set before calling this function.

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

export()

Starts the process of exporting Devices from the organization in CSV format.

Example

```
>>> cb.select(Device).set_status(["ACTIVE"]).export()
```

Required Permissions:

device(READ)

Returns

The asynchronous job that will provide the export output when the server has prepared it.

Return type

Job

facets(fieldlist, max_rows=0)

Return information about the facets for all matching devices, using the defined criteria.

Example

```
>>> query = api.select(Device).where('')
>>> facets = query.facets(['policy_id', 'status', 'os', 'ad_group_id'])
>>> for f in facets:
...     print(f"Field {f.field} - {len(f.values_)} distinct values")
```

Required Permissions:

device(READ)

Parameters

- **fieldlist** (*list[str]*) – List of facet field names. Valid names are “policy_id”, “status”, “os”, “ad_group_id”, “cloud_provider_account_id”, “auto_scaling_group_name”, and “virtual_private_cloud_id”.
- **max_rows** (*int*) – The maximum number of rows to return. 0 means return all rows.

Returns

A list of facet information.

Return type

list[*DeviceFacet*]

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(*q=None, **kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- ***MoreThanOneResultError*** – If the query returns more than one item
- ***ObjectNotFoundError*** – If the query returns zero items

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

quarantine(*enable*)

Set the quarantine option for the specified devices.

Required Permissions:

device.quarantine(EXECUTE)

Parameters

enable (*bool*) – True to enable quarantine, False to disable it.

Returns

The JSON output from the request.

Return type

str

scroll(rows=10000)

Iteratively paginate all Devices beyond the 10k max search limits.

To fetch the next set of Devices repeatedly call the scroll function until *DeviceSearchQuery.num_remaining* == 0 or no results are returned.

Example

```
>>> cb.select(Device).set_status(["ACTIVE"]).scroll(100)
```

Required Permissions:

device(READ)

Parameters

rows (int) – The number of rows to fetch

Returns

The list of results

Return type

list[*Device*]

set_ad_group_ids(ad_group_ids)

Restricts the devices that this query is performed on to the specified AD group IDs.

Parameters

ad_group_ids (list) – List of AD group IDs to restrict the search to.

Returns

This instance.

Return type

DeviceSearchQuery

set_auto_scaling_group_name(group_names)

Restricts the devices that this query is performed on to the specified auto scaling group names.

Parameters

group_names (list) – List of group names to restrict search to.

Returns

This instance.

Return type

DeviceSearchQuery

set_cloud_provider_account_id(account_ids)

Restricts the devices that this query is performed on to the specified cloud provider account IDs.

Parameters

account_ids (list) – List of account IDs to restrict search to.

Returns

This instance.

Return type

DeviceSearchQuery

set_deployment_type(*deployment_type*)

Restricts the devices that this query is performed on to the specified deployment types.

Parameters

deployment_type (*list*) – List of deployment types to restrict search to.

Returns

This instance.

Return type

DeviceSearchQuery

set_device_ids(*device_ids*)

Restricts the devices that this query is performed on to the specified device IDs.

Parameters

device_ids (*list*) – List of device IDs to restrict the search to.

Returns

This instance.

Return type

DeviceSearchQuery

set_exclude_sensor_versions(*sensor_versions*)

Restricts the devices that this query is performed on to exclude specified sensor versions.

Parameters

sensor_versions (*list*) – List of sensor versions to be excluded.

Returns

This instance.

Return type

DeviceSearchQuery

set_last_contact_time(**args*, ***kwargs*)

Restricts the devices that this query is performed on to the specified last contact time.

Parameters

- ***args** (*list*) – Not used, retained for compatibility.
- ****kwargs** (*dict*) – Keyword arguments to this function. The critical ones are “start” (the start time), “end” (the end time), and “range” (the range value).

Returns

This instance.

Return type

DeviceSearchQuery

set_max_rows(*max_rows*)

Sets the max number of devices to fetch in a singular query

Parameters

max_rows (*integer*) – Max number of devices. Must be in the range (0, 10000).

Returns

This instance.

Return type

DeviceSearchQuery

set_os(*operating_systems*)

Restricts the devices that this query is performed on to the specified operating systems.

Parameters

operating_systems (*list*) – List of operating systems to restrict search to. Valid values in this list are “WINDOWS”, “ANDROID”, “MAC”, “IOS”, “LINUX”, and “OTHER”.

Returns

This instance.

Return type

DeviceSearchQuery

set_policy_ids(*policy_ids*)

Restricts the devices that this query is performed on to the specified policy IDs.

Parameters

policy_ids (*list*) – List of policy IDs to restrict the search to.

Returns

This instance.

Return type

DeviceSearchQuery

set_status(*statuses*)

Restricts the devices that this query is performed on to the specified status values.

Parameters

statuses (*list*) – List of statuses to restrict search to. Valid values in this list are “PENDING”, “REGISTERED”, “UNINSTALLED”, “DEREGISTERED”, “ACTIVE”, “INACTIVE”, “ERROR”, “ALL”, “BYPASS_ON”, “BYPASS”, “QUARANTINE”, “SENSOR_OUTOFDATE”, “DELETED”, and “LIVE”.

Returns

This instance.

Return type

DeviceSearchQuery

set_target_priorities(*target_priorities*)

Restricts the devices that this query is performed on to the specified target priority values.

Parameters

target_priorities (*list*) – List of priorities to restrict search to. Valid values in this list are “LOW”, “MEDIUM”, “HIGH”, and “MISSION_CRITICAL”.

Returns

This instance.

Return type

DeviceSearchQuery

set_virtual_private_cloud_id(*cloud_ids*)

Restricts the devices that this query is performed on to the specified virtual private cloud IDs.

Parameters

cloud_ids (*list*) – List of cloud IDs to restrict search to.

Returns

This instance.

Return type

DeviceSearchQuery

sort_by(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Example

```
>>> cb.select(Device).sort_by("status")
```

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

This instance.

Return type

DeviceSearchQuery

uninstall_sensor()

Uninstall the specified sensor devices.

Required Permissions:

device.uninstall(EXECUTE)

Returns

The JSON output from the request.

Return type

str

update_criteria(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

update_policy(*policy_id*)

Set the current policy for the specified devices.

Required Permissions:

device.policy(UPDATE)

Parameters

policy_id (*int*) – ID of the policy to set for the devices.

Returns

The JSON output from the request.

Return type

str

update_sensor_version(*sensor_version*)

Update the sensor version for the specified devices.

Required Permissions:

org.kits(EXECUTE)

Parameters

sensor_version (*dict*) – New version properties for the sensor.

Returns

The JSON output from the request.

Return type

str

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

```
log = <Logger cbc_sdk.platform.devices (WARNING)>
```

“Device Models

4.11.7 Events Module

Model and Query Classes for Events

class Event(*cb*, *model_unique_id=None*, *initial_data=None*, *force_init=False*, *full_doc=True*)

Bases: [UnrefreshableModel](#)

Events can be queried for via *CBCloudAPI.select* or an already selected process with *Process.events()*.

Examples

```
>>> events_query = (api.select(Event).where(process_guid=
    "WNEXFKQ7-00050603-0000066c-00000000-1d6c9acb43e29bb"))
# retrieve results synchronously
>>> events = [event for event in events_query]
# retrieve results asynchronously
>>> future = events_query.execute_async()
>>> events = future.result()
# use an already selected process
>>> process = api.select(Process, "WNEXFKQ7-00050603-0000066c-00000000-
    ↪1d6c9acb43e29bb")
>>> events_query = process.events()
>>> events = [event for event in events_query]
```

Initialize the Event object.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **model_unique_id** (*str*) – The unique ID for this particular instance of the model object.
- **initial_data** (*dict*) – The data to use when initializing the model object.
- **force_init** (*bool*) – True to force object initialization.
- **full_doc** (*bool*) – True to mark the object as fully initialized.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class EventFacet(*cb, model_unique_id, initial_data*)Bases: [UnrefreshableModel](#)

Represents the results of an EventFacetQuery.

EventFacet objects contain both Terms and Ranges. Each of those contain facet fields and values.

Access all of the Terms facet data with [EventFacet.Terms.facets\(\)](#) or see just the field names with [EventFacet.Terms.fields\(\)](#).Access all of the Ranges facet data with [EventFacet.Ranges.facets\(\)](#) or see just the field names with [EventFacet.Ranges.fields\(\)](#).Event Facets can be queried for via `CBCloudAPI.select(EventFacet)`. Specify a Process GUID with `.where(process_guid="example_guid")`, and facet field(s) with `.add_facet_field("my_facet_field")`.**Examples**

```
>>> event_facet_query = (api.select(EventFacet).where(process_guid=
"WNEXFKQ7-00050603-0000066c-00000000-1d6c9acb43e29bb"))
>>> event_facet_query.add_facet_field("event_type")
# retrieve results synchronously
>>> facet = event_facet_query.results
# retrieve results asynchronously
>>> future = event_facet_query.execute_async()
>>> result = future.result()
# result is a list with one item, so access the first item
>>> facet = result[0]
```

Initialize an EventFacet object with initial_data.

class Ranges(*cb, initial_data*)Bases: [UnrefreshableModel](#)

Represents the range (bucketed) facet fields and values associated with an Event Facet query.

Initialize a ProcessFacet Ranges object with initial_data.

property facetsReturns the reified `EventFacet.Terms._facets` for this result.**property fields**

Returns the ranges fields for this result.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class Terms(*cb, initial_data*)

Bases: [*UnrefreshableModel*](#)

Represents the facet fields and values associated with an Event Facet query.

Initialize a ProcessFacet Terms object with initial_data.

property facets

Returns the terms' facets for this result.

property fields

Returns the terms facets' fields for this result.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

property ranges_

Returns the reified *EventFacet.Ranges* for this result.

refresh()

Reload this object from the server.

property terms_

Returns the reified *EventFacet.Terms* for this result.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class EventFacetQuery(*cls, cb, query=None*)

Bases: *FacetQuery*

Represents the logic for an Event Facet query.

Initialize the FacetQuery object.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

add_exclusions(*key, newlist*)

Add to the exclusions on this query with a custom exclusions key.

Will overwrite any existing exclusion for the specified key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).add_exclusions("type", ["WATCHLIST"])
>>> query = api.select(Alert).add_exclusions("type", "WATCHLIST")
```

add_facet_field(*field*)

Sets the facet fields to be received by this query.

Parameters

field (*str* or [*str*]) – Field(s) to be received.

Returns

The Query object that will receive the specified field(s).

Return type

Query (AsyncQuery)

Example

```
>>> cb.select(ProcessFacet).add_facet_field(["process_name", "process_username
↪ "])
```

add_range(*range*)

Sets the facet ranges to be received by this query.

Parameters

range (*dict* or [*dict*]) – Range(s) to be received.

Returns

The Query object that will receive the specified range(s).

Return type

Query (AsyncQuery)

Note: The range parameter must be in this dictionary format:

```
{
    "bucket_size": "<object>",
    "start": "<object>",
    "end": "<object>",
    "field": "<string>"
},
```

where “bucket_size”, “start”, and “end” can be numbers or ISO 8601 timestamps.

Examples

```
>>> cb.select(ProcessFacet).add_range({"bucket_size": 5, "start": 0, "end": 10,
↪ "field": "netconn_count"})
>>> cb.select(ProcessFacet).add_range({"bucket_size": "+1DAY", "start": "2020-
↪ 11-01T00:00:00Z",
... "end": "2020-11-12T00:00:00Z", "field": "backend_timestamp"})
```

and_(*q=None*, ***kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

limit(*limit*)

Sets the maximum number of facets per category (i.e. any Process Search Fields in *self._fields*).

The default limit for Process Facet searches in the Carbon Black Cloud backend is 100.

Parameters

limit (*int*) – Maximum number of facets per category.

Returns

The Query object with new limit parameter.

Return type

Query (AsyncQuery)

Example

```
>>> cb.select(ProcessFacet).where(process_name="foo.exe").limit(50)
```

not_(*q=None*, ***kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

property results

Save query results to `self._results` with `self._search()` method.

set_rows(*rows*)

Sets the number of facet results to return with the query.

Parameters

rows (*int*) – Number of rows to return.

Returns

The Query object with the new rows parameter.

Return type

Query (AsyncQuery)

Example

```
>>> cb.select(ProcessFacet).set_rows(50)
```

set_time_range(*start=None, end=None, window=None*)

Sets the 'time_range' query body parameter, determining a time window based on 'device_timestamp'.

Parameters

- **start** (*str in ISO 8601 timestamp*) – When to start the result search.
- **end** (*str in ISO 8601 timestamp*) – When to end the result search.
- **window** (*str*) – Time window to execute the result search, ending on the current time.
- **"-2w"** (*Should be in the form*) –
- **y=year** (*where*) –
- **w=week** –
- **d=day** –
- **h=hour** –
- **m=minute** –

- `s=second.` –

Note:

- *window* will take precedent over *start* and *end* if provided.
-

Examples

```
>>> query = api.select(Process).set_time_range(start="2020-10-20T20:34:07Z").
↳where("query is required")
>>> second_query = api.select(Process).
...     set_time_range(start="2020-10-20T20:34:07Z", end="2020-10-30T20:34:07Z
↳").where("query is required")
>>> third_query = api.select(Process).set_time_range(window='-3d').where("query_
↳is required")
```

timeout(*msecs*)

Sets the timeout on an AsyncQuery.

Parameters

msecs (*int*) – Timeout duration, in milliseconds. This value can never be greater than the configured default timeout. If this is 0, the configured default timeout value is used.

Returns

The Query object with new milliseconds parameter.

Return type

Query (AsyncQuery)

Example

```
>>> cb.select(ProcessFacet).where(process_name="foo.exe").timeout(5000)
```

update_criteria(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

update_exclusions(*key*, *newlist*)

Update the exclusion on this query with a custom exclusion key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*list*) – List of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).update_exclusions("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None*, ***kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, `QueryBuilder`, or `solrq.Q` object
- ****kwargs** (*dict*) – Arguments to construct a `solrq.Q` with

Returns

This Query object.

Return type

Query

class EventQuery(*doc_class*, *cb*)

Bases: *Query*

Represents the logic for an Event query.

Initialize the Query object.

Parameters

- **doc_class** (*class*) – The class of the model this query returns.
- **cb** (`CBCloudAPI`) – A reference to the `CBCloudAPI` object.

add_criteria(*key*, *newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

add_exclusions(*key*, *newlist*)

Add to the exclusions on this query with a custom exclusions key.

Will overwrite any existing exclusion for the specified key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).add_exclusions("type", ["WATCHLIST"])
>>> query = api.select(Alert).add_exclusions("type", "WATCHLIST")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None*, ***kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type*Query***batch_size**(*new_batch_size*)

Set the batch size of the paginated query.

Parameters

new_batch_size (*int*) – The new batch size.

Returns

A new query with the updated batch size.

Return type*PaginatedQuery***execute_async**()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type*Future***first**()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type*obj***not_**(*q=None, **kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type*Query***one**()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type*obj***Raises**

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

set_fields(*fields*)

Sets the fields to be returned with the response.

Parameters

fields (*str* or *list[str]*) – Field or list of fields to be returned.

set_rows(*rows*)

Sets the ‘rows’ query body parameter, determining how many rows of results to request.

Parameters

rows (*int*) – How many rows to request.

set_start(*start*)

Sets the ‘start’ query body parameter, determining where to begin retrieving results from.

Parameters

start (*int*) – Where to start results from.

set_time_range(*start=None, end=None, window=None*)

Sets the ‘time_range’ query body parameter, determining a time window based on ‘device_timestamp’.

Parameters

- **start** (*str in ISO 8601 timestamp*) – When to start the result search.
- **end** (*str in ISO 8601 timestamp*) – When to end the result search.
- **window** (*str*) – Time window to execute the result search, ending on the current time. Should be in the form “-2w”, where y=year, w=week, d=day, h=hour, m=minute, s=second.

Note:

- *window* will take precedent over *start* and *end* if provided.
-

Examples

```
>>> query = api.select(Process).set_time_range(start="2020-10-20T20:34:07Z").
↳ where("query is required")
>>> second_query = api.select(Process).
...     set_time_range(start="2020-10-20T20:34:07Z", end="2020-10-30T20:34:07Z
↳ ").where("query is required")
>>> third_query = api.select(Process).set_time_range(window='-3d').where("query_
↳ is required")
```


sort_by(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

The query with sorting parameters.

Return type

Query

Example

```
>>> cb.select(Process).where(process_name="cmd.exe").sort_by("device_timestamp")
```

update_criteria(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

update_exclusions(*key*, *newlist*)

Update the exclusion on this query with a custom exclusion key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*list*) – List of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).update_exclusions("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, `QueryBuilder`, or `solrq.Q` object
- ****kwargs** (*dict*) – Arguments to construct a `solrq.Q` with

Returns

This Query object.

Return type

Query

4.11.8 Grants Module

Model and Query Classes for Administrative Grants and Profiles

class `Grant`(*cb, model_unique_id, initial_data=None*)

Bases: `MutableBaseModel`

Represents a grant of access to the Carbon Black Cloud.

Parameters

- **principal** – URN of principal
- **expires** – Date and time the grant expires
- **roles** – URNs of roles assigned to grant (obsolete)
- **profiles** – Profiles assigned to this grant
- **org_ref** – URN of org that this grant references
- **principal_name** – Name of principal
- **created_by** – URN of user that created this grant
- **updated_by** – URN of user that last updated this grant
- **create_time** – Date and time the grant was created
- **update_time** – Date and time the grant was last updated
- **can_manage** – True if can manage (TBD)

Initialize the Grant object.

Parameters

- **cb** (`BaseAPI`) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – URN of the principal associated with this grant.

- **initial_data** (*dict*) – Initial data used to populate the grant.

class GrantBuilder(*cb, principal*)

Bases: object

Auxiliary object used to construct a new grant.

Creates the empty GrantBuilder object.

Parameters

- **cb** ([CBCloudAPI](#)) – The reference to the API object that accesses the server.
- **principal** (*str*) – The URN for the principal.

add_role(*role*)

Adds a role to be associated with the new grant.

Parameters

role (*str*) – URN of the role to be added.

Returns

This object.

Return type

[GrantBuilder](#)

build()

Builds the new Grant object from the entered data.

Returns

The new Grant object.

Return type

[Grant](#)

create_profile(*template=None*)

Returns either a new Profile, or a ProfileBuilder to begin the process of adding profile to the new grant.

Parameters

template (*dict*) – Optional template to use for creating the profile object.

Returns

If a template was specified, return the new Profile object.

ProfileBuilder: If template was None, returns a ProfileBuilder object. Call methods on it to set up the new profile, and then call build() to create the new profile.

Return type

[Profile](#)

set_org(*org*)

Sets the organization reference to be associated with the new grant.

Parameters

org (*str*) – Organization key or URN of the organization.

Returns

This object.

Return type

[GrantBuilder](#)

set_principal_name(*name*)

Sets the principal name to be associated with the new object.

Parameters

name (*str*) – Principal name to be used.

Returns

This object.

Return type*GrantBuilder***set_roles(*roles*)**

Sets the roles to be associated with the new grant.

Parameters

roles (*list*) – List of role URNs.

Returns

This object.

Return type*GrantBuilder*

class Profile(*cb, grant, model_unique_id, initial_data=None*)

Bases: *MutableBaseModel*

Represents an access profile assigned to a grant.

Parameters

- **profile_uuid** – UUID identifying this profile
- **orgs** – Organization references for this profile
- **org_groups** – Organization groups added to this grant (TBD)
- **roles** – URNs of roles assigned to profile
- **conditions** – Access conditions to be imposed on this profile
- **can_manage** – True if can manage (TBD)

Initialize the Profile object.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **grant** (*Grant*) – Reference to the Grant containing this Profile.
- **model_unique_id** (*str*) – UUID of this profile.
- **initial_data** (*dict*) – Initial data used to populate the profile.

property allowed_orgs

Returns the list of organization URNs allowed by this profile.

delete()

Delete this object.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type*Any***is_dirty()**

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

matches_template(*template*)

Returns whether or not the profile matches the given template.

Parameters

template (*dict*) – The profile template to match against.

Returns

True if this profile matches the template, False if not.

Return type

bool

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type*MutableBaseModel***set_disabled**(*flag*)

Sets the “disabled” flag on a profile.

Parameters

flag (*bool*) – True to disable the profile, False to enable it.

set_expiration(*expiration*)

Sets the expiration time on a profile.

Parameters

expiration (*str*) – Expiration time to set on the profile (ISO 8601 format).

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(*fulltouch=False*)

Force this object to be considered as changed.

validate()

Validates this object.

Returns

True if the object is validated.

Return type

bool

Raises

InvalidObjectError – If the object has missing fields.

class ProfileBuilder(*grant*)

Bases: object

Auxiliary object used to construct a new profile on a grant.

Create the empty ProfileBuilder object.

Parameters

grant (*Grant/GrantBuilder*) – The grant or GrantBuilder the new profile will be attached to.

add_org(org)

Adds the specified organization to the list of organizations for which the new profile is allowed.

Parameters

org (*str*) – Organization key or URN of the organization to be added.

Returns

This object.

Return type

ProfileBuilder

add_role(role)

Adds a role identifier to the list of roles associated with the new profile.

Parameters

role (*str*) – URN of the role to add.

Returns

This object.

Return type

ProfileBuilder

build()

Builds the new Profile object from the entered data.

Returns

The new Profile object.

Return type

Profile

set_conditions(conditions_structure)

Sets the access conditions associated with the new profile.

Parameters

conditions_structure (*dict*) – The conditions associated with the new profile, with ‘cidr’, ‘expiration’, and ‘disabled’ members.

Returns

This object.

Return type

ProfileBuilder

set_disabled(flag)

Sets whether or not the new profile is disabled.

Parameters

flag (*bool*) – True if this profile is disabled, False if not.

Returns

This object.

Return type

ProfileBuilder

set_expiration(expiration)

Sets the expiration time on the new profile.

Parameters

expiration (*str*) – The expiration time, specified as ISO 8601.

Returns

This object.

Return type

ProfileBuilder

set_orgs(*orgs_list*)

Set the list of organizations to which the new profile is allowed access.

Parameters

orgs_list (*list*) – List of organization keys or URNs.

Returns

This object.

Return type

ProfileBuilder

set_roles(*roles_list*)

Sets the list of roles associated with the new profile.

Parameters

roles_list (*list*) – A list of role URNs.

Returns

This object.

Return type

ProfileBuilder

classmethod create(*cb, template=None, **kwargs*)

Returns either a new Grant, or a GrantBuilder to begin the process of creating a new grant.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.
- **template** (*dict*) – Optional template to use for creating the grant object.
- **kwargs** (*dict*) – Additional arguments to be used to specify the principal, if template is None.
- **ID.** (The arguments to be used are 'org_key' and 'userid' for the two parts of the) –

Returns

The new grant object, if the template is specified.

GrantBuilder: If template was None, returns a GrantBuilder object. Call methods on it to set up the new grant, and then call build() to create the new grant.

Return type

Grant

Raises

ApiError – If the principal is inadequately specified (whether for the Grant or GrantBuilder).

create_profile(*template=None*)

Returns either a new Profile, or a ProfileBuilder to begin the process of adding a new profile to this grant.

Parameters

template (*dict*) – Optional template to use for creating the profile object.

Returns

If a template was specified, return the new Profile object.

ProfileBuilder: If template was None, returns a ProfileBuilder object. Call methods on it to set up the new profile, and then call build() to create the new profile.

Return type

Profile

delete()

Delete this object.

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

classmethod get_permitted_role_urns(cb)

Returns a list of the URNs of all permitted roles that we can assign to a user.

Parameters

cb ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.

Returns

A list of string role URNs that we are permitted to manage (assign to users).

Return type

list

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

property profiles_

Return the profiles associated with this grant.

Returns

The profiles associated with this grant, each represented as a Profile object.

Return type

list

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

MutableBaseModel

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(*fulltouch=False*)

Force this object to be considered as changed.

validate()

Validates this object.

Returns

True if the object is validated.

Return type

bool

Raises

InvalidObjectError – If the object has missing fields.

class GrantQuery(*doc_class, cb*)

Bases: *BaseQuery*, *IterableQueryMixin*, *AsyncQueryMixin*

Query for retrieving grants in bulk.

Initialize the Query object.

Parameters

- **doc_class** (*class*) – The class of the model this query returns.
- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.

add_principal(*principal_urn, org_urn*)

Add a new principal to the query.

Parameters

- **principal_urn** (*str*) – URN of the principal to search for grants on.
- **org_urn** (*str*) – URN of the organization to which the principal belongs.

Returns

This object.

Return type

GrantQuery

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *[MoreThanOneResultError](#)* – If the query returns more than one item
- *[ObjectNotFoundError](#)* – If the query returns zero items

log = <Logger cbc_sdk.platform.grants (WARNING)>

Grant and Profile Models

normalize_org(*org*)

Internal function to normalize an org reference to a URN.

4.11.9 Jobs Module

Model and Query Classes for Jobs API

class Job(*cb, model_unique_id, initial_data=None*)

Bases: *[NewBaseModel](#)*

Represents a job currently executing in the background.

Parameters

- **connector_id** – Connector ID for the job
- **create_time** – Time this job was created

- **errors** – Errors for the job
- **id** – ID of the job
- **job_parameters** – Parameters that were used for this job
- **last_update_time** – Last time this job was updated
- **org_key** – Organization key of the org this job is being run against
- **owner_id** – ID of the job owner
- **status** – Current job status
- **type** – Type of job this is

Initialize the Job object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*int*) – ID of the job.
- **initial_data** (*dict*) – Initial data used to populate the job.

`await_completion()`

Create a Python Future to check for job completion and return results when available.

Returns a Future object which can be used to await results that are ready to fetch. This function call does not block.

Required Permissions:

`jobs.status(READ)`

Returns

A future which can be used to wait for this job's completion. When complete, the result of the Future will be this object.

Return type

Future

`get(attrname, default_val=None)`

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

`get_output_as_file(filename)`

Export the results from the job, writing the results to the given file.

Required Permissions:

`jobs.status(READ)`

Parameters

filename (*str*) – Name of the file to write the results to.

get_output_as_lines()

Export the results from the job, returning the data as iterated lines of text.

This is only intended for output that can reasonably be represented as lines of text, such as plain text or CSV. If a job outputs structured text like JSON or XML, this method should not be used.

Required Permissions:

jobs.status(READ)

Returns

An iterable that can be used to get each line of text in turn as a string.

Return type

iterable

get_output_as_stream(output)

Export the results from the job, writing the results to the given stream.

Required Permissions:

jobs.status(READ)

Parameters

output (*RawIOBase*) – Stream to write the CSV data from the request to.

get_output_as_string()

Export the results from the job, returning the results as a string.

Required Permissions:

jobs.status(READ)

Returns

The results from the job.

Return type

str

get_progress()

Get and return the current progress information for the job.

Required Permissions:

jobs.status(READ)

Returns

Total number of items to be operated on by this job. int: Total number of items for which operation has been completed. str: Current status message for the job.

Return type

int

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class JobQuery(*doc_class, cb*)

Bases: [BaseQuery](#), [IterableQueryMixin](#), [AsyncQueryMixin](#)

Query for retrieving current jobs.

Initialize the Query object.

Parameters

- **doc_class** (*class*) – The class of the model this query returns.
- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- [MoreThanOneResultError](#) – If the query returns more than one item
- [ObjectNotFoundError](#) – If the query returns zero items

4.11.10 Legacy Alerts Module

Model and Query Classes for Legacy Alerts and Workflows used Alert API v6 and SDK 1.4.3 or earlier

class LegacyAlertSearchQueryCriterionMixin

Bases: [*CriteriaBuilderSupportMixin*](#)

Represents a legacy alert, based on Alert API v6 or SDK 1.4.3 or earlier.

set_alert_ids(*alert_ids*)

Restricts the alerts that this query is performed on to the specified alert IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

alert_ids (*list*) – List of string alert IDs.

Returns

This instance.

Return type

[*AlertSearchQuery*](#)

set_blocked_threat_categories(*categories*)

The field *blocked_threat_category* was deprecated and not included in v7. This method has been removed.

See [Developer Network Alerts v6 Migration](#) for more details.

Args: *categories* (*list*): List of threat categories to look for.

Raises

[*FunctionalityDecommissioned*](#) – If the requested attribute is no longer available.

set_categories(*categories*)

The field *categories* was deprecated and not included in v7. This method has been removed.

In Alerts v7, only records with the type THREAT are returned. Records that in v6 had the category MONITORED (Observed) are now Observations See [Developer Network Alerts v6 Migration](#) for more details.

Parameters

categories (*list*) – List of categories to be restricted to.

Raises

[*FunctionalityDecommissioned*](#) – If the requested attribute is no longer available.

set_cluster_names(*names*)

Restricts the alerts that this query is performed on to the specified Kubernetes cluster names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

names (*list*) – List of Kubernetes cluster names to look for.

Returns

This instance.

Return type

[*ContainerRuntimeAlertSearchQuery*](#)

set_create_time(*args, **kwargs)

Restricts the alerts that this query is performed on to the specified creation time.

The time may either be specified as a start and end point or as a range. In SDK 1.5.0 to align with Alerts v7 API, create_time is set as time_range outside of criteria.

Deprecated:

Use *add_time_criteria(field_name, start, end, range)* instead.

Parameters

- ***args** (*list*) – Not used.
- ****kwargs** (*dict*) – Used to specify start= for start time, end= for end time, and range= for range.

Returns

This instance.

Return type

AlertSearchQuery

set_device_ids(device_ids)

Restricts the alerts that this query is performed on to the specified device IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

device_ids (*list*) – List of integer device IDs.

Returns

This instance.

Return type

AlertSearchQuery

set_device_locations(locations)

Restricts the alerts that this query is performed on to the specified device locations.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

locations (*list*) – List of device locations to look for. Valid values are “ONSITE”, “OFF-SITE”, and “UNKNOWN”.

Returns

This instance.

Return type

CBAnalyticsAlertSearchQuery

set_device_names(device_names)

Restricts the alerts that this query is performed on to the specified device names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

device_names (*list*) – List of string device names.

Returns

This instance.

Return type

AlertSearchQuery

set_device_os(device_os)

Restricts the alerts that this query is performed on to the specified device operating systems.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

device_os (*list*) – List of string operating systems. Valid values are “WINDOWS”, “ANDROID”, “MAC”, “IOS”, “LINUX”, and “OTHER.”

Returns

This instance.

Return type

AlertSearchQuery

set_device_os_versions(device_os_versions)

Restricts the alerts that this query is performed on to the specified device operating system versions.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

device_os_versions (*list*) – List of string operating system versions.

Returns

This instance.

Return type

AlertSearchQuery

set_device_username(users)

Restricts the alerts that this query is performed on to the specified user names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

users (*list*) – List of string user names.

Returns

This instance.

Return type

AlertSearchQuery

set_egress_group_ids(ids)

Restricts the alerts that this query is performed on to the specified egress group IDs.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

ids (*list*) – List of egress group IDs to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_egress_group_names(*names*)

Restricts the alerts that this query is performed on to the specified egress group names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

names (*list*) – List of egress group names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_external_device_friendly_names(*names*)

Restricts the alerts that this query is performed on to the specified external device friendly names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

names (*list*) – List of external device friendly names to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_external_device_ids(*ids*)

Restricts the alerts that this query is performed on to the specified external device IDs.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

ids (*list*) – List of external device IDs to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_group_results(*do_group*)

The field *group_results* was deprecated and not included in v7. This method has been removed.

It previously specified whether to group the results of the query. Use the [Grouped Alerts Operations](#) (#grouped-alerts-operations) instead. See [Developer Network Alerts v6 Migration](#) for more details.

Parameters

do_group (*bool*) – True to group the results, False to not do so.

Raises

[FunctionalityDecommissioned](#) – If the requested attribute is no longer available.

set_ip_reputations(*reputations*)

Restricts the alerts that this query is performed on to the specified IP reputation values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

reputations (*list*) – List of IP reputation values to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_kill_chain_statuses(*statuses*)

The field *kill_chain_status* was deprecated and not included in v7. This method has been removed.

See [Developer Network Alerts v6 Migration](#) for more details.

Args: statuses (list): List of kill chain statuses to look for.

Raises

[FunctionalityDecommissioned](#) – If the requested attribute is no longer available.

set_legacy_alert_ids(*alert_ids*)

Restricts the alerts that this query is performed on to the specified legacy alert IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

alert_ids (*list*) – List of string legacy alert IDs.

Returns

This instance.

Return type

[AlertSearchQuery](#)

set_namespaces(*namespaces*)

Restricts the alerts that this query is performed on to the specified Kubernetes namespaces.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

namespaces (*list*) – List of Kubernetes namespaces to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_not_blocked_threat_categories(*categories*)

The field *not_blocked_threat_category* was deprecated and not included in v7. This method has been removed.

See [Developer Network Alerts v6 Migration](#) for more details.

Args: categories (list): List of threat categories to look for.

Raises

[FunctionalityDecommissioned](#) – If the requested attribute is no longer available.

set_policy_applied(*applied_statuses*)

Restricts the alerts that this query is performed on to the specified policy status values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

applied_statuses (*list*) – List of status values to look for. Valid values are “APPLIED” and “NOT_APPLIED”.

Returns

This instance.

Return type

CBAnalyticsAlertSearchQuery

set_policy_ids(*policy_ids*)

Restricts the alerts that this query is performed on to the specified policy IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

policy_ids (*list*) – List of integer policy IDs.

Returns

This instance.

Return type

[AlertSearchQuery](#)

set_policy_names(*policy_names*)

Restricts the alerts that this query is performed on to the specified policy names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

policy_names (*list*) – List of string policy names.

Returns

This instance.

Return type*AlertSearchQuery***set_ports(*ports*)**

Restricts the alerts that this query is performed on to the specified `netconn_local_ports`.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Note that in SDK 1.5.0, to align with Alerts API v7, the search field was updated from `port` to `netconn_local_port`. It is possible to search on either `netconn_local_port` or `netconn_remote_port` using the `add_criteria(fieldname, [field values])` method.

Parameters

ports (*list*) – List of `netconn_local_ports` to look for.

Returns

This instance.

Return type*ContainerRuntimeAlertSearchQuery***set_process_names(*process_names*)**

Restricts the alerts that this query is performed on to the specified process names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

process_names (*list*) – List of string process names.

Returns

This instance.

Return type*AlertSearchQuery***set_process_sha256(*shas*)**

Restricts the alerts that this query is performed on to the specified process SHA-256 hash values.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

shas (*list*) – List of string process SHA-256 hash values.

Returns

This instance.

Return type*AlertSearchQuery***set_product_ids(*ids*)**

Restricts the alerts that this query is performed on to the specified product IDs.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

ids (*list*) – List of product IDs to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_product_names(*names*)

Restricts the alerts that this query is performed on to the specified product names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

names (*list*) – List of product names to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_protocols(*protocols*)

Restricts the alerts that this query is performed on to the specified protocols.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

protocols (*list*) – List of protocols to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_reason_code(*reason*)

Restricts the alerts that this query is performed on to the specified reason codes (enum values).

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

reason (*list*) – List of string reason codes to look for.

Returns

This instance.

Return type

CBAnalyticsAlertSearchQuery

set_remote_domains(*domains*)

Restricts the alerts that this query is performed on to the specified remote domains.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

domains (*list*) – List of remote domains to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_remote_ips(*addrs*)

Restricts the alerts that this query is performed on to the specified remote IP addresses.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

addrs (*list*) – List of remote IP addresses to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_replica_ids(*ids*)

Restricts the alerts that this query is performed on to the specified pod names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

ids (*list*) – List of pod names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_reputations(*reps*)

Restricts the alerts that this query is performed on to the specified reputation values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

reps (*list*) – List of string reputation values. Valid values are “KNOWN_MALWARE”, “SUSPECT_MALWARE”, “PUP”, “NOT_LISTED”, “ADAPTIVE_WHITE_LIST”, “COMMON_WHITE_LIST”, “TRUSTED_WHITE_LIST”, and “COMPANY_BLACK_LIST”.

Returns

This instance.

Return type

AlertSearchQuery

set_rule_ids(*ids*)

Restricts the alerts that this query is performed on to the specified Kubernetes policy rule IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

In SDK prior to 1.5.0 this was only supported for Container Runtime Alerts so will convert to `k8s_rule_id` in criteria. In SDK 1.5.0 and later, aligned to Alert v7 API, use `add_criteria()` should be used for both `k8s_rule_id` and for other alert types, `rule_id`.

Parameters

ids (*list*) – List of Kubernetes policy rule IDs to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_rule_names(*names*)

Restricts the alerts that this query is performed on to the specified Kubernetes policy rule names.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

names (*list*) – List of Kubernetes policy rule names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_run_states(*states*)

Restricts the alerts that this query is performed on to the specified run states.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

states (*list*) – List of run states to look for. Valid values are “DID_NOT_RUN”, “RAN”, and “UNKNOWN”.

Returns

This instance.

Return type

CBAalyticsAlertSearchQuery

set_sensor_actions(*actions*)

Restricts the alerts that this query is performed on to the specified sensor actions.

Deprecated:

Use `add_criteria(field_name, [field_value])` instead.

Parameters

actions (*list*) – List of sensor actions to look for. Valid values are “POLICY_NOT_APPLIED”, “ALLOW”, “ALLOW_AND_LOG”, “TERMINATE”, and “DENY”.

Returns

This instance.

Return type

CBAalyticsAlertSearchQuery

set_serial_numbers(*serial_numbers*)

Restricts the alerts that this query is performed on to the specified serial numbers.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

serial_numbers (*list*) – List of serial numbers to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_tags(*tags*)

Restricts the alerts that this query is performed on to the specified tag values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

tags (*list*) – List of string tag values.

Returns

This instance.

Return type

AlertSearchQuery

set_target_priorities(*priorities*)

Restricts the alerts that this query is performed on to the specified target priority values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

priorities (*list*) – List of string target priority values. Valid values are “LOW”, “MEDIUM”, “HIGH”, and “MISSION_CRITICAL”.

Returns

This instance.

Return type

AlertSearchQuery

set_threat_cause_vectors(*vectors*)

The field *threat_cause_vector* was deprecated and not included in v7. This method has been removed.

See [Developer Network Alerts v6 Migration](#) for more details.

Parameters

vectors (*list*) – List of threat cause vectors to look for.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_threat_ids(*threats*)

Restricts the alerts that this query is performed on to the specified threat ID values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

threats (*list*) – List of string threat ID values.

Returns

This instance.

Return type

AlertSearchQuery

set_types(*alerttypes*)

Restricts the alerts that this query is performed on to the specified alert type values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

alerttypes (*list*) – List of string alert type values. Valid values are “CB_ANALYTICS”, “WATCHLIST”, “DEVICE_CONTROL”, and “CONTAINER_RUNTIME”. In SDK 1.5.0, to align with Alert API v7, more alert types are available but the *add_criteria* method must be used.

Returns

This instance.

Return type

AlertSearchQuery

Note: - When filtering by fields that take a list parameter, an empty list will be treated as a wildcard and match everything.

set_vendor_ids(*ids*)

Restricts the alerts that this query is performed on to the specified vendor IDs.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

ids (*list*) – List of vendor IDs to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_vendor_names(*names*)

Restricts the alerts that this query is performed on to the specified vendor names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

names (*list*) – List of vendor names to look for.

Returns

This instance.

Return type

DeviceControlAlertSearchQuery

set_watchlist_ids(ids)

Restricts the alerts that this query is performed on to the specified watchlist ID values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

ids (*list*) – List of string watchlist ID values.

Returns

This instance.

Return type

WatchlistAlertSearchQuery

set_watchlist_names(names)

Restricts the alerts that this query is performed on to the specified watchlist name values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

names (*list*) – List of string watchlist name values.

Returns

This instance.

Return type

WatchlistAlertSearchQuery

set_workflows(workflow_vals)

Restricts the alerts that this query is performed on to the specified workflow status values.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

workflow_vals (*list*) – List of string alert type values. Valid values are “OPEN” and “DISMISSED”.

Returns

This instance.

Return type

AlertSearchQuery

set_workload_ids(ids)

The field *workload_id* was deprecated and not included in v7. This method has been removed.

Use *workload_name* instead. See [Developer Network Alerts v6 Migration](#) for more details.

Parameters

ids (*list*) – List of workload IDs to look for.

Raises

FunctionalityDecommissioned – If the requested attribute is no longer available.

set_workload_kinds(*kinds*)

Restricts the alerts that this query is performed on to the specified workload types.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

kinds (*list*) – List of workload types to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

set_workload_names(*names*)

Restricts the alerts that this query is performed on to the specified workload names.

Deprecated:

Use *add_criteria(field_name, [field_value])* instead.

Parameters

names (*list*) – List of workload names to look for.

Returns

This instance.

Return type

ContainerRuntimeAlertSearchQuery

4.11.11 Network Threat Metadata Module

Model Class for NetworkThreatMetadata

```
class NetworkThreatMetadata(cb, model_unique_id=None, initial_data=None, force_init=False,  
                             full_doc=True)
```

Bases: *NewBaseModel*

Represents a NetworkThreatMetadata

Parameters

- **detector_abstract** – Abstract or description of the detector
- **detector_goal** – Description of what the detector is achieving
- **false_negatives** – Highlights why detector could not have been triggered
- **false_positives** – Highlights why detector could have been triggered
- **threat_public_comment** – Public comment of the threat

Initialize the NetworkThreatMetadata object.

Required Permissions:

org.xdr.metadata (READ)

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **model_unique_id** (*Any*) – The unique ID for this particular instance of the model object.
- **initial_data** (*dict*) – Not used, retained for compatibility.
- **force_init** (*bool*) – False to not force object initialization.
- **full_doc** (*bool*) – True to mark the object as fully initialized.

Raises

[ApiError](#) – if model_unique_id is not provided

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

4.11.12 Observations Module

Model and Query Classes for Observations

class Observation(*cb*, *model_unique_id=None*, *initial_data=None*, *force_init=False*, *full_doc=False*)

Bases: [NewBaseModel](#)

Represents an Observation

Initialize the Observation object.

Required Permissions:

org.search.events (READ)

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.

- **model_unique_id** (*Any*) – The unique ID for this particular instance of the model object.
- **initial_data** (*dict*) – The data to use when initializing the model object.
- **force_init** (*bool*) – True to force object initialization.
- **full_doc** (*bool*) – False to mark the object as not fully initialized.

static bulk_get_details(*cb, alert_id=None, observation_ids=None, timeout=0*)

Bulk get details

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **alert_id** (*str*) – An alert id to fetch associated observations
- **observation_ids** (*list*) – A list of observation ids to fetch
- **timeout** (*int*) – Observations details request timeout in milliseconds. This may never be greater than the configured default timeout. If this value is 0, the configured default timeout is used.

Returns

list of Observations

Return type

list

Raises

[ApiError](#) – if cb is not instance of CBCloudAPI

deobfuscate_cmdline()

Deobfuscates the command line of the process pointed to by the observation and returns the deobfuscated result.

Required Permissions:

script.deobfuscation(EXECUTE)

Returns

A dict containing information about the obfuscated command line, including the deobfuscated result.

Return type

dict

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_details(*timeout=0, async_mode=False*)

Requests detailed results.

Parameters

- **timeout** (*int*) – Observations details request timeout in milliseconds. This may never be greater than the configured default timeout. If this value is 0, the configured default timeout is used.
- **async_mode** (*bool*) – True to request details in an asynchronous manner.

Returns

Observation object enriched with the details fields

Return type

Observation

Note:

- When using asynchronous mode, this method returns a python future. You can call `result()` on the future object to wait for completion and get the results.
-

Examples

```
>>> observation = api.select(Observation, observation_id)
>>> observation.get_details()
```

```
>>> observations = api.select(Observation).where(process_pid=2000)
>>> observations[0].get_details()
```

get_network_threat_metadata()

Requests Network Threat Metadata.

Returns

Get the metadata for a given detector (rule).

Return type

NetworkThreatMetadata

Raises

ApiError – when `rule_id` is not returned for the `Observation`

Examples

```
>>> observation = api.select(Observation, observation_id)
>>> threat_metadata = observation.get_network_threat_metadata()
```

refresh()

Reload this object from the server.

static search_suggestions(*cb, query, count=None*)

Returns suggestions for keys and field values that can be used in a search.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **query** (*str*) – A search query to use.
- **count** (*int*) – (optional) Number of suggestions to be returned

Returns

A list of search suggestions expressed as dict objects.

Return type

list

Raises

[ApiError](#) – if cb is not instance of CBCloudAPI

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class ObservationFacet(*cb, model_unique_id, initial_data*)

Bases: [UnrefreshableModel](#)

Represents an observation facet retrieved.

Parameters

- **terms** – Contains the Observations Facet search results
- **ranges** – Groupings for search result properties that are ISO 8601 timestamps or numbers
- **contacted** – The number of searchers contacted for this query
- **completed** – The number of searchers that have reported their results

Initialize the Terms object with initial data.

class Ranges(*cb, initial_data*)

Bases: [UnrefreshableModel](#)

Represents the range (bucketed) facet fields and values associated with an Observation Facet query.

Initialize an ObservationFacet Ranges object with initial_data.

property facets

Returns the reified *ObservationFacet.Terms._facets* for this result.

property fields

Returns the ranges fields for this result.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class Terms(*cb, initial_data*)

Bases: [*UnrefreshableModel*](#)

Represents the facet fields and values associated with an Observation Facet query.

Initialize an ObservationFacet Terms object with initial_data.

property facets

Returns the terms' facets for this result.

property fields

Returns the terms facets' fields for this result.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

property ranges_

Returns the reified *ObservationFacet.Ranges* for this result.

refresh()

Reload this object from the server.

property terms_

Returns the reified *ObservationFacet.Terms* for this result.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class **ObservationGroup**(*cb, initial_data=None*)

Bases: `object`

Represents ObservationGroup

Initialize ObservationGroup object

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **initial_data** (*dict*) – The data to use when initializing the model object.

Notes

The constructed object will have the following data: - group_start_timestamp - group_end_timestamp - group_key - group_value

class **ObservationQuery**(*doc_class, cb*)

Bases: [Query](#)

Represents the query logic for an Observation query.

This class specializes *Query* to handle the particulars of observations querying.

Initialize the ObservationQuery object.

Parameters

- **doc_class** (*class*) – The class of the model this query returns.
- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

add_exclusions(key, newlist)

Add to the exclusions on this query with a custom exclusions key.

Will overwrite any existing exclusion for the specified key.

Parameters

- **key** (str) – The key for the exclusion item to be set.
- **newlist** (str or list[str]) – Value or list of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).add_exclusions("type", ["WATCHLIST"])
>>> query = api.select(Alert).add_exclusions("type", "WATCHLIST")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(q=None, **kwargs)

Add a conjunctive filter to this query.

Parameters

- **q** (Any) – Query string or *solrq.Q* object
- ****kwargs** (dict) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

batch_size(new_batch_size)

Set the batch size of the paginated query.

Parameters

new_batch_size (int) – The new batch size.

Returns

A new query with the updated batch size.

Return type

PaginatedQuery

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

get_group_results(*fields*, *max_events_per_group*=None, *rows*=500, *start*=None, *range_duration*=None, *range_field*=None, *range_method*=None)

Get group results grouped by provided fields.

Parameters

- **fields** (*str* / *list*) – field or fields by which to perform the grouping
- **max_events_per_group** (*int*) – Maximum number of events in a group, if not provided, all events will be returned
- **rows** (*int*) – Number of rows to request, can be paginated
- **start** (*int*) – First row to use for pagination
- **ranges** (*dict*) – dict with information about duration, field, method

Returns

grouped results

Return type

dict

Examples

```
>>> for group in api.select(Observation).where(process_pid=2000).get_group_
    ↪results("device_name"):
>>>     ...
```

not_(*q*=None, ***kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(kwargs)**

or_() criteria are explicitly provided to Observation queries.

This method overrides the base class in order to provide *or_()* functionality rather than raising an exception.

set_fields(fields)

Sets the fields to be returned with the response.

Parameters

fields (*str* or *list[str]*) – Field or list of fields to be returned.

set_rows(rows)

Sets the ‘rows’ query body parameter to the ‘start search’ API call, determining how many rows to request.

Parameters

rows (*int*) – How many rows to request.

Returns

ObservationQuery object

Return type

Query

Example

```
>>> cb.select(Observation).where(process_name="foo.exe").set_rows(50)
```

set_start(start)

Sets the ‘start’ query body parameter, determining where to begin retrieving results from.

Parameters

start (*int*) – Where to start results from.

set_time_range(start=None, end=None, window=None)

Sets the ‘time_range’ query body parameter, determining a time window based on ‘device_timestamp’.

Parameters

- **start** (*str* in *ISO 8601 timestamp*) – When to start the result search.
- **end** (*str* in *ISO 8601 timestamp*) – When to end the result search.
- **window** (*str*) – Time window to execute the result search, ending on the current time. Should be in the form “-2w”, where y=year, w=week, d=day, h=hour, m=minute, s=second.

Note:

- *window* will take precedent over *start* and *end* if provided.

Examples

```
>>> query = api.select(Process).set_time_range(start="2020-10-20T20:34:07Z").
↳where("query is required")
>>> second_query = api.select(Process).
...     set_time_range(start="2020-10-20T20:34:07Z", end="2020-10-30T20:34:07Z
↳").where("query is required")
>>> third_query = api.select(Process).set_time_range(window='-3d').where("query_
↳is required")
```

sort_by(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

The query with sorting parameters.

Return type

Query

Example

```
>>> cb.select(Process).where(process_name="cmd.exe").sort_by("device_timestamp")
```

timeout(*msecs*)

Sets the timeout on a observation query.

Parameters

msecs (*int*) – Timeout duration, in milliseconds. This may never be greater than the configured default timeout. If this value is 0, the configured default timeout is used.

Returns

The Query object with new milliseconds parameter.

Return type

Query (ObservationQuery)

Example

```
>>> cb.select(Observation).where(process_name="foo.exe").timeout(5000)
```

update_criteria(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.

- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

update_exclusions(*key*, *newlist*)

Update the exclusion on this query with a custom exclusion key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*list*) – List of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).update_exclusions("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None*, ***kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, `QueryBuilder`, or `solrq.Q` object
- ****kwargs** (*dict*) – Arguments to construct a `solrq.Q` with

Returns

This Query object.

Return type

Query

4.11.13 Policies Module

Policy implementation as part of Platform API

class Policy(*cb, model_unique_id=None, initial_data=None, force_init=False, full_doc=False*)

Bases: [MutableBaseModel](#)

Represents a policy within the organization.

Create one of these objects (either directly or with the `CBCloudAPI.create()` method) and set its properties, then call its `save()` method to create the policy on the server. This requires the `org.policies(CREATE)` permission.

Alternatively, you may call `Policy.create()` to get a `PolicyBuilder`, use its methods to set the properties of the new policy, call its `build()` method to build the populated `Policy`, then call the policy `save()` method.

To update a `Policy`, change the values of its property fields, then call the policy's `save()` method. This requires the `org.policies(UPDATE)` permission.

To delete an existing `Policy`, call its `delete()` method. This requires the `org.policies(DELETE)` permission.

For information on values for policy settings including enumeration values, see the Policy Service API page: <https://developer.carbonblack.com/reference/carbon-black-cloud/platform/latest/policy-service/#fields>

Parameters

- **id** – The policy identifier
- **name** – Defined name for the policy
- **org_key** – The organization key associated with the console instance
- **priority_level** – The priority level designated for policy
- **position** – Relative priority of this policy within the organization. Lower values indicate higher priority.
- **is_system** – Indicates that the policy was created by VMware
- **description** – The description of the policy
- **auto_deregister_inactive_vdi_interval_ms** – The time in milliseconds to wait after a VDI is inactive before setting the VDI to a DEREGISTERED state
- **auto_delete_known_bad_hashes_delay** – Enables the Carbon Black Cloud to automatically delete known malware after a specified time in milliseconds
- **av_settings** – Anti-Virus settings for endpoints and workloads assigned to the policy
- **rules** – Permission or prevention rules
- **directory_action_rules** – Rules to deny or allow the deployed sensors to send uploads from specific paths
- **sensor_settings** – Settings to configure sensor behavior and capabilities
- **managed_detection_response_permissions** – Permissions for Managed Detection and Response analysts to perform remediations on endpoints and workloads assigned to the policy
- **version** – Version of the policy

Initialize the Policy object.

Required Permissions:

org.policies (READ)

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*int*) – ID of the policy.
- **initial_data** (*dict*) – Initial data used to populate the policy.
- **force_init** (*bool*) – If True, forces the object to be refreshed after constructing. Default False.
- **full_doc** (*bool*) – If True, object is considered “fully” initialized. Default False.

class PolicyBuilder(*cb*)

Bases: object

Builder object to simplify the creation of new Policy objects.

To use, call `Policy.create()` to get a `PolicyBuilder`, use its methods to set the properties of the new policy, call its `build()` method to build the populated Policy, then call the `policy save()` method. The `org.policy(CREATE)` permission is required.

Examples

```
>>> builder = Policy.create(api)
>>> builder.set_name("New Policy").set_priority("MEDIUM").set_description("New_
↳ policy description")
>>> # more calls here to set up rules, sensor settings, etc.
>>> policy = builder.build()
>>> policy.save()
```

Initialize the PolicyBuilder object.

Parameters

cb ([BaseAPI](#)) – Reference to API object used to communicate with the server.

add_directory_action_rule(*path, file_upload, protection*)

Add a directory action rule to the new policy.

Parameters

- **path** (*str*) – Path to the file or directory.
- **file_upload** (*bool*) – True to allow the deployed sensor to upload from that path.
- **protection** (*bool*) – True to deny the deployed sensor to upload from that path.

Returns

This object.

Return type

[PolicyBuilder](#)

add_rule(*app_type, app_value, operation, action, required=True*)

Add a new rule as discrete data elements to the new policy.

Parameters

- **app_type** (*str*) – Specifies “NAME_PATH”, “SIGNED_BY”, or “REPUTATION”.
- **app_value** (*str*) – Value of the attribute specified by *app_type* to be matched.
- **operation** (*str*) – The type of behavior the application is performing.

- **action** (*str*) – The action the sensor will take when the application performs the specified action.
- **required** (*bool*) – True if this rule is required, False if not.

Returns

This object.

Return type

PolicyBuilder

Raises

InvalidObjectError – If the rule data passed in is not valid.

add_rule_config(*config_id*, *name*, *category*, ***kwargs*)

Add a new rule configuration as discrete data elements to the new policy.

Parameters

- **config_id** (*str*) – ID of the rule configuration object (a GUID).
- **name** (*str*) – Name of the rule configuration object.
- **category** (*str*) – Category of the rule configuration object.
- ****kwargs** (*dict*) – Parameter values for the rule configuration object.

Returns

This object.

Return type

PolicyBuilder

Raises

InvalidObjectError – If the rule configuration data passed in is not valid.

add_rule_config_copy(*rule_config*)

Adds a copy of an existing rule configuration to this new policy.

Parameters

rule_config (*PolicyRuleConfig*) – The rule configuration to copy and add to this object.

Returns

This object.

Return type

PolicyBuilder

Raises

InvalidObjectError – If the rule configuration data passed in is not valid.

add_rule_copy(*rule*)

Adds a copy of an existing rule to this new policy.

Parameters

rule (*PolicyRule*) – The rule to copy and add to this object.

Returns

This object.

Return type

PolicyBuilder

Raises

InvalidObjectError – If the rule data passed in is not valid.

add_sensor_setting(*name*, *value*)

Add a sensor setting to the policy.

Parameters

- **name** (*str*) – Sensor setting name.
- **value** (*str*) – Sensor setting value.

Returns

This object.

Return type*PolicyBuilder***Raises***ApiError* – If the sensor setting name is not a valid one.**build()**

Build a new Policy object using the contents of this builder.

The new policy must have *save()* called on it to be saved to the server.

Returns

The new Policy object.

Return type*Policy***set_auto_delete_bad_hash_delay(*delay*)**

Set the delay in milliseconds after which known malware will be deleted.

Parameters

delay (*int*) – The desired delay interval in milliseconds.

Returns

This object.

Return type*PolicyBuilder***set_auto_deregister_interval(*interval*)**

Set the time in milliseconds after a VDI goes inactive to deregister it.

Parameters

interval (*int*) – The desired interval in milliseconds.

Returns

This object.

Return type*PolicyBuilder***set_avira_protection_cloud(*enabled*, *max_exe_delay*=None, *max_file_size*=None, *risk_level*=None)**

Set the settings for third-party unknown binary reputation analysis.

Parameters

- **enabled** (*bool*) – True to enable unknown binary reputation analysis.
- **max_exe_delay** (*int*) – Time before sending unknown binary for analysis, in seconds.
- **max_file_size** (*int*) – Maximum size of file to send for analysis, in megabytes.
- **risk_level** (*int*) – Risk level to send for analysis (0-7).

Returns

This object.

Return type*PolicyBuilder***set_description(*descr*)**

Set the new policy description.

Parameters

descr (*str*) – The new policy description.

Returns

This object.

Return type*PolicyBuilder***set_managed_detection_response_permissions(*policy_mod*, *quarantine*)**

Set the permissions for managed detection and response.

Parameters

- **policy_mod** (*bool*) – True to allow MDR team to modify the policy.
- **quarantine** (*bool*) – True to allow MDR team to quarantine endpoints/workloads associated with the policy.

Returns

This object.

Return type

PolicyBuilder

set_name(*name*)

Set the new policy name.

Parameters

name (*str*) – The new policy name.

Returns

This object.

Return type

PolicyBuilder

set_on_access_scan(*enabled, mode='NORMAL'*)

Sets the local scan settings.

Parameters

- **enabled** (*bool*) – True to enable local scan.
- **mode** (*str*) – The mode to operate in, either “NORMAL” or “AGGRESSIVE”.

Returns

This object.

Return type

PolicyBuilder

Raises

ApiError – If an invalid value is passed for the “mode” parameter.

set_on_demand_scan(*enabled, profile='NORMAL', scan_usb='AUTOSCAN', scan_cd_dvd='AUTOSCAN'*)

Sets the background scan settings.

Parameters

- **enabled** (*bool*) – True to enable background scan.
- **profile** (*str*) – The background scan mode, either “NORMAL” or “AGGRESSIVE”.
- **scan_usb** (*str*) – Either “AUTOSCAN” to scan USB devices, or “DISABLED” to not do so.
- **scan_cd_dvd** (*str*) – Either “AUTOSCAN” to scan CDs and DVDs, or “DISABLED” to not do so.

Returns

This object.

Return type

PolicyBuilder

Raises

ApiError – If an invalid value is passed for any parameter.

set_on_demand_scan_schedule(*days, start_hour, range_hours, recover_if_missed=True*)

Sets the schedule for when background scans will be performed.

Parameters

- **days** (*list[str]*) – The days on which to perform background scans.
- **start_hour** (*int*) – The hour of the day at which to perform the scans.
- **range_hours** (*int*) – The range of hours over which to perform the scans.
- **recover_if_missed** (*bool*) – True if the background scan should be performed ASAP if it’s been missed.

Returns

This object.

Return type

PolicyBuilder

Raises

ApiError – If an invalid value is passed for a day of the week.

set_priority(priority)

Set the new policy's priority. Default is MEDIUM.

Parameters

priority (*str*) – The priority, either “LOW”, “MEDIUM”, “HIGH”, or “MISSION_CRITICAL”.

Returns

This object.

Return type

PolicyBuilder

Raises

ApiError – If an invalid priority value is passed in.

set_signature_update(enabled)

Set the enable status for signature updates.

Parameters

enabled (*bool*) – True to enable signature updates.

Returns

This object.

Return type

PolicyBuilder

set_signature_update_schedule(full_interval_hours, initial_random_delay_hours, interval_hours)

Set the signature update schedule.

Parameters

- **full_interval_hours** (*int*) – The interval in hours between signature updates.
- **initial_random_delay_hours** (*int*) – The initial delay in hours before the first signature update.
- **interval_hours** (*int*) – The interval in hours between signature updates.

Returns

This object.

Return type

PolicyBuilder

set_update_servers_offsite(names)

Sets the list of update servers for offsite devices.

Parameters

names (*list[str]*) – The list of update servers, as URIs.

Returns

This object.

Return type

PolicyBuilder

set_update_servers_onsite(names, preferred_servers=None)

Sets the list of update servers for internal devices.

Parameters

- **names** (*list[str]*) – The list of available update servers, as URIs.
- **preferred_servers** (*list[str]*) – The list of update servers to be considered “preferred,” as URIs.

Returns

This object.

Return type

PolicyBuilder

set_update_servers_override(names)

Sets the list of update servers to override offsite/onsite settings.

Parameters

names (*list[str]*) – The server names to use, as a list of URIs.

Returns

This object.

Return type

PolicyBuilder

add_rule(new_rule)

Adds a rule to this Policy.

Parameters

new_rule (*dict(str, str)*) – The new rule to add to this Policy.

Notes

- The new rule must conform to this dictionary format:

```
{“action”: “ACTION”, “application”: {“type”: “TYPE”, “value”: “VALUE”}, “operation”:
“OPERATION”, “required”: “REQUIRED”}
```

- The dictionary keys have these possible values:

“action”: [“IGNORE”, “ALLOW”, “DENY”, “TERMINATE_PROCESS”, “TERMINATE_THREAD”, “TERMINATE”]

“type”: [“NAME_PATH”, “SIGNED_BY”, “REPUTATION”]

“value”: Any string value to match on

“operation”: [“BYPASS_ALL”, “INVOKE_SCRIPT”, “INVOKE_SYSAPP”, “POL_INVOKE_NOT_TRUSTED”, “INVOKE_CMD_INTERPRETER”, “RANSOM”, “NETWORK”, “PROCESS_ISOLATION”, “CODE_INJECTION”, “MEMORY_SCRAPE”, “RUN_INMEMORY_CODE”, “ESCALATE”, “RUN”]

“required”: [True, False]

property core_prevention_rule_configs

Returns a dictionary of core prevention rule configuration IDs and objects for this Policy.

Returns

A dictionary with core prevention rule configuration IDs as keys and **CorePreventionRuleConfig** objects as values.

Return type

dict

property core_prevention_rule_configs_list

Returns a list of core prevention rule configuration objects for this Policy.

Returns

A list of **CorePreventionRuleConfig** objects.

Return type

list

classmethod `create(cb)`

Begins creating a policy by returning a PolicyBuilder.

Parameters

cb ([BaseAPI](#)) – Reference to API object used to communicate with the server.

Returns

The new policy builder object.

Return type[*PolicyBuilder*](#)**property** `data_collection_rule_configs`

Returns a dictionary of data collection rule configuration IDs and objects for this Policy.

Returns

A dictionary with data collection rule configuration IDs as keys and **DataCollectionRuleConfig** objects as values.

Return type

dict

property `data_collection_rule_configs_list`

Returns a list of data collection rule configuration objects for this Policy.

Returns

A list of **DataCollectionRuleConfig** objects.

Return type

list

delete()

Delete this object.

delete_rule(rule_id)

Deletes a rule from this Policy.

Parameters

rule_id (*int*) – The ID of the rule to be deleted.

Raises

[*ApiError*](#) – If the rule ID does not exist in this policy.

delete_rule_config(rule_config_id)

Deletes a rule configuration from this Policy.

Parameters

rule_config_id (*str*) – The ID of the rule configuration to be deleted.

Raises

[*ApiError*](#) – If the rule configuration ID does not exist in this policy.

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.

- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_ruleconfig_parameter_schema(*ruleconfig_id*)

Returns the parameter schema for a specified rule configuration.

Uses cached rule configuration presentation data if present.

Parameters

ruleconfig_id (*str*) – The rule configuration ID (UUID).

Returns

The parameter schema for this particular rule configuration (a JSON schema).

Return type

dict

Raises

InvalidObjectError – If the rule configuration ID is not valid.

property host_based_firewall_rule_config

Returns the host-based firewall rule configuration for this policy.

Returns

The host-based firewall rule configuration, or None.

Return type

HostBasedFirewallRuleConfig

Raises

InvalidObjectError – If there's more than one host-based firewall rule configuration (should not happen).

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

property latestRevision

Returns the latest revision of this policy (compatibility method).

property object_rule_configs

Returns a dictionary of rule configuration IDs and objects for this Policy.

Returns

A dictionary with rule configuration IDs as keys and PolicyRuleConfig objects as values.

Return type

dict

property object_rule_configs_list

Returns a list of rule configuration objects for this Policy.

Returns

A list of PolicyRuleConfig objects.

Return type

list

property object_rules

Returns a dictionary of rule objects and rule IDs for this Policy.

Returns

A dictionary with rule IDs as keys and PolicyRule objects as values.

Return type

dict

property policy

Returns the contents of this policy [compatibility method].

preview_add_policy_override(devices)

Previews changes to the effective policies for devices which result from setting this policy override on them.

Required Permissions:

org.policies (READ)

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **devices** (*list*) – The devices which will have their policies overridden. Each entry in this list is either an integer device ID or a Device object.

Returns

A list of DevicePolicyChangePreview objects representing the assets that change which policy is effective as the result of this operation.

Return type

list[[DevicePolicyChangePreview](#)]

classmethod preview_policy_rank_changes(cb, changes_list)

Previews changes in the ranking of policies, and determines how this will affect asset groups.

Example:

```
>>> cb = CBCloudAPI(profile='sample')
>>> changes = Policy.preview_policy_rank_changes(cb, [(667251, 1)])
>>> # also: changes = Policy.preview_policy_rank_changes(cb, [{"id": 667251,
↪ "position": 1}])
>>> len(changes)
2
>>> changes[0].current_policy_id
660578
>>> changes[0].new_policy_id
667251
```

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.

- **changes_list** (*list*) – The list of proposed changes in the ranking of policies. Each change may be in the form of a dict, in which case the “id” and “position” members are used to designate the policy ID and the new position, or in the form of a list or tuple, in which case the first element specifies the policy ID, and the second element specifies the new position. In all cases, “position” values are limited to values in the range [1..N_], where N_ is the total number of policies in the organization.

Returns

A list of objects containing data previewing the policy changes.

Return type

list[[DevicePolicyChangePreview](#)]

preview_rank_change(*new_rank*)

Previews a change in the ranking of this policy, and determines how this will affect asset groups.

Parameters

new_rank (*int*) – The new rank to give this policy. Ranks are limited to values in the range [1..N_], where N_ is the total number of policies in the organization.

Returns

A list of objects containing data previewing the policy changes.

Return type

list[[DevicePolicyChangePreview](#)]

property priorityLevel

Returns the priority level of this policy (compatibility method).

refresh()

Reload this object from the server.

replace_rule(*rule_id*, *new_rule*)

Replaces a rule in this policy.

Parameters

- **rule_id** (*int*) – The ID of the rule to be replaced.
- **new_rule** (*dict*) – The data for the new rule.

Raises

[ApiError](#) – If the rule ID does not exist in this policy.

replace_rule_config(*rule_config_id*, *new_rule_config*)

Replaces a rule configuration in this policy.

Parameters

- **rule_config_id** (*str*) – The ID of the rule configuration to be replaced.
- **new_rule_config** (*dict*) – The data for the new rule configuration.

Raises

[ApiError](#) – If the rule configuration ID does not exist in this policy.

reset()

Undo any changes made to this object’s fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

MutableBaseModel

set_auth_event_collection(flag)

Sets auth event collection to be enabled or disabled on this policy.

Parameters

flag (*bool*) – True to enable auth event data collection, False to disable it.

Raises

ApiError – If the parameter setting operation failed.

set_data_collection(parameter, value)

Sets a data collection parameter value on any data collection rule configurations in the policy that have it.

As a safety check, this method also validates that the type of the existing value of that parameter is the same as the type of the new value we want to set for that parameter.

Parameters

- **parameter** (*str*) – The name of the parameter to set.
- **value** (*Any*) – The value of the parameter to set.

Raises

ApiError – If the parameter setting operation failed.

set_xdr_collection(flag)

Sets XDR collection to be enabled or disabled on this policy.

Parameters

flag (*bool*) – True to enable XDR data collection, False to disable it.

Raises

ApiError – If the parameter setting operation failed.

property systemPolicy

Returns whether or not this is a system policy (compatibility method).

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(fulltouch=False)

Force this object to be considered as changed.

valid_rule_configs()

Returns a dictionary identifying all valid rule configurations for this policy.

Returns

A dictionary mapping string ID values (UUIDs) to dicts containing entries for name, description, and category.

Return type
dict

validate()

Validates this object.

Returns
True if the object is validated.

Return type
bool

Raises
[InvalidObjectError](#) – If the object has missing fields.

class PolicyQuery(*doc_class, cb*)

Bases: *[BaseQuery](#)*, *[IterableQueryMixin](#)*, *[AsyncQueryMixin](#)*

Query for retrieving policies (summary info only).

Initialize the Query object.

Parameters

- **doc_class** (*class*) – The class of the model this query returns.
- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.

add_descriptions(*descrs*)

Add policy description(s) to the list to search for.

Parameters
descrs (*str/list*) – Either a single policy description or a list of descriptions.

Returns
This object instance.

Return type
[PolicyQuery](#)

Raises
[ApiError](#) – If not supplied with a string or a list of strings.

add_names(*names*)

Add policy name(s) to the list to search for.

Parameters
names (*str/list*) – Either a single policy name or a list of names.

Returns
This object instance.

Return type
[PolicyQuery](#)

Raises
[ApiError](#) – If not supplied with a string or a list of strings.

add_policy_ids(*ids*)

Add policy ID(s) to the list to search for.

Parameters

ids (*int/list*) – Either a single policy ID or a list of IDs.

Returns

This object instance.

Return type

PolicyQuery

Raises

ApiError – If not supplied with an int or a list of ints.

add_priorities(*priorities*)

Add policy priority/priorities to the list to search for.

Parameters

priorities (*str/list*) – Either a single policy priority value or a list of priority values.

Returns

This object instance.

Return type

PolicyQuery

Raises

ApiError – If not supplied with a string priority value or a list of string priority values.

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

set_system(system)

Set to look for either system or non-system policies.

Parameters

system (*bool*) – True to look for system policies, False to look for non-system policies.

Returns

This object instance.

Return type*PolicyQuery***Raises**

ApiError – If not supplied with a Boolean.

class PolicyRule(*cb, parent, model_unique_id=None, initial_data=None, force_init=False, full_doc=False*)

Bases: *MutableBaseModel*

Represents a rule in the policy.

Create one of these objects, associating it with a Policy, and set its properties, then call its save() method to add the rule to the policy. This requires the org.policies(UPDATE) permission.

To update a PolicyRule, change the values of its property fields, then call the rule's save() method. This requires the org.policies(UPDATE) permission.

To delete an existing PolicyRule, call its delete() method. This requires the org.policies(UPDATE) permission.

Parameters

- **id** – The identifier of the rule
- **action** – The action the sensor will take when an application attempts to perform the selected operation
- **application** – The path, signature or reputation of the application
- **operation** – The type of behavior an application is performing

Initialize the PolicyRule object.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **parent** (*Policy*) – The “parent” policy of this rule.
- **model_unique_id** (*int*) – ID of the rule.
- **initial_data** (*dict*) – Initial data used to populate the rule.
- **force_init** (*bool*) – If True, forces the object to be refreshed after constructing. Default False.
- **full_doc** (*bool*) – If True, object is considered “fully” initialized. Default False.

delete()

Delete this object.

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

property is_deleted

Returns True if this rule object has been deleted.

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

MutableBaseModel

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(fulltouch=False)

Force this object to be considered as changed.

validate()

Validates this rule against its constraints.

Raises

InvalidObjectError – If the rule object is not valid.

4.11.14 RuleConfigs Module

Policy rule configuration implementation as part of Platform API

class CorePreventionRuleConfig(*cb, parent, model_unique_id=None, initial_data=None, force_init=False, full_doc=False*)

Bases: *PolicyRuleConfig*

Represents a core prevention rule configuration in the policy.

Create one of these objects, associating it with a Policy, and set its properties, then call its `save()` method to add the rule configuration to the policy. This requires the `org.policies(UPDATE)` permission.

To update a `CorePreventionRuleConfig`, change the values of its property fields, then call its `save()` method. This requires the `org.policies(UPDATE)` permission.

To delete an existing `CorePreventionRuleConfig`, call its `delete()` method. This requires the `org.policies(DELETE)` permission.

Parameters

- **id** – The ID of this rule config
- **name** – The name of this rule config
- **description** – The description of this rule config
- **inherited_from** – Indicates where the rule config was inherited from
- **category** – The category for this rule config
- **parameters** – The parameters associated with this rule config

Initialize the `CorePreventionRuleConfig` object.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **parent** (*Policy*) – The “parent” policy of this rule configuration.
- **model_unique_id** (*str*) – ID of the rule configuration.
- **initial_data** (*dict*) – Initial data used to populate the rule configuration.
- **force_init** (*bool*) – If True, forces the object to be refreshed after constructing. Default False.
- **full_doc** (*bool*) – If True, object is considered “fully” initialized. Default False.

`delete()`

Delete this object.

`get(attrname, default_val=None)`

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_assignment_mode()

Returns the assignment mode of this core prevention rule configuration.

Returns

The assignment mode, either “REPORT” or “BLOCK”.

Return type

str

get_parameter(name, default_value=None)

Returns a parameter value from the rule configuration.

Parameters

- **name** (str) – The parameter name.
- **default_value** (Any) – The default value to return if there’s no parameter by that name. Default is None.

Returns

The parameter value, or None if there is no value.

Return type

Any

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

property parameter_names

Returns a list of parameter names in this rule configuration.

Returns

A list of parameter names in this rule configuration.

Return type

list[str]

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object’s fields.

save()

Save any changes made to this object’s fields.

Returns

This object.

Return type*MutableBaseModel*

set_assignment_mode(mode)

Sets the assignment mode of this core prevention rule configuration.

Parameters

mode (*str*) – The new mode to set, either “REPORT” or “BLOCK”. The default is “BLOCK”.

set_parameter(name, value)

Sets a parameter value into the rule configuration.

Parameters

- **name** (*str*) – The parameter name.
- **value** (*Any*) – The new value to be set.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(fulltouch=False)

Force this object to be considered as changed.

validate()

Validates this rule configuration against its constraints.

Raises

[*InvalidObjectError*](#) – If the rule object is not valid.

class DataCollectionRuleConfig(*cb, parent, model_unique_id=None, initial_data=None, force_init=False, full_doc=False*)

Bases: [*PolicyRuleConfig*](#)

Represents a data collection rule configuration in the policy.

Create one of these objects, associating it with a Policy, and set its properties, then call its save() method to add the rule configuration to the policy. This requires the org.policies(UPDATE) permission.

To update a DataCollectionRuleConfig, change the values of its property fields, then call its save() method. This requires the org.policies(UPDATE) permission.

To delete an existing CorePreventionRuleConfig, call its delete() method. This requires the org.policies(DELETE) permission.

Parameters

- **id** – The ID of this rule config
- **name** – The name of this rule config
- **description** – The description of this rule config
- **inherited_from** – Indicates where the rule config was inherited from
- **category** – The category for this rule config
- **parameters** – The parameters associated with this rule config

Initialize the CorePreventionRuleConfig object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **parent** ([Policy](#)) – The “parent” policy of this rule configuration.
- **model_unique_id** (*str*) – ID of the rule configuration.
- **initial_data** (*dict*) – Initial data used to populate the rule configuration.
- **force_init** (*bool*) – If True, forces the object to be refreshed after constructing. Default False.
- **full_doc** (*bool*) – If True, object is considered “fully” initialized. Default False.

delete()

Delete this object.

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_parameter(name, default_value=None)

Returns a parameter value from the rule configuration.

Parameters

- **name** (*str*) – The parameter name.
- **default_value** (*Any*) – The default value to return if there’s no parameter by that name. Default is None.

Returns

The parameter value, or None if there is no value.

Return type

Any

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

property parameter_names

Returns a list of parameter names in this rule configuration.

Returns

A list of parameter names in this rule configuration.

Return type

list[str]

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type*MutableBaseModel***set_parameter(name, value)**

Sets a parameter value into the rule configuration.

Parameters

- **name** (str) – The parameter name.
- **value** (Any) – The new value to be set.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(fulltouch=False)

Force this object to be considered as changed.

validate()

Validates this rule configuration against its constraints.

Raises

InvalidObjectError – If the rule object is not valid.

```
class HostBasedFirewallRuleConfig(cb, parent, model_unique_id=None, initial_data=None,
                                   force_init=False, full_doc=False)
```

Bases: *PolicyRuleConfig*

Represents a host-based firewall rule configuration in the policy.

Parameters

- **id** – The ID of this rule config
- **name** – The name of this rule config
- **description** – The description of this rule config
- **inherited_from** – Indicates where the rule config was inherited from
- **category** – The category for this rule config
- **parameters** – The parameters associated with this rule config

Initialize the CorePreventionRuleConfig object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **parent** ([Policy](#)) – The “parent” policy of this rule configuration.
- **model_unique_id** (*str*) – ID of the rule configuration.
- **initial_data** (*dict*) – Initial data used to populate the rule configuration.
- **force_init** (*bool*) – If True, forces the object to be refreshed after constructing. Default False.
- **full_doc** (*bool*) – If True, object is considered “fully” initialized. Default False.

class FirewallRule(*cb, parent, initial_data*)

Bases: [MutableBaseModel](#)

Represents a single firewall rule.

Parameters

- **action** – The action to take when rule is hit
- **application_path** – The application path to limit the rule
- **direction** – The direction the network request is being made from
- **enabled** – Whether the rule is enabled
- **protocol** – The type of network request
- **local_ip_address** – IPv4 address of the local side of the network connection (stored as dotted decimal)
- **local_port_ranges** – TCP or UDP port used by the local side of the network connection
- **remote_ip_address** – IPv4 address of the remote side of the network connection (stored as dotted decimal)
- **remote_port_ranges** – TCP or UDP port used by the remote side of the network connection
- **test_mode** – Enables host-based firewall hits without blocking network traffic or generating alerts

Initialize the FirewallRule object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the firewall rule.
- **parent** ([HostBasedFirewallRuleConfig](#)) – The parent rule configuration.

delete()

Delete this object.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

refresh()

Reload this object from the server.

remove()

Removes this rule from the rule group that contains it.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

MutableBaseModel

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(*fulltouch=False*)

Force this object to be considered as changed.

validate()

Validates this object.

Returns

True if the object is validated.

Return type

bool

Raises

InvalidObjectError – If the object has missing fields.

class FirewallRuleGroup(*cb, parent, initial_data*)

Bases: *MutableBaseModel*

Represents a group of related firewall rules.

Parameters

- **name** – Name of the rule group
- **description** – Description of the rule group
- **rules** – List of rules in the rule group

Initialize the FirewallRuleGroup object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the firewall rule group.
- **parent** ([HostBasedFirewallRuleConfig](#)) – The parent rule configuration.

append_rule(*name, action, direction, protocol, remote_ip, **kwargs*)

Creates a new FirewallRule object and appends it to this rule group.

Parameters

- **name** (*str*) – The name for the new rule.
- **action** (*str*) – The action to be taken by this rule. Valid values are “ALLOW,” “BLOCK,” and “BLOCK_ALERT.”
- **direction** (*str*) – The traffic direction this rule matches. Valid values are “IN,” “OUT,” and “BOTH.”
- **protocol** (*str*) – The network protocol this rule matches. Valid values are “TCP” and “UDP.”
- **remote_ip** (*str*) – The remote IP address this rule matches.
- **kwargs** (*dict*) – Additional parameters which may be added to the new rule.

Returns

The new rule object.

Return type

[FirewallRule](#)

delete()

Delete this object.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

refresh()

Reload this object from the server.

remove()

Removes this rule group from the rule configuration.

reset()

Undo any changes made to this object’s fields.

property rules_

Returns a list of the firewall rules within this rule group.

Returns

List of contained firewall rules.

Return type

list(*HostBasedFirewallRuleConfig.FirewallRule*)

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

MutableBaseModel

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(*fulltouch=False*)

Force this object to be considered as changed.

validate()

Validates this object.

Returns

True if the object is validated.

Return type

bool

Raises

InvalidObjectError – If the object has missing fields.

append_rule_group(*name, description*)

Creates a new FirewallRuleGroup object and appends it to the list of rule groups in the rule configuration.

Parameters

- **name** (*str*) – The name of the new rule group.
- **description** (*str*) – The description of the new rule group.

Returns

The newly added rule group.

Return type

FirewallRuleGroup

copy_rules(args*)**

Copies the parameters for host-based firewall rule configurations to another policy or policies.

Required Permissions:

org.firewall.rules(UPDATE)

Parameters

args (*list[Any]*) – References to policies to copy to. May be Policy objects, integers, or string representations of integers.

Returns

Result structure from copy operation.

Return type

dict

Raises*ApiError* – If the parameters could not be converted to policy IDs.**property default_action**

Returns the default action of this rule configuration.

Returns

The default action of this rule configuration, either “ALLOW” or “BLOCK.”

Return type

str

delete()

Delete this object.

property enabled

Returns whether or not the host-based firewall is enabled.

Returns

True if the host-based firewall is enabled, False if not.

Return type

bool

export_rules(format='json')

Exports the rules from this host-based firewall rule configuration.

Required Permissions:

org.firewall.rules(READ)

Parameters**format** (*str*) – The format to return the rule data in. Valid values are “csv” and “json” (the default).**Returns**

The exported rule configuration data.

Return type

str

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_parameter(name, default_value=None)

Returns a parameter value from the rule configuration.

Parameters

- **name** (*str*) – The parameter name.
- **default_value** (*Any*) – The default value to return if there’s no parameter by that name. Default is None.

Returns

The parameter value, or None if there is no value.

Return type

Any

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

property parameter_names

Returns a list of parameter names in this rule configuration.

Returns

A list of parameter names in this rule configuration.

Return type

list[str]

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object’s fields.

property rule_groups

Returns the list of rule groups in this rule configuration.

Returns

The list of rule groups.

Return type

list[*FirewallRuleGroup*]

save()

Save any changes made to this object’s fields.

Returns

This object.

Return type

MutableBaseModel

set_default_action(action)

Sets the default action of this rule configuration.

Parameters

action (*str*) – The new default action of this rule configuration. Valid values are “ALLOW” and “BLOCK.”

set_enabled(flag)

Sets whether or not the host-based firewall is enabled.

Parameters

flag (*bool*) – True if the host-based firewall should be enabled, False if not.

set_parameter(name, value)

Sets a parameter value into the rule configuration.

Parameters

- **name** (*str*) – The parameter name.
- **value** (*Any*) – The new value to be set.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(fulltouch=False)

Force this object to be considered as changed.

validate()

Validates this rule configuration against its constraints.

Raises

[*InvalidObjectError*](#) – If the rule object is not valid.

class PolicyRuleConfig(*cb, parent, model_unique_id=None, initial_data=None, force_init=False, full_doc=False*)

Bases: [*MutableBaseModel*](#)

Represents a rule configuration in the policy.

Create one of these objects, associating it with a Policy, and set its properties, then call its `save()` method to add the rule configuration to the policy. This requires the `org.policies(UPDATE)` permission.

To update a PolicyRuleConfig, change the values of its property fields, then call its `save()` method. This requires the `org.policies(UPDATE)` permission.

To delete an existing PolicyRuleConfig, call its `delete()` method. This requires the `org.policies(DELETE)` permission.

Parameters

- **id** – The ID of this rule config
- **name** – The name of this rule config
- **description** – The description of this rule config
- **inherited_from** – Indicates where the rule config was inherited from
- **category** – The category for this rule config
- **parameters** – The parameters associated with this rule config

Initialize the PolicyRuleConfig object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **parent** ([Policy](#)) – The “parent” policy of this rule configuration.
- **model_unique_id** (*str*) – ID of the rule configuration.
- **initial_data** (*dict*) – Initial data used to populate the rule configuration.
- **force_init** (*bool*) – If True, forces the object to be refreshed after constructing. Default False.
- **full_doc** (*bool*) – If True, object is considered “fully” initialized. Default False.

delete()

Delete this object.

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_parameter(name, default_value=None)

Returns a parameter value from the rule configuration.

Parameters

- **name** (*str*) – The parameter name.
- **default_value** (*Any*) – The default value to return if there’s no parameter by that name. Default is None.

Returns

The parameter value, or None if there is no value.

Return type

Any

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

property parameter_names

Returns a list of parameter names in this rule configuration.

Returns

A list of parameter names in this rule configuration.

Return type

list[str]

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type*MutableBaseModel***set_parameter(name, value)**

Sets a parameter value into the rule configuration.

Parameters

- **name** (str) – The parameter name.
- **value** (Any) – The new value to be set.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(fulltouch=False)

Force this object to be considered as changed.

validate()

Validates this rule configuration against its constraints.

Raises*InvalidObjectError* – If the rule object is not valid.

4.11.15 Previewer Module

This module contains the DevicePolicyChangePreview object.

When methods on Device, Policy, or AssetGroup are called to “preview” changes in device policy, a list of these objects is returned. Each object represents a change in “effective” policy on one or more devices.

class DevicePolicyChangePreview(cb, preview_data)

Bases: object

Contains data previewing a change in device policies.

Changes to policies may happen via asset group memberships, policy rank changes, device policy overrides, or other causes.

Each one of these objects shows, for a given group of assets, the current policy that is the “effective policy” for those assets, the new policy that will be the “effective policy” for those assets, the number of assets affected, and which assets they are.

Creates a new instance of `AssetGroupChangePreview`.

Parameters

- **cb** (`BaseAPI`) – Reference to API object used to communicate with the server.
- **preview_data** (`dict`) – Contains the preview data returned by the server API.

property asset_count

The number of assets to be affected by the change in their effective policy.

property asset_query

A Device query which looks up the assets that are to be affected by the change in their effective policy.

Once the query is created, it can be modified with additional criteria or options before it is executed.

property assets

The list of assets, i.e. Device objects, to be affected by the change in their effective policy.

Required Permissions:

device (READ)

property current_policy

The Policy object that is the current “effective” policy for a group of assets.

property current_policy_id

The ID of the policy that is the current “effective” policy for a group of assets.

property current_policy_position

The position, or rank, of the policy that is the current “effective” policy for a group of assets.

property new_policy

The Policy object that will become the new “effective” policy for a group of assets.

property new_policy_id

The ID of the policy that will become the new “effective” policy for a group of assets.

property new_policy_position

The position, or rank, of the policy that will become the new “effective” policy for a group of assets.

4.11.16 Processes Module

Model and query that allow location and manipulation of process data reported by an organization’s sensors.

This data can be used to identify applications that are acting abnormally and over time, cull the outliers from the total observed process activity, and retroactively identify the origination point for attacks that previously escaped notice. Use cases include:

- Finding the process that was identified in an alert with a process search.
- Finding processes that match targeted behavioral characteristics identified in Carbon Black or third-party threat intelligence reports.
- Finding additional details about processes that were potentially involved in malicious activity identified elsewhere.
- Using faceting to get filtering terms or prevalent values in a set of processes.

Locating processes generally requires the Endpoint Standard or Enterprise EDR products.

Typical usage example:

```
>>> query = api.select(Process).where("process_name:chrome.exe")
>>> for process in query:
...     print(f"Chrome PID = {process.process_guid}")
```

class AsyncProcessQuery(*doc_class, cb*)

Bases: [Query](#)

A query object used to search for Process objects asynchronously.

Create one of these objects by calling `select(Process)` on a `CBCloudAPI` object.

Initialize the `AsyncProcessQuery` object.

Parameters

- **doc_class** (*class*) – The class of the model this query returns.
- **cb** (`CBCloudAPI`) – A reference to the `CBCloudAPI` object.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

add_exclusions(*key, newlist*)

Add to the exclusions on this query with a custom exclusions key.

Will overwrite any existing exclusion for the specified key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).add_exclusions("type", ["WATCHLIST"])
>>> query = api.select(Alert).add_exclusions("type", "WATCHLIST")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None, **kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

batch_size(*new_batch_size*)

Set the batch size of the paginated query.

Parameters

new_batch_size (*int*) – The new batch size.

Returns

A new query with the updated batch size.

Return type

PaginatedQuery

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(*q=None, **kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

set_fields(*fields*)

Sets the fields to be returned with the response.

Parameters

fields (*str* or *list[str]*) – Field or list of fields to be returned.

set_rows(*rows*)

Sets the number of rows to request per batch.

This will not limit the total results to the specified number of rows; instead, the query will use this to determine how many rows to request at a time from the server.

Parameters

rows (*int*) – How many rows to request.

set_start(*start*)

Sets the ‘start’ query body parameter, determining where to begin retrieving results from.

Parameters

start (*int*) – Where to start results from.

set_time_range(*start=None, end=None, window=None*)

Sets the 'time_range' query body parameter, determining a time window based on 'device_timestamp'.

Parameters

- **start** (*str in ISO 8601 timestamp*) – When to start the result search.
- **end** (*str in ISO 8601 timestamp*) – When to end the result search.
- **window** (*str*) – Time window to execute the result search, ending on the current time. Should be in the form “-2w”, where y=year, w=week, d=day, h=hour, m=minute, s=second.

Note:

- *window* will take precedent over *start* and *end* if provided.
-

Examples

```
>>> query = api.select(Process).set_time_range(start="2020-10-20T20:34:07Z").
↳ where("query is required")
>>> second_query = api.select(Process).
...     set_time_range(start="2020-10-20T20:34:07Z", end="2020-10-30T20:34:07Z
↳ ").where("query is required")
>>> third_query = api.select(Process).set_time_range(window='-3d').where("query_
↳ is required")
```

sort_by(*key, direction='ASC'*)

Sets the sorting behavior on a query's results.

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

The query with sorting parameters.

Return type

Query

Example

```
>>> cb.select(Process).where(process_name="cmd.exe").sort_by("device_timestamp")
```

timeout(*msecs*)

Sets the timeout on a process query.

Parameters

msecs (*int*) – Timeout duration, in milliseconds. This can never be greater than the configured default timeout. If this is 0, the configured default timeout is used.

Returns

The modified query object.

Return type*AsyncProcessQuery***Example**

```
>>> cb.select(Process).where(process_name="foo.exe").timeout(5000)
```

update_criteria(key, newlist)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_↵value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

update_exclusions(key, newlist)

Update the exclusion on this query with a custom exclusion key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*list*) – List of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).update_exclusions("my.criteria.key", ["criteria_↵value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(q=None, **kwargs)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

class Process(*cb, model_unique_id=None, initial_data=None, force_init=False, full_doc=False*)

Bases: *UnrefreshableModel*

Information about a process running on one of the endpoints connected to the Carbon Black Cloud.

Objects of this type are retrieved through queries to the Carbon Black Cloud server, such as via `AsyncProcessQuery`.

Examples

```
>>> # use the Process GUID directly
>>> process = api.select(Process, "WNEXFKQ7-00050603-0000066c-00000000-
↳ 1d6c9acb43e29bb")
```

```
>>> # use the Process GUID in a where() clause
>>> process_query = api.select(Process).where(process_guid=
...     "WNEXFKQ7-00050603-0000066c-00000000-1d6c9acb43e29bb")
>>> process_query_results = list(process_query)
>>> process_2 = process_query_results[0]
```

Initialize the Process object.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the *CBCloudAPI* object.
- **model_unique_id** (*str*) – The unique ID (GUID) for this process.
- **initial_data** (*dict*) – The data to use when initializing the model object.
- **force_init** (*bool*) – True to force object initialization.
- **full_doc** (*bool*) – True to mark the object as fully initialized.

class Summary(*cb, model_unique_id=None, initial_data=None, force_init=False, full_doc=True*)

Bases: *UnrefreshableModel*

A summary of organization-specific information for a process.

The preferred interface for interacting with Summary models is `Process.summary`.

Example

```
>>> process = api.select(Process, "WNEXFKQ7-00050603-0000066c-00000000-
↳ 1d6c9acb43e29bb")
>>> summary = process.summary
```

Initialize the Summary object.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the *CBCloudAPI* object.

- **model_unique_id** (*str*) – The unique ID for this particular instance of the model object.
- **initial_data** (*dict*) – The data to use when initializing the model object.
- **force_init** (*bool*) – True to force object initialization.
- **full_doc** (*bool*) – True to mark the object as fully initialized.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class Tree(*cb*, *model_unique_id=None*, *initial_data=None*, *force_init=False*, *full_doc=True*)

Bases: [UnrefreshableModel](#)

Summary of organization-specific information for a process.

The preferred interface for interacting with Tree models is `Process.tree`.

Example

```
>>> process = api.select(Process, "WNEXFKQ7-00050603-0000066c-00000000-  
↪1d6c9acb43e29bb")  
>>> tree = process.tree
```

Initialize the Tree object.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **model_unique_id** (*str*) – The unique ID for this particular instance of the model object.
- **initial_data** (*dict*) – The data to use when initializing the model object.
- **force_init** (*bool*) – True to force object initialization.
- **full_doc** (*bool*) – True to mark the object as fully initialized.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

approve_process_sha256(*description*=")

Approves the application by adding the process_sha256 to the WHITE_LIST.

Parameters

description (*str*) – The justification for why the application was added to the WHITE_LIST.

Returns

ReputationOverride object created in the Carbon Black Cloud.

Return type

cbc_sdk.platform.ReputationOverride

ban_process_sha256(*description*=")

Bans the application by adding the process_sha256 to the BLACK_LIST.

Parameters

description (*str*) – The justification for why the application was added to the BLACK_LIST.

Returns

cbc_sdk.platform.ReputationOverride) ReputationOverride object created in the Carbon Black Cloud.

property children

Returns a list of child processes for this process.

deobfuscate_cmdline()

Deobfuscates the command line of the process and returns the deobfuscated result.

Required Permissions:

script.deobfuscation(EXECUTE)

Returns

A dict containing information about the obfuscated command line, including the deobfuscated result.

Return type

dict

events(*kwargs*)**

Returns a query for events associated with this process's process GUID.

Parameters

kwargs – Arguments to filter the event query with.

Example

```
>>> [print(event) for event in process.events()]
>>> [print(event) for event in process.events(event_type="modload")]
```

facets()

Returns a FacetQuery for a Process.

This represents the search for a summary of result groupings (facets). The returned AsyncFacetQuery object must have facet fields or ranges specified before it can be submitted, using the `add_facet_field()` or `add_range()` methods.

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_details(timeout=0, async_mode=False)

Requests detailed information about this process from the Carbon Black Cloud server.

Required Permissions:

org.search.events(CREATE, READ)

Parameters

- **timeout** (*int*) – Event details request timeout in milliseconds. This value can never be greater than the configured default timeout. If this value is 0, the configured default timeout is used.
- **async_mode** (*bool*) – True to request details in an asynchronous manner.

Returns

If **async_mode** is **True**. Call `result()` on this Future to wait for completion and retrieve the results.

dict: If **async_mode** is **False**.

Return type

Future

property parents

Returns the parent process associated with this process, or `None` if there is no recorded parent.

property process_md5

Returns a string representation of the MD5 hash for this process.

property process_pids

Returns a list of integer PIDs associated with this process, or `None` if there are none.

property process_sha256

Returns a string representation of the SHA256 hash for this process.

refresh()

Reload this object from the server.

property siblings

Returns a list of sibling processes for this process.

property summary

Returns organization-specific information about this process.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

property tree

Returns a process tree associated with this process.

Example

```
>>> tree = process.tree
```

class ProcessFacet(*cb, model_unique_id, initial_data*)

Bases: [UnrefreshableModel](#)

Represents the results of a process facet query.

ProcessFacet objects contain both Terms and Ranges. Each of those contain facet fields and values.

Access all of the Terms facet data with [ProcessFacet.Terms.facets\(\)](#) or see just the field names with [ProcessFacet.Terms.fields\(\)](#).

Access all of the Ranges facet data with [ProcessFacet.Ranges.facets\(\)](#) or see just the field names with [ProcessFacet.Ranges.fields\(\)](#).

Process facets can be queried for via `CBCloudAPI.select(ProcessFacet)`. Specify facet field(s) with `.add_facet_field("my_facet_field")`.

Optionally, you can limit the facet query to a single process with the following two options. Using the solrq builder, specify process GUID with `.where(process_guid="example_guid")` and modify the query with `.or_(parent_effective_reputation="KNOWN_MALWARE")` and `.and_(parent_effective_reputation="KNOWN_MALWARE")`.

If you want full control over the query string, specify the process GUID in the query string `.where("process_guid: example_guid OR parent_effective_reputation: KNOWN_MALWARE")`

Examples:

```
>>> process_facet_query = api.select(ProcessFacet).where(process_guid=
...     "WNEXFKQ7-00050603-0000066c-00000000-1d6c9acb43e29bb")
>>> process_facet_query.add_facet_field("device_name")
```

```
# retrieve results synchronously >>> facet = process_facet_query.results
# retrieve results asynchronously >>> future = process_facet_query.execute_async() >>> result
= future.result()
# result is a list with one item, so access the first item >>> facet = result[0]
```

Parameters

- **job_id** – The Job ID assigned to this query
- **terms** – Contains the Process Facet search results
- **ranges** – Groupings for search result properties that are ISO 8601 timestamps or numbers
- **contacted** – The number of searchers contacted for this query
- **completed** – The number of searchers that have reported their results

Initialize a ProcessFacet object with `initial_data`.

class `Ranges(cb, initial_data)`

Bases: `UnrefreshableModel`

The range (bucketed) facet fields and values associated with a process facet query.

Initialize a ProcessFacet.Ranges object with `initial_data`.

property facets

Returns the reified facets for this result.

property fields

Returns the ranges fields for this result.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class `Terms(cb, initial_data)`

Bases: `UnrefreshableModel`

The facet fields and values associated with a process facet query.

Initialize a ProcessFacet.Terms object with `initial_data`.

property facets

Returns the terms' facets for this result.

property fields

Returns the terms facets' fields for this result.

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

property ranges_

Returns the reified `ProcessFacet.Ranges` for this result.

refresh()

Reload this object from the server.

property terms_

Returns the reified `ProcessFacet.Terms` for this result.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class `SummaryQuery(doc_class, cb)`

Bases: [BaseQuery](#), [AsyncQueryMixin](#), [QueryBuilderSupportMixin](#)

A query used to search for `Process.Summary` or `Process.Tree` objects.

Create one of these queries with a `select()` on either `Process.Summary` or `Process.Tree`. These queries are also created by accessing the `summary` or `tree` properties on `Process`.

Initialize the `SummaryQuery` object.

Parameters

- **doc_class** (*class*) – The class of the model this query returns.
- **cb** ([CBCloudAPI](#)) – A reference to the `CBCloudAPI` object.

and_(*q=None, **kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or `solrq.Q` object
- ****kwargs** (*dict*) – Arguments to construct a `solrq.Q` with

Returns

This Query object.

Return type

[Query](#)

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

not_(*q=None, **kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (`solrq.Q`) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a `solrq.Q` with.

Returns

This Query object.

Return type

[Query](#)

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (`solrq.Q`) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a `solrq.Q` with.

Returns

This Query object.

Return type*Query***property results**

Return the results of this query. If the query has not yet been run, it is run to determine the results.

Required Permissions:

org.search.events(CREATE, READ)

set_time_range(*start=None, end=None, window=None*)

Sets the time_range query body parameter, determining a time window based on device_timestamp.

Parameters

- **start** (*str in ISO 8601 timestamp*) – When to start the result search.
- **end** (*str in ISO 8601 timestamp*) – When to end the result search.
- **window** (*str*) – Time window to execute the result search, ending on the current time. Should be in the form “-nx”, where n is an integer and x is y=year, w=week, d=day, h=hour, m=minute, s=second.

Note: window will take precedent over start and end if provided.

Example

```
>>> query = api.select(Event).set_time_range(start="2020-10-20T20:34:07Z")
>>> second_query = api.select(Event).set_time_range
...     (start="2020-10-20T20:34:07Z", end="2020-10-30T20:34:07Z")
>>> third_query = api.select(Event).set_time_range(window='-3d')
```

timeout(*msecs*)

Sets the timeout on a process query.

Parameters

msecs (*int*) – Timeout duration, in milliseconds. This can never be greater than the configured default timeout. If this value is 0, the configured default timeout is used.

Returns

The modified query object.

Return type*SummaryQuery***Example**

```
>>> cb.select(Process).where(process_name="foo.exe").timeout(5000)
```

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

4.11.17 Reputation Module

Model and Query Classes for Reputation

class ReputationOverride(*cb, model_unique_id, initial_data=None*)

Bases: *PlatformModel*

Represents a reputation override.

Parameters

- **id** – An identifier for a reputation override
- **created_by** – Creator of the override
- **create_time** – Time the override was created
- **description** – Justification for override
- **override_list** – The override list to add a new reputation (BLACK_LIST only valid for SHA256)
- **override_type** – Process property match when applying override
- **sha256_hash** – A hexadecimal string of length 64 characters representing the SHA-256 hash of the application
- **filename** – An application name for the hash
- **signed_by** – Name of the signer for the application
- **certificate_authority** – Certificate authority that authorizes the validity of the certificate
- **path** – The absolute path to file or directory where tool exists on disk
- **include_child_processes** – Include tool's child processes on approved list

Initialize the ReputationOverride object.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

classmethod bulk_delete(*cb, overrides*)

Deletes reputation overrides in bulk by id.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **overrides** (*List*) – List of reputation override ids

Example

```
>>>
[
    "e9410b754ea011ebbfd0db2585a41b07"
]
```

classmethod `create(cb, initial_data)`

Returns all vendors and products that have been seen for the organization.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*Object*) – The initial data for a ReputationOverride

Example

```
>>>
{
    "description": "Banned as known malware",
    "override_list": "BLACK_LIST",
    "override_type": "SHA256",
    "sha256_hash":
    → "dd191a5b23df92e13a8852291f9fb5ed594b76a28a5a464418442584afd1e048",
    "filename": "foo.exe"
}
```

Returns

The created ReputationOverride object based on the specified properties

Return type

ReputationOverride

delete()

Delete this object.

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class ReputationOverrideQuery(*doc_class, cb*)

Bases: [BaseQuery](#), [QueryBuilderSupportMixin](#), [IterableQueryMixin](#), [AsyncQueryMixin](#)

Represents a query that is used to locate ReputationOverride objects.

Initialize the ReputationOverrideQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None, **kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

[Query](#)

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(*q=None, **kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- ***MoreThanOneResultError*** – If the query returns more than one item
- ***ObjectNotFoundError*** – If the query returns zero items

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

set_override_list(*override_list*)

Sets the *override_list* criteria filter.

Parameters

override_list (*str*) – Override List to filter on.

Returns

The ReputationOverrideQuery with specified *override_list*.

set_override_type(*override_type*)

Sets the *override_type* criteria filter.

Parameters

override_type (*str*) – Override List to filter on.

Returns

The ReputationOverrideQuery with specified *override_type*.

sort_by(*key, direction='ASC'*)

Sets the sorting behavior on a query's results.

Example

```
>>> cb.select(ReputationOverride).sort_by("create_time")
```

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

This instance.

Return type

ReputationOverrideQuery

Raises

ApiError – If an invalid direction value is passed.

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, *QueryBuilder*, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

4.11.18 Users Module

Model and Query Classes for Users

class User(*cb, model_unique_id, initial_data=None*)

Bases: *MutableBaseModel*

Represents a user in the Carbon Black Cloud.

Parameters

- **org_key** – Organization key for this user
- **auth_method** – Method to be used for the user to authenticate
- **admin_login_version** – Version number of the user information
- **email** – User’s E-mail address
- **login_name** – Login name for the user
- **login_id** – Login ID (user ID) for this user
- **phone** – User’s phone number
- **first_name** – User’s first name
- **last_name** – User’s last name

- **org_id** – ID of the organization the user is in
- **org_admin_version** – TBD
- **role** – Not used, always “DEPRECATED”
- **contact_id** – ID of the user’s contact information
- **contact_version** – Version of the user’s contact information

Initialize the User object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*int*) – Login ID of this user.
- **initial_data** (*dict*) – Initial data used to populate the user.

class UserBuilder(*cb*)

Bases: `object`

Auxiliary object used to construct a new User.

Create the empty UserBuilder object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.

add_grant_profile(*orgs, roles*)

Adds a grant profile for the new user.

Parameters

- **orgs** (*list[str]*) – List of organizations to be allowed, specified as keys or URNs.
- **roles** (*list[str]*) – List of roles to be granted, specified as URNs.

Returns

This object.

Return type

[UserBuilder](#)

build()

Builds the new user.

Notes

The new user will not be “findable” by other API functions until it has been activated and its initial password has been set.

set_auth_method(*method*)

Sets the authentication method for the new user. The default is ‘PASSWORD’.

Parameters

- **method** (*str*) – The authentication method for the new user.

Returns

This object.

Return type

[UserBuilder](#)

set_email(*email*)

Sets the E-mail address for the new user.

Parameters

- **email** (*str*) – The E-mail address for the new user.

Returns

This object.

Return type

UserBuilder

set_first_name(*first_name*)

Sets the first name for the new user.

Parameters

first_name (*str*) – The first name for the new user.

Returns

This object.

Return type

UserBuilder

set_last_name(*last_name*)

Sets the last name for the new user.

Parameters

last_name (*str*) – The last name for the new user.

Returns

This object.

Return type

UserBuilder

set_phone(*phone*)

Sets the phone number for the new user.

Parameters

phone (*str*) – The phone number for the new user.

Returns

This object.

Return type

UserBuilder

set_role(*role*)

Sets the role URN for the new user.

Parameters

role (*str*) – The URN of the role to set for the user.

Returns

This object.

Return type

UserBuilder

add_profiles(*profile_templates*)

Add the specified profiles to the user's grant.

Parameters

profile_templates (*list[dict]*) – List of profile templates to be added to the user.

classmethod bulk_add_profiles(*users, profile_templates*)

Add the specified profiles to the specified users' grants.

Parameters

- **users** (*list[User]*) – List of User objects specifying users to be modified.
- **profile_templates** (*list[dict]*) – List of profile templates to be added to the users.

classmethod bulk_create(*cb, user_templates, profile_templates*)

Creates a series of new users.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **user_templates** (*list[dict]*) – List of templates for users to be created.
- **profile_templates** (*list[dict]*) – List of profile templates to be applied to each user.

classmethod **bulk_delete**(*users*)

Deletes all the listed users.

Parameters

users (*list[User]*) – List of User objects specifying users to be deleted.

classmethod **bulk_disable_all_access**(*users*)

Disables all access profiles held by the listed users.

Parameters

users (*list[User]*) – List of User objects specifying users to be disabled.

classmethod **bulk_disable_profiles**(*users, profile_templates*)

Disable the specified profiles in the specified users' grants.

Parameters

- **users** (*list[User]*) – List of User objects specifying users to be modified.
- **profile_templates** (*list[dict]*) – List of profile templates to be disabled.

change_role(*role_urn, org=None*)

Add the specified role to the user (either to the grant or the profiles).

Parameters

- **role_urn** (*str*) – URN of the role to be added.
- **org** (*str*) – If specified, only profiles that match this organization will have the role added. Organization may be specified as either an org key or a URN.

Raises

[ApiError](#) – If the user is a “legacy” user that has no grant.

classmethod **create**(*cb, template=None*)

Creates a new user.

Parameters

- **cb** ([CBCloudAPI](#)) – A reference to the CBCloudAPI object.
- **template** (*dict*) – Optional template data for creating the new user.

Returns

If **template** is **None**, returns an instance of this object. Call methods on the object to set the values associated with the new user, and then call **build()** to create it.

Return type

[UserBuilder](#)

delete()

Delete this object.

disable_all_access()

Disables all access profiles held by this user.

Raises

ApiError – If the user is a “legacy” user that has no grant.

disable_profiles(profile_templates)

Disable the specified profiles in the user’s grant.

Parameters

profile_templates (*list[dict]*) – List of profile templates to be disabled.

Raises

ApiError – If the user is a “legacy” user that has no grant.

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

grant()

Locates the access grant for this user.

Returns

Access grant for this user, or None if the user has none.

Return type

Grant

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

property org_urn

Returns the URN for this user’s organization (used in accessing Grants).

Returns

URN for this user’s organization.

Return type

str

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object’s fields.

reset_google_authenticator_registration()

Forces Google Authenticator registration to be reset for this user.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

MutableBaseModel

set_profile_expiration(profile_templates, expiration_date)

Set the expiration time for the specified profiles in the user's grant.

Parameters

- **profile_templates** (*list[dict]*) – List of profile templates to be reset.
- **expiration_date** (*str*) – New expiration date, in ISO 8601 format.

Raises

ApiError – If the user is a “legacy” user that has no grant.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(fulltouch=False)

Force this object to be considered as changed.

property urn

Returns the URN for this user (used in accessing Grants).

Returns

URN for this user.

Return type

str

validate()

Validates this object.

Returns

True if the object is validated.

Return type

bool

Raises

InvalidObjectError – If the object has missing fields.

class UserQuery(doc_class, cb)

Bases: *BaseQuery*, *IterableQueryMixin*, *AsyncQueryMixin*

Query for retrieving users in bulk.

Initialize the Query object.

Parameters

- **doc_class** (*class*) – The class of the model this query returns.
- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

email_addresses(addr)

Limit the query to users with the specified E-mail addresses. Call multiple times to add multiple addresses.

Parameters

addr (*list[str]*) – List of addresses to be added to the query.

Returns

This object.

Return type

UserQuery

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

user_ids(userids)

Limit the query to users with the specified user IDs. Call multiple times to add multiple user IDs.

Parameters

userids (*list[str]*) – List of user IDs to be added to the query.

Returns

This object.

Return type

UserQuery

log = <Logger cbc_sdk.platform.users (WARNING)>

User Models

normalize_profile_list(*profile_templates*)

Internal function to normalize a list of profile templates.

4.11.19 Vulnerability Assessment Module

Model and Query Classes for Vulnerability Assessment API

class AffectedAssetQuery(*vulnerability, cb*)

Bases: *VulnerabilityQuery*

Query Class for the Vulnerability

Initialize the AffectedAssetQuery.

Parameters

- **vulnerability** (*class*) – The vulnerability that will be returned by this query.
- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.

add_criteria(*key, value, operator='EQUALS'*)

Restricts the vulnerabilities that this query is performed on to the specified key value pair.

Parameters

- **key** (*str*) – Property from the vulnerability object
- **value** (*str*) – Value of the property to filter by
- **operator** (*str*) – (optional) logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None, **kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

export()

Performs the query and export the results in the form of a Job.

Example

```
>>> # Create the Vulnerability query
>>> query = cb.select(Vulnerability).set_severity('CRITICAL')
>>> # Export the results
>>> job = query.export()
>>> # wait for the export to finish
>>> job.await_completion()
>>> # write the results to a file
>>> job.get_output_as_file("vulnerabilities.csv")
```

Returns

The export job.

Return type

Job

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(q=None, **kwargs)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type*Query***one()**

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(*q=None*, *kwargs*)**

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type*Query***set_deployment_type(*deployment_type*, *operator*)**

Restricts the vulnerabilities that this query is performed on to the specified deployment type.

Parameters

- **deployment_type** (*str*) – deployment type (“ENDPOINT”, “AWS”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery***set_device_type(*device_type*, *operator*)**

Restricts the vulnerabilities that this query is performed on to the specified device type.

Parameters

- **device_type** (*str*) – device type (“WORKLOAD”, “ENDPOINT”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery*

set_highest_risk_score(*highest_risk_score, operator*)

Restricts the vulnerabilities that this query is performed on to the specified `highest_risk_score`.

Parameters

- **highest_risk_score** (*double*) – `highest_risk_score`.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_last_sync_ts(*last_sync_ts, operator*)

Restricts the vulnerabilities that this query is performed on to the specified `last_sync_ts`.

Parameters

- **last_sync_ts** (*str*) – `last_sync_ts`.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_name(*name, operator*)

Restricts the vulnerabilities that this query is performed on to the specified `name`.

Parameters

- **name** (*str*) – `name`.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_os_arch(*os_arch, operator*)

Restricts the vulnerabilities that this query is performed on to the specified `os_arch`.

Parameters

- **os_arch** (*str*) – `os_arch`.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_os_name(*os_name, operator*)

Restricts the vulnerabilities that this query is performed on to the specified `os_name`.

Parameters

- **os_name** (*str*) – `os_name`.

- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_os_product_id(*os_product_id*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *os_product_id*.

Parameters

- **os_product_id** (*str*) – *os_product_id*.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

AffectedAssetQuery

set_os_type(*os_type*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *os_type*.

Parameters

- **os_type** (*str*) – *os_type* (“CENTOS”, “RHEL”, “SLES”, “UBUNTU”, “WINDOWS”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_os_version(*os_version*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *os_version*.

Parameters

- **os_version** (*str*) – *os_version*.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_severity(*severity*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *severity*.

Parameters

- **severity** (*str*) – *severity* (“CRITICAL”, “IMPORTANT”, “MODERATE”, “LOW”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery***set_sync_status**(*sync_status*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *sync_status*.

Parameters

- **sync_status** (*str*) – *sync_status* (“NOT_STARTED”, “MATCHED”, “ERROR”, “NOT_MATCHED”, “NOT_SUPPORTED”, “CANCELLED”, “IN_PROGRESS”, “ACTIVE”, “COMPLETED”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery***set_sync_type**(*sync_type*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *sync_type*.

Parameters

- **sync_type** (*str*) – *sync_type* (“MANUAL”, “SCHEDULED”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery***set_vcenter**(*vcenter_uuid*)

Restricts the vulnerabilities that this query is performed on to the specified *vcenter* id.

Parameters

vcenter_uuid (*str*) – *vcenter* uuid.

Returns

This instance.

Return type*VulnerabilityQuery***set_visibility**(*visibility*)

Restricts the vulnerabilities that this query is performed on to the specified *visibility*

Parameters

visibility (*str*) – The visibility state of the vulnerability. (supports ACTIVE, DIS-MISSED)

Returns

This instance.

Return type*VulnerabilityQuery***set_vm_id**(*vm_id*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *vm_id*.

Parameters

- **vm_id** (*str*) – vm_id.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_vuln_count(*vuln_count*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *vuln_count*.

Parameters

- **vuln_count** (*str*) – vuln_count.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

sort_by(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Example

```
>>> cb.select(Vulnerability).sort_by("status")
```

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

This instance.

Return type

VulnerabilityQuery

Raises

ApiError – If an invalid direction value is passed.

where(*q*=None, ***kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

class Vulnerability(*cb, model_unique_id, os_product_id=None, initial_data=None*)

Bases: [NewBaseModel](#)

Represents a vulnerability

Parameters

- **affected_assets** – List of affected assets
- **category** – Vulnerability category
- **device_count** – Number of affected devices
- **os_info** – Information about the operating system associated with the vulnerability
- **os_product_id** – Operating system product ID
- **product_info** – Information about the vulnerable product
- **vuln_info** – Information about the vulnerability

Initialize the Vulnerability object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the vulnerability represented.
- **os_product_id** (*str*) – *os_product_id* of the vulnerability used to uniquely identify a CVE with multiple OS/Product instances
- **initial_data** (*dict*) – Initial data used to populate the alert.

class AssetView(*cb, initial_data=None*)

Bases: `list`

Represents a list of Vulnerability for an organization.

Initialize Vulnerability.AssetView object

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*list[dict]*) – list of assets and their vulnerability view

append(*object, /*)

Append object to the end of the list.

clear()

Remove all items from list.

copy()

Return a shallow copy of the list.

count(*value, /*)

Return number of occurrences of value.

extend(*iterable, /*)

Extend list by appending elements from the iterable.

index(*value, start=0, stop=9223372036854775807, /*)

Return first index of value.

Raises `ValueError` if the value is not present.

insert(*index*, *object*, /)

Insert object before index.

pop(*index=-1*, /)

Remove and return item at index (default last).

Raises IndexError if list is empty or index is out of range.

remove(*value*, /)

Remove first occurrence of value.

Raises ValueError if the value is not present.

reverse()

Reverse *IN PLACE*.

sort(**, key=None, reverse=False*)

Sort the list in ascending order and return None.

The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).

If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.

The reverse flag can be set to sort in descending order.

class OrgSummary(*cb*, *initial_data=None*)

Bases: [UnrefreshableModel](#)

Represents a vulnerability summary for an organization.

Parameters

- **monitored_assets** – Number of assets being monitored
- **severity_summary** – Information about vulnerabilities at each severity level

Initialize Vulnerability.OrgSummary object

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – dictionary of the data

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

severity_levels()

Returns the severity levels

Returns

List of severities

Return type

Severities (list[str])

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

get(attrname, default_val=None)

Return an attribute of this object.

Parameters

- **attrname** (str) – Name of the attribute to be returned.
- **default_val** (Any) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

get_affected_assets()

Returns an AffectedAssetQuery to fetch the list of devices affected by the Vulnerability.

Args;

os_product_id (str) operating system product ID

Returns

AffectedAssetQuery

perform_action(type, reason=None, notes=None)

Take an action to manage the Vulnerability.

Parameters

- **type** (str) – The type of action. (supports DISMISS, DISMISS_EDIT, or UNDISMISS)
- **reason** (str) – The reason the vulnerability is dismissed. Required when type is DISMISS or DISMISS_EDIT. (supports FALSE_POSITIVE, RESOLUTION_DEFERRED, NON_ISSUE, NON_CRITICAL_ASSET, UNDER_RESOLUTION, OTHER)
- **notes** (str) – Notes to be associated with the dismissal. Required when reason is OTHER.

Returns

The action response

Return type

obj

Raises

[*ApiError*](#) – If the request is invalid or missing required properties

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class VulnerabilityAssetViewQuery(*doc_class, cb*)

Bases: [*VulnerabilityQuery*](#)

Represents a query that is used fetch the Vulnerability Asset View

Initialize the VulnerabilityAssetViewQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** ([*BaseAPI*](#)) – Reference to API object used to communicate with the server.

add_criteria(*key, value, operator='EQUALS'*)

Restricts the vulnerabilities that this query is performed on to the specified key value pair.

Parameters

- **key** (*str*) – Property from the vulnerability object
- **value** (*str*) – Value of the property to filter by
- **operator** (*str*) – (optional) logic operator to apply to property value.

Returns

This instance.

Return type

[*VulnerabilityQuery*](#)

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None, **kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

[*Query*](#)

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

export()

Performs the query and export the results in the form of a Job.

Returns

The export job.

Return type

Job

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(q=None, **kwargs)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- ***MoreThanOneResultError*** – If the query returns more than one item
- ***ObjectNotFoundError*** – If the query returns zero items

or_(q=None, **kwargs)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

set_deployment_type(*deployment_type*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified deployment type.

Parameters

- **deployment_type** (*str*) – deployment type (“ENDPOINT”, “AWS”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_device_type(*device_type*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified device type.

Parameters

- **device_type** (*str*) – device type (“WORKLOAD”, “ENDPOINT”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_highest_risk_score(*highest_risk_score*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified highest_risk_score.

Parameters

- **highest_risk_score** (*double*) – highest_risk_score.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_last_sync_ts(*last_sync_ts*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified last_sync_ts.

Parameters

- **last_sync_ts** (*str*) – last_sync_ts.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_name(*name*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified name.

Parameters

- **name** (*str*) – name.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_os_arch(*os_arch*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified os_arch.

Parameters

- **os_arch** (*str*) – os_arch.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_os_name(*os_name*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified os_name.

Parameters

- **os_name** (*str*) – os_name.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_os_type(*os_type*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified os type.

Parameters

- **os_type** (*str*) – os type (“CENTOS”, “RHEL”, “SLES”, “UBUNTU”, “WINDOWS”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_os_version(*os_version*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified os_version.

Parameters

- **os_version** (*str*) – os_version.

- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_severity(*severity*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified severity.

Parameters

- **severity** (*str*) – severity (“CRITICAL”, “IMPORTANT”, “MODERATE”, “LOW”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_sync_status(*sync_status*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified sync_status.

Parameters

- **sync_status** (*str*) – sync_status (“NOT_STARTED”, “MATCHED”, “ERROR”, “NOT_MATCHED”, “NOT_SUPPORTED”, “CANCELLED”, “IN_PROGRESS”, “ACTIVE”, “COMPLETED”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_sync_type(*sync_type*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified sync_type.

Parameters

- **sync_type** (*str*) – sync_type (“MANUAL”, “SCHEDULED”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_vcenter(*vcenter_uuid*)

Restricts the vulnerabilities that this query is performed on to the specified vcenter id.

Parameters

vcenter_uuid (*str*) – vcenter uuid.

Returns

This instance.

Return type*VulnerabilityQuery***set_visibility**(*visibility*)

Restricts the vulnerabilities that this query is performed on to the specified visibility

Parameters

visibility (*str*) – The visibility state of the vulnerability. (supports ACTIVE, DIS-MISSED)

Returns

This instance.

Return type*VulnerabilityQuery***set_vm_id**(*vm_id*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified vm_id.

Parameters

- **vm_id** (*str*) – vm_id.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery***set_vuln_count**(*vuln_count*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified vuln_count.

Parameters

- **vuln_count** (*str*) – vuln_count.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery***sort_by**(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Example

```
>>> cb.select(Vulnerability).sort_by("status")
```

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

This instance.

Return type*VulnerabilityQuery***Raises***ApiError* – If an invalid direction value is passed.**where**(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type*Query***class VulnerabilityOrgSummaryQuery**(*doc_class, cb, device=None*)Bases: *BaseQuery*

Represents a query that is used fetch the VulnerabilitySummary

Initialize the VulnerabilityQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **device** (*cbc_sdk.platform.devices.Device*) – Optional Device object to indicate VulnerabilityQuery is for a specific device

set_severity(*severity*)

Restricts the vulnerability summary to a severity level

Parameters**severity** (*str*) – filters the vulnerability summary per severity (CRITICAL, IMPORTANT, MODERATE, LOW)**Returns**

This instance.

Return type*VulnerabilityOrgSummaryQuery***set_vcenter**(*vcenter_uuid*)

Restricts the vulnerability summary to a specific vcenter

Parameters**vcenter_uuid** (*str*) – vcenter uuid.**Returns**

This instance.

Return type*VulnerabilityOrgSummaryQuery*

set_visibility(*visibility*)

Restricts the vulnerabilities that this query is performed on to the specified visibility

Parameters

visibility (*str*) – The visibility state of the vulnerability. (supports ACTIVE, DIS-MISSED)

Returns

This instance.

Return type

VulnerabilityOrgSummaryQuery

submit()

Performs the query and returns the Vulnerability.OrgSummary

Returns

The vulnerability summary for the organization

Return type

Vulnerability.OrgSummary

class VulnerabilityQuery(*doc_class, cb, device=None*)

Bases: *BaseQuery, QueryBuilderSupportMixin, IterableQueryMixin, AsyncQueryMixin*

Represents a query that is used to locate Vulnerability objects.

Initialize the VulnerabilityQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **device** (*cbc_sdk.platform.devices.Device*) – Optional Device object to indicate VulnerabilityQuery is for a specific device

add_criteria(*key, value, operator='EQUALS'*)

Restricts the vulnerabilities that this query is performed on to the specified key value pair.

Parameters

- **key** (*str*) – Property from the vulnerability object
- **value** (*str*) – Value of the property to filter by
- **operator** (*str*) – (optional) logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None*, ***kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

export()

Performs the query and export the results in the form of a Job.

Example

```
>>> # Create the Vulnerability query
>>> query = cb.select(Vulnerability).set_severity('CRITICAL')
>>> # Export the results
>>> job = query.export()
>>> # wait for the export to finish
>>> job.await_completion()
>>> # write the results to a file
>>> job.get_output_as_file("vulnerabilities.csv")
```

Returns

The export job.

Return type

Job

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(*q=None*, ***kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.

- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- ***MoreThanOneResultError*** – If the query returns more than one item
- ***ObjectNotFoundError*** – If the query returns zero items

or_(q=None, **kwargs)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

set_deployment_type(deployment_type, operator)

Restricts the vulnerabilities that this query is performed on to the specified deployment type.

Parameters

- **deployment_type** (*str*) – deployment type (“ENDPOINT”, “AWS”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_device_type(device_type, operator)

Restricts the vulnerabilities that this query is performed on to the specified device type.

Parameters

- **device_type** (*str*) – device type (“WORKLOAD”, “ENDPOINT”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery***set_highest_risk_score**(*highest_risk_score*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *highest_risk_score*.

Parameters

- **highest_risk_score** (*double*) – *highest_risk_score*.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery***set_last_sync_ts**(*last_sync_ts*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *last_sync_ts*.

Parameters

- **last_sync_ts** (*str*) – *last_sync_ts*.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery***set_name**(*name*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *name*.

Parameters

- **name** (*str*) – *name*.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery***set_os_arch**(*os_arch*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *os_arch*.

Parameters

- **os_arch** (*str*) – *os_arch*.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery*

set_os_name(*os_name*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *os_name*.

Parameters

- **os_name** (*str*) – *os_name*.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_os_type(*os_type*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *os_type*.

Parameters

- **os_type** (*str*) – *os_type* (“CENTOS”, “RHEL”, “SLES”, “UBUNTU”, “WINDOWS”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_os_version(*os_version*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *os_version*.

Parameters

- **os_version** (*str*) – *os_version*.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_severity(*severity*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *severity*.

Parameters

- **severity** (*str*) – *severity* (“CRITICAL”, “IMPORTANT”, “MODERATE”, “LOW”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_sync_status(*sync_status*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *sync_status*.

Parameters

- **sync_status** (*str*) – sync_status (“NOT_STARTED”, “MATCHED”, “ERROR”, “NOT_MATCHED”, “NOT_SUPPORTED”, “CANCELLED”, “IN_PROGRESS”, “ACTIVE”, “COMPLETED”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_sync_type(*sync_type, operator*)

Restricts the vulnerabilities that this query is performed on to the specified sync_type.

Parameters

- **sync_type** (*str*) – sync_type (“MANUAL”, “SCHEDULED”)
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type

VulnerabilityQuery

set_vcenter(*vcenter_uuid*)

Restricts the vulnerabilities that this query is performed on to the specified vcenter id.

Parameters

vcenter_uuid (*str*) – vcenter uuid.

Returns

This instance.

Return type

VulnerabilityQuery

set_visibility(*visibility*)

Restricts the vulnerabilities that this query is performed on to the specified visibility

Parameters

visibility (*str*) – The visibility state of the vulnerability. (supports ACTIVE, DIS-MISSED)

Returns

This instance.

Return type

VulnerabilityQuery

set_vm_id(*vm_id, operator*)

Restricts the vulnerabilities that this query is performed on to the specified vm_id.

Parameters

- **vm_id** (*str*) – vm_id.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery***set_vuln_count**(*vuln_count*, *operator*)

Restricts the vulnerabilities that this query is performed on to the specified *vuln_count*.

Parameters

- **vuln_count** (*str*) – *vuln_count*.
- **operator** (*str*) – logic operator to apply to property value.

Returns

This instance.

Return type*VulnerabilityQuery***sort_by**(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Example

```
>>> cb.select(Vulnerability).sort_by("status")
```

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

This instance.

Return type*VulnerabilityQuery***Raises**

ApiError – If an invalid direction value is passed.

where(*q*=None, ***kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type*Query*

```
log = <Logger cbc_sdk.platform.vulnerability_assessment (WARNING)>
```

Vulnerability models

4.12 Workload Package

4.12.1 NSX Remediation Module

NSX Remediation for Workloads

class NSXRemediationJob(*cb, running_job_ids*)

Bases: object

An object that runs and monitors an NSX Remediation operation.

Creates a new NSXRemediationJob object.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **running_job_ids** (*list[str]*) – The list of running job IDs.

async_wait_result()

Sets up a Future which can be used to wait asynchronously for all running jobs to be completed.

Required Permissions:

appliances.registration(READ)

Returns

A future representing the job and its results.

Return type

Future

await_result()

Waits for all running jobs to be completed and returns the final status.

Required Permissions:

appliances.registration(READ)

Returns

The final status, mapping individual job IDs to status value dicts.

Return type

dict

classmethod start_request(*cb, device_ids, tag, set_tag=True*)

Starts an NSX Remediation request and returns the job object.

Required Permissions:

appliances.nsx.remediation(EXECUTE)

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **device_ids** (*int / list*) – The device ID(s) to run the remediation request on.
- **tag** (*str*) – The NSX tag to apply to specified devices. Valid values are “CB-NSX-Quarantine”, “CB-NSX-Isolate”, and “CB-NSX-Custom”.
- **set_tag** (*bool*) – True to toggle the specified tag on, False to toggle it off. Default True.

Returns

The object representing all running jobs.

Return type

NSXRemediationJob

Raises

- **ApiError** – If the parameters to start the request are incorrect.
- **ServerError** – If the request could not be successfully started.

property status

Returns the current status.

Returns

The current status, mapping individual job IDs to status value dicts.

Return type

dict

4.12.2 Sensor Lifecycle Module

Sensor Lifecycle Management for Workloads

class SensorKit(*cb, initial_data=None*)

Bases: *UnrefreshableModel*

Represents the information about a sensor, including installation file URLs.

Parameters

- **sensor_type** – The type of information this sensor is for.
- **sensor_url** – The URL for downloading the sensor installation package.
- **sensor_config_url** – The URL for downloading the sensor configuration information.
- **error_code** – Code for any error that occurred while getting the sensor information.
- **message** – Message for any error that occurred while getting the sensor information.

Initialize the SensorKit object.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **initial_data** (*dict*) – Initial data used to populate the sensor kit data.

classmethod from_type(*cb, device_type, architecture, sensor_type, version*)

Helper method used to create a temporary SensorKit object from its four components.

This method CANNOT be used to create an object that will be persisted to the server.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **device_type** (*str*) – Device type to be used. Valid values are “WINDOWS”, “LINUX”, and “MAC”.
- **architecture** (*str*) – Architecture to be used. Valid values are “32”, “64”, and “OTHER”.

- **sensor_type** (*str*) – Sensor type to be used. Valid values are “WINDOWS”, “MAC”, “RHEL”, “UBUNTU”, “SUSE”, and “AMAZON_LINUX”.
- **version** (*str*) – Sensor version number to be used.

Returns

A `SensorType` object with those specified values.

Return type

`SensorType`

Raises

[`ApiError`](#) – If an invalid value was used for one of the three limited values.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

classmethod **get_config_template**(*cb*)

Retrieve the sample config.ini file with the properties populated from the server.

Parameters

cb ([`BaseAPI`](#)) – Reference to API object used to communicate with the server.

Returns

Text of the sample configuration file.

Return type

`str`

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class **SensorKitQuery**(*doc_class*, *cb*)

Bases: [`BaseQuery`](#), [`CriteriaBuilderSupportMixin`](#), [`IterableQueryMixin`](#), [`AsyncQueryMixin`](#)

Query class used to read in SensorKit objects.

Initialize the `SensorKitQuery`.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** ([`BaseAPI`](#)) – Reference to API object used to communicate with the server.

add_criteria(*key*, *newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

add_sensor_kit_type(*skit=None*, ***kwargs*)

Add a sensor kit type to the request.

Parameters

- **skit** (*SensorKit*) – The sensor kit type to be added to the request.
- ****kwargs** (*dict*) – If skit is None, the keyword arguments ‘device_type’, ‘architecture’, ‘sensor_type’, and ‘version’ are used to create the sensor kit type to be added.

Returns

Reference to this object.

Return type

SensorKitQuery

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

config_params(*params*)

Sets the configuration parameters for the sensor kit query request.

Parameters

params (*str*) – The text of a config.ini file with a list of sensor properties to configure on installation.

Returns

Reference to this object.

Return type

SensorKitQuery

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

expires(*expiration_date_time*)

Sets the expiration date and time for the sensor kit query request.

Parameters

expiration_date_time (*str*) – The time at which the sensor download link will expire, expressed as ISO 8601 UTC.

Returns

Reference to this object.

Return type

SensorKitQuery

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

update_criteria(*key, newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

4.12.3 VM Workloads Search Module

Model and Query Classes for VM Workloads Search API

class `AWSComputeResource`(*cb, model_unique_id, initial_data=None*)

Bases: `BaseComputeResource`

Models an AWS compute resource.

Initialize the `AWSComputeResource` object.

Parameters

- **cb** (`BaseAPI`) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

classmethod `bulk_install`(*cb, compute_resources, sensor_kit_types, config_file=None*)

Install a sensor on a list of compute resources.

Parameters

- **cb** (`BaseAPI`) – Reference to API object used to communicate with the server.
- **compute_resources** (*list*) – A list of `ComputeResource` objects used to specify compute resources to install sensors on.
- **sensor_kit_types** (*list*) – A list of `SensorKit` objects used to specify sensor types to choose from in installation.
- **config_file** (*str*) – The text of a config.ini file with a list of sensor properties to configure on installation.

Returns

A dict with two members, 'type' and 'code', indicating the status of the installation.

Return type

dict

Raises

NotImplementedError – Always, for `BaseComputeResource`.

classmethod `bulk_install_by_id`(*cb, compute_resources, sensor_kit_types, config_file=None*)

Install a sensor on a list of compute resources, specified by ID.

Parameters

- **cb** (`BaseAPI`) – Reference to API object used to communicate with the server.

- **compute_resources** (*list*) – A list of dicts, each of which contains the keys ‘vcenter_uuid’ and ‘compute_resource_id’, specifying the compute resources to install sensors on.
- **sensor_kit_types** (*list*) – A list of SensorKit objects used to specify sensor types to choose from in installation.
- **config_file** (*str*) – The text of a config.ini file with a list of sensor properties to configure on installation.

Returns

A dict with two members, ‘type’ and ‘code’, indicating the status of the installation.

Return type

dict

Raises

NotImplementedError – Always, for BaseComputeResource.

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

install_sensor(*sensor_version*, *config_file=None*)

Install a sensor on this compute resource.

Parameters

- **sensor_version** (*str*) – The version number of the sensor to be used.
- **config_file** (*str*) – The text of a config.ini file with a list of sensor properties to configure on installation.

Returns

A dict with two members, ‘type’ and ‘code’, indicating the status of the installation.

Return type

dict

Raises

NotImplementedError – Always, for BaseComputeResource.

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class `AWSComputeResourceQuery`(*doc_class*, *cb*)

Bases: `BaseComputeResourceQuery`

Represents a query that is used to locate AWSComputeResource objects.

Initialize the ComputeResourceQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** (`BaseAPI`) – Reference to API object used to communicate with the server.

add_criteria(*key*, *newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None*, ***kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or `solrq.Q` object
- ****kwargs** (*dict*) – Arguments to construct a `solrq.Q` with

Returns

This Query object.

Return type

`Query`

download(*download_format=None*)

Downloads all compute resources matching the specific criteria.

Example

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import VCenterComputeResource
>>> cbc = CBCloudAPI()
>>> query = cbc.select(VCenterComputeResource).set_os_type(["UBUNTU"]).set_
    eligibility(["ELIGIBLE"])
>>> query.set_installation_status(["ERROR"])
>>> job = query.download("CSV")
>>> job.await_completion()
>>> print(job.get_output_as_string())
```

Required Permissions:

public.cloud.inventory(READ) or _API.Public.Cloud:Public.cloud.inventory:READ,
jobs.status(READ)

Parameters

download_format (*str*) – The download format to be used. Valid values are “JSON” (the default) and “CSV”.

Returns

Asynchronous job which will supply the results of the download when they’re complete.

Return type

Job

Raises

ApiError – If the format specified was not valid, or if the server did not properly return the job.

exclude_auto_scaling_group_name(*auto_scaling_group_name*)

Excludes the specified auto scaling group name from appearing in the search results.

Parameters

auto_scaling_group_name (*list*) – List of string auto scaling group names.

Returns

This instance.

Return type

AWSComputeResourceQuery

exclude_availability_zone(*availability_zone*)

Excludes the specified availability zone from appearing in the search results.

Parameters

availability_zone (*list*) – List of string availability zones.

Returns

This instance.

Return type

AWSComputeResourceQuery

exclude_cloud_provider_account_id(*cloud_provider_account_id*)

Excludes the specified cloud provider account ID from appearing in the search results.

Parameters

cloud_provider_account_id (*list*) – List of string cloud provider account IDs.

Returns

This instance.

Return type

AWSComputeResourceQuery

exclude_cloud_provider_resource_id(*cloud_provider_resource_id*)

Excludes the specified cloud provider resource ID from appearing in the search results.

Parameters

cloud_provider_resource_id (*list*) – List of string cloud provider resource IDs.

Returns

This instance.

Return type

AWSComputeResourceQuery

exclude_cloud_provider_tags(*cloud_provider_tags*)

Excludes the specified cloud provider tags from appearing in the search results.

Parameters

cloud_provider_tags (*list*) – List of string cloud provider tags.

Returns

This instance.

Return type

AWSComputeResourceQuery

exclude_id(*id_value*)

Excludes the specified compute resource ID from appearing in the search results.

Parameters

id_value (*list*) – List of string compute resource IDs.

Returns

This instance.

Return type

AWSComputeResourceQuery

exclude_installation_status(*installation_status*)

Excludes the specified installation status from appearing in the search results.

Parameters

installation_status (*list*) – List of string installation statuses.

Returns

This instance.

Return type

AWSComputeResourceQuery

exclude_name(*name*)

Excludes the specified compute resource name from appearing in the search results.

Parameters

name (*list*) – List of string compute resource names.

Returns

This instance.

Return type

AWSComputeResourceQuery

exclude_platform(platform)

Excludes the specified platform from appearing in the search results.

Parameters

platform (*list*) – List of string platforms.

Returns

This instance.

Return type

AWSComputeResourceQuery

exclude_platform_details(platform_details)

Excludes the specified platform details from appearing in the search results.

Parameters

platform_details (*list*) – List of string platform details.

Returns

This instance.

Return type

AWSComputeResourceQuery

exclude_region(region)

Excludes the specified region from appearing in the search results.

Parameters

region (*list*) – List of string regions.

Returns

This instance.

Return type

AWSComputeResourceQuery

exclude_subnet_id(subnet_id)

Excludes the specified subnet ID from appearing in the search results.

Parameters

subnet_id (*list*) – List of string subnet IDs.

Returns

This instance.

Return type

AWSComputeResourceQuery

exclude_virtual_private_cloud_id(virtual_private_cloud_id)

Excludes the specified virtual private cloud ID from appearing in the search results.

Parameters

virtual_private_cloud_id (*list*) – List of string virtual private cloud IDs.

Returns

This instance.

Return type*AWSComputeResourceQuery***execute_async()**

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

facet(fields, rows=None)

Facets all compute resources matching the specified criteria and returns the facet results.

Example

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import AWSComputeResource
>>> cbc = CBCloudAPI()
>>> query = cbc.select(AWSComputeResource)
>>> facets = query.facet(['platform', 'virtual_private_cloud_id'])
```

Required Permissions:

public.cloud.inventory(READ) or _API.Public.Cloud:Public.cloud.inventory:READ

Parameters

- **fields** (*list[str]*) – List of the fields to be faceted on.
- **rows** (*int*) – Number of the top entries to return. Default is 20.

Returns

The facet data.

Return type

list[*ComputeResourceFacet*]

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(q=None, **kwargs)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type*Query***one()**

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type*obj***Raises**

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(q=None, **kwargs)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type*Query***set_auto_scaling_group_name(auto_scaling_group_name)**

Restricts the search that this query is performed on to the specified auto scaling group name.

Parameters

auto_scaling_group_name (*list*) – List of string auto scaling group names.

Returns

This instance.

Return type*AWSComputeResourceQuery***set_availability_zone(availability_zone)**

Restricts the search that this query is performed on to the specified availability zone.

Parameters

availability_zone (*list*) – List of string availability zones.

Returns

This instance.

Return type*AWSComputeResourceQuery***set_cloud_provider_account_id(cloud_provider_account_id)**

Restricts the search that this query is performed on to the specified cloud provider account ID.

Parameters

cloud_provider_account_id (*list*) – List of string cloud provider account IDs.

Returns

This instance.

Return type

AWSComputeResourceQuery

set_cloud_provider_resource_id(*cloud_provider_resource_id*)

Restricts the search that this query is performed on to the specified cloud provider resource ID.

Parameters

cloud_provider_resource_id (*list*) – List of string cloud provider resource IDs.

Returns

This instance.

Return type

AWSComputeResourceQuery

set_cloud_provider_tags(*cloud_provider_tags*)

Restricts the search that this query is performed on to the specified cloud provider tags.

Parameters

cloud_provider_tags (*list*) – List of string cloud provider tags.

Returns

This instance.

Return type

AWSComputeResourceQuery

set_id(*id_value*)

Restricts the search that this query is performed on to the specified compute resource ID.

Parameters

id_value (*list*) – List of string compute resource IDs.

Returns

This instance.

Return type

AWSComputeResourceQuery

set_installation_status(*installation_status*)

Restricts the search that this query is performed on to the specified installation status.

Parameters

installation_status (*list*) – List of string installation statuses.

Returns

This instance.

Return type

AWSComputeResourceQuery

set_name(*name*)

Restricts the search that this query is performed on to the specified compute resource name.

Parameters

name (*list*) – List of string compute resource names.

Returns

This instance.

Return type*AWSComputeResourceQuery***set_platform(platform)**

Restricts the search that this query is performed on to the specified platform.

Parameters

platform (*list*) – List of string platforms.

Returns

This instance.

Return type*AWSComputeResourceQuery***set_platform_details(platform_details)**

Restricts the search that this query is performed on to the specified platform details.

Parameters

platform_details (*list*) – List of string platform details.

Returns

This instance.

Return type*AWSComputeResourceQuery***set_region(region)**

Restricts the search that this query is performed on to the specified region.

Parameters

region (*list*) – List of string regions.

Returns

This instance.

Return type*AWSComputeResourceQuery***set_subnet_id(subnet_id)**

Restricts the search that this query is performed on to the specified subnet ID.

Parameters

subnet_id (*list*) – List of string subnet IDs.

Returns

This instance.

Return type*AWSComputeResourceQuery***set_virtual_private_cloud_id(virtual_private_cloud_id)**

Restricts the search that this query is performed on to the specified virtual private cloud ID.

Parameters

virtual_private_cloud_id (*list*) – List of string virtual private cloud IDs.

Returns

This instance.

Return type*AWSComputeResourceQuery*

sort_by(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Example

```
>>> cb.select(ComputeResource).sort_by("name")
```

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order.

Returns

This instance.

Return type

BaseComputeResourceQuery

summarize(*summary_fields*)

Get compute resource summaries on required fields of the resources with the specified criteria.

Example

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import AWSComputeResource
>>> cbc = CBCloudAPI()
>>> query = cbc.select(AWSComputeResource)
>>> summary = query.summarize(['availability_zone', 'region', 'virtual_private_
↳ cloud_id'])
```

Required Permissions:

public.cloud.inventory(READ) or _API.Public.Cloud:Public.cloud.inventory:READ

Parameters

summary_fields (*list[str]*) – The fields to be summarized.

Returns

A mapping of field names to the number of resources with that field.

Return type

map[str, int]

update_criteria(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

class BaseComputeResource(*cb, model_unique_id, initial_data=None*)

Bases: *NewBaseModel*

Internal BaseComputeResource model

Initialize the BaseComputeResource object.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the compute resource represented.
- **initial_data** (*dict*) – Initial data used to populate the resource object.

classmethod bulk_install(*cb, compute_resources, sensor_kit_types, config_file=None*)

Install a sensor on a list of compute resources.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **compute_resources** (*list*) – A list of ComputeResource objects used to specify compute resources to install sensors on.
- **sensor_kit_types** (*list*) – A list of SensorKit objects used to specify sensor types to choose from in installation.
- **config_file** (*str*) – The text of a config.ini file with a list of sensor properties to configure on installation.

Returns

A dict with two members, 'type' and 'code', indicating the status of the installation.

Return type

dict

Raises

NotImplementedError – Always, for BaseComputeResource.

classmethod `bulk_install_by_id(cb, compute_resources, sensor_kit_types, config_file=None)`

Install a sensor on a list of compute resources, specified by ID.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **compute_resources** (*list*) – A list of dicts, each of which contains the keys ‘vcenter_uuid’ and ‘compute_resource_id’, specifying the compute resources to install sensors on.
- **sensor_kit_types** (*list*) – A list of SensorKit objects used to specify sensor types to choose from in installation.
- **config_file** (*str*) – The text of a config.ini file with a list of sensor properties to configure on installation.

Returns

A dict with two members, ‘type’ and ‘code’, indicating the status of the installation.

Return type

dict

Raises

NotImplementedError – Always, for BaseComputeResource.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

install_sensor(*sensor_version, config_file=None*)

Install a sensor on this compute resource.

Parameters

- **sensor_version** (*str*) – The version number of the sensor to be used.
- **config_file** (*str*) – The text of a config.ini file with a list of sensor properties to configure on installation.

Returns

A dict with two members, ‘type’ and ‘code’, indicating the status of the installation.

Return type

dict

Raises

NotImplementedError – Always, for BaseComputeResource.

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class BaseComputeResourceQuery(*doc_class, cb*)

Bases: [BaseQuery](#), [QueryBuilderSupportMixin](#), [CriteriaBuilderSupportMixin](#), [IterableQueryMixin](#), [AsyncQueryMixin](#)

Base class for compute resource queries, not intended for direct use.

Initialize the BaseComputeResourceQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None, **kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type*Query***download**(*download_format=None*)

Downloads all compute resources matching the specific criteria.

Example

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import VCenterComputeResource
>>> cbc = CBCloudAPI()
>>> query = cbc.select(VCenterComputeResource).set_os_type(["UBUNTU"]).set_
    eligibility(["ELIGIBLE"])
>>> query.set_installation_status(["ERROR"])
>>> job = query.download("CSV")
>>> job.await_completion()
>>> print(job.get_output_as_string())
```

Required Permissions:

public.cloud.inventory(READ) or _API.Public.Cloud:Public.cloud.inventory:READ,
jobs.status(READ)

Parameters

download_format (*str*) – The download format to be used. Valid values are “JSON” (the default) and “CSV”.

Returns

Asynchronous job which will supply the results of the download when they’re complete.

Return type*Job***Raises**

ApiError – If the format specified was not valid, or if the server did not properly return the job.

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type*Future***facet**(*fields, rows=None*)

Facets all compute resources matching the specified criteria and returns the facet results.

Example

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import AWSComputeResource
>>> cbc = CBCloudAPI()
>>> query = cbc.select(AWSComputeResource)
>>> facets = query.facet(['platform', 'virtual_private_cloud_id'])
```

Required Permissions:

public.cloud.inventory(READ) or _API.Public.Cloud:Public.cloud.inventory:READ

Parameters

- **fields** (*list[str]*) – List of the fields to be faceted on.
- **rows** (*int*) – Number of the top entries to return. Default is 20.

Returns

The facet data.

Return type

list[*ComputeResourceFacet*]

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(q=None, **kwargs)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- ***MoreThanOneResultError*** – If the query returns more than one item
- ***ObjectNotFoundError*** – If the query returns zero items

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

sort_by(*key, direction='ASC'*)

Sets the sorting behavior on a query's results.

Example

```
>>> cb.select(ComputeResource).sort_by("name")
```

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order.

Returns

This instance.

Return type

BaseComputeResourceQuery

update_criteria(*key, newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

class ComputeResourceFacet(*cb, model_unique_id, initial_data=None*)

Bases: *UnrefreshableModel*

Facet data returned by the facet() method of the query.

Initialize the ComputeResourceFacet object.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the facet represented.
- **initial_data** (*dict*) – Initial data used to populate the facet.

class ComputeResourceFacetValue(*cb, model_unique_id, initial_data=None*)

Bases: *UnrefreshableModel*

Represents a single facet value inside a ComputeResourceFacet.

Initialize the ComputeResourceFacetValue object.

Parameters

- **cb** (*BaseAPI*) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the facet value represented.
- **initial_data** (*dict*) – Initial data used to populate the facet value.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

get(*attrname*, *default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

property values

Returns the values for this particular facet.

Returns

The values of this facet.

Return type

list[[*ComputeResourceFacet.ComputeResourceFacetValue*](#)]

class VCenterComputeResource(*cb*, *model_unique_id*, *initial_data=None*)

Bases: [*BaseComputeResource*](#)

Models a vCenter compute resource.

Initialize the VCenterComputeResource object.

Parameters

- **cb** ([*BaseAPI*](#)) – Reference to API object used to communicate with the server.
- **model_unique_id** (*str*) – ID of the alert represented.
- **initial_data** (*dict*) – Initial data used to populate the alert.

classmethod bulk_install(*cb*, *compute_resources*, *sensor_kit_types*, *config_file=None*)

Install a sensor on a list of compute resources.

Parameters

- **cb** ([*BaseAPI*](#)) – Reference to API object used to communicate with the server.
- **compute_resources** (*list*) – A list of *ComputeResource* objects used to specify compute resources to install sensors on.
- **sensor_kit_types** (*list*) – A list of *SensorKit* objects used to specify sensor types to choose from in installation.

- **config_file** (*str*) – The text of a config.ini file with a list of sensor properties to configure on installation.

Returns

A dict with two members, 'type' and 'code', indicating the status of the installation.

Return type

dict

classmethod **bulk_install_by_id**(*cb, compute_resources, sensor_kit_types, config_file=None*)

Install a sensor on a list of compute resources, specified by ID.

Parameters

- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.
- **compute_resources** (*list*) – A list of dicts, each of which contains the keys 'vcenter_uuid' and 'compute_resource_id', specifying the compute resources to install sensors on.
- **sensor_kit_types** (*list*) – A list of SensorKit objects used to specify sensor types to choose from in installation.
- **config_file** (*str*) – The text of a config.ini file with a list of sensor properties to configure on installation.

Returns

A dict with two members, 'type' and 'code', indicating the status of the installation.

Return type

dict

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

install_sensor(*sensor_version, config_file=None*)

Install a sensor on this compute resource.

Parameters

- **sensor_version** (*str*) – The version number of the sensor to be used.
- **config_file** (*str*) – The text of a config.ini file with a list of sensor properties to configure on installation.

Returns

A dict with two members, 'type' and 'code', indicating the status of the installation.

Return type

dict

Raises

[ApiError](#) – If the compute node is not eligible or is of an invalid type.

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class VCenterComputeResourceQuery(*doc_class, cb*)

Bases: [BaseComputeResourceQuery](#)

Represents a query that is used to locate ComputeResource objects.

Initialize the ComputeResourceQuery.

Parameters

- **doc_class** (*class*) – The model class that will be returned by this query.
- **cb** ([BaseAPI](#)) – Reference to API object used to communicate with the server.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None, **kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

download(*download_format=None*)

Downloads all compute resources matching the specific criteria.

Example

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import VCenterComputeResource
>>> cbc = CBCloudAPI()
>>> query = cbc.select(VCenterComputeResource).set_os_type(["UBUNTU"]).set_
    eligibility(["ELIGIBLE"])
>>> query.set_installation_status(["ERROR"])
>>> job = query.download("CSV")
>>> job.await_completion()
>>> print(job.get_output_as_string())
```

Required Permissions:

public.cloud.inventory(READ) or _API.Public.Cloud:Public.cloud.inventory:READ,
jobs.status(READ)

Parameters

download_format (*str*) – The download format to be used. Valid values are “JSON” (the default) and “CSV”.

Returns

Asynchronous job which will supply the results of the download when they’re complete.

Return type

Job

Raises

ApiError – If the format specified was not valid, or if the server did not properly return the job.

exclude_appliance_uuid(*appliance_uuid*)

Excludes the specified appliance UUID from appearing in the search results.

Parameters

appliance_uuid (*list*) – List of string appliance uuids.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_cluster_name(*cluster_name*)

Excludes the specified cluster name from appearing in the search results.

Parameters

cluster_name (*list*) – List of string cluster names.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_datacenter_name(*datacenter_name*)

Excludes the specified datacenter name from appearing in the search results.

Parameters

datacenter_name (*list*) – List of string datacenter names.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_device_guid(*device_guid*)

Excludes the specified device GUID from appearing in the search results.

Parameters

device_guid (*list*) – List of string device GUIDs.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_eligibility(*eligibility*)

Excludes the specified eligibility from appearing in the search results.

Parameters

eligibility (*list*) – List of string eligibilities.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_eligibility_code(*eligibility_code*)

Excludes the specified eligibility code from appearing in the search results.

Parameters

eligibility_code (*list*) – List of string eligibility codes.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_esx_host_name(*esx_host_name*)

Excludes the specified ESX host name from appearing in the search results.

Parameters

esx_host_name (*list*) – List of string ESX host names.

Returns

This instance.

Return type*VCenterComputeResourceQuery***exclude_esx_host_uuid**(*esx_host_uuid*)

Excludes the specified ESX host UUID from appearing in the search results.

Parameters**esx_host_uuid** (*list*) – List of string ESX host UUIDs.**Returns**

This instance.

Return type*VCenterComputeResourceQuery***exclude_host_name**(*host_name*)

Excludes the specified host name from appearing in the search results.

Parameters**host_name** (*list*) – List of string host names.**Returns**

This instance.

Return type*VCenterComputeResourceQuery***exclude_installation_status**(*installation_status*)

Excludes the specified installation status from appearing in the search results.

Parameters**installation_status** (*list*) – List of string installation statuses.**Returns**

This instance.

Return type*VCenterComputeResourceQuery***exclude_installation_type**(*installation_type*)

Excludes the specified installation type from appearing in the search results.

Parameters**installation_type** (*list*) – List of string installation types.**Returns**

This instance.

Return type*VCenterComputeResourceQuery***exclude_ip_address**(*ip_address*)

Excludes the specified IP address from appearing in the search results.

Parameters**ip_address** (*list*) – List of string IP addresses.**Returns**

This instance.

Return type*VCenterComputeResourceQuery*

exclude_name(*name*)

Excludes the specified name from appearing in the search results.

Parameters

name (*list*) – List of string names.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_os_architecture(*os_architecture*)

Excludes the specified OS architecture from appearing in the search results.

Parameters

os_architecture (*list*) – List of string OS architectures.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_os_description(*os_description*)

Excludes the specified OS description from appearing in the search results.

Parameters

os_description (*list*) – List of string OS descriptions.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_os_type(*os_type*)

Excludes the specified OS type from appearing in the search results.

Parameters

os_type (*list*) – List of string OS types.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_registration_id(*registration_id*)

Excludes the specified registration ID from appearing in the search results.

Parameters

registration_id (*list*) – List of string registration IDs.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_uuid(uuid)

Excludes the specified UUID from appearing in the search results.

Parameters

uuid (*list*) – List of string UUIDs.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_vcenter_host_url(vcenter_host_url)

Excludes the specified vCenter host URL from appearing in the search results.

Parameters

vcenter_host_url (*list*) – List of string vCenter host URLs.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_vcenter_name(vcenter_name)

Excludes the specified vCenter name from appearing in the search results.

Parameters

vcenter_name (*list*) – List of string vCenter names.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_vcenter_uuid(vcenter_uuid)

Excludes the specified vCenter UUID from appearing in the search results.

Parameters

vcenter_uuid (*list*) – List of string vCenter UUIDs.

Returns

This instance.

Return type

VCenterComputeResourceQuery

exclude_vmwaretools_version(vmwaretools_version)

Excludes the specified VMware Tools version from appearing in the search results.

Parameters

vmwaretools_version (*list*) – List of string VMware Tools versions.

Returns

This instance.

Return type

VCenterComputeResourceQuery

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

facet(fields, rows=None)

Facets all compute resources matching the specified criteria and returns the facet results.

Example

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.workload import AWSComputeResource
>>> cbc = CBCloudAPI()
>>> query = cbc.select(AWSComputeResource)
>>> facets = query.facet(['platform', 'virtual_private_cloud_id'])
```

Required Permissions:

public.cloud.inventory(READ) or _API.Public.Cloud:Public.cloud.inventory:READ

Parameters

- **fields** (*list[str]*) – List of the fields to be faceted on.
- **rows** (*int*) – Number of the top entries to return. Default is 20.

Returns

The facet data.

Return type

list[*ComputeResourceFacet*]

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(q=None, **kwargs)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(q=None, **kwargs)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

set_appliance_uuid(appliance_uuid)

Restricts the search that this query is performed on to the specified appliance uuid.

Parameters

appliance_uuid (*list*) – List of string appliance uuids.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_cluster_name(cluster_name)

Restricts the search that this query is performed on to the specified cluster name.

Parameters

cluster_name (*list*) – List of string cluster names.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_datacenter_name(datacenter_name)

Restricts the search that this query is performed on to the specified datacenter name.

Parameters

datacenter_name (*list*) – List of string datacenter names.

Returns

This instance.

Return type*VCenterComputeResourceQuery***set_device_guid(device_guid)**

Restricts the search that this query is performed on to the specified device GUID.

Parameters

device_guid (*list*) – List of string device GUIDs.

Returns

This instance.

Return type*VCenterComputeResourceQuery***set_eligibility(eligibility)**

Restricts the search that this query is performed on to the specified eligibility.

Parameters

eligibility (*list*) – List of string eligibilities.

Returns

This instance.

Return type*VCenterComputeResourceQuery***set_eligibility_code(eligibility_code)**

Restricts the search that this query is performed on to the specified eligibility code.

Parameters

eligibility_code (*list*) – List of string eligibility codes.

Returns

This instance.

Return type*VCenterComputeResourceQuery***set_esx_host_name(esx_host_name)**

Restricts the search that this query is performed on to the specified ESX host name.

Parameters

esx_host_name (*list*) – List of string ESX host names.

Returns

This instance.

Return type*VCenterComputeResourceQuery***set_esx_host_uuid(esx_host_uuid)**

Restricts the search that this query is performed on to the specified ESX host UUID.

Parameters

esx_host_uuid (*list*) – List of string ESX host UUIDs.

Returns

This instance.

Return type*VCenterComputeResourceQuery*

set_host_name(*host_name*)

Restricts the search that this query is performed on to the specified host name.

Parameters

host_name (*list*) – List of string host names.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_installation_status(*installation_status*)

Restricts the search that this query is performed on to the specified installation status.

Parameters

installation_status (*list*) – List of string installation status.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_installation_type(*installation_type*)

Restricts the search that this query is performed on to the specified installation type.

Parameters

installation_type (*list*) – List of string installation types.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_ip_address(*ip_address*)

Restricts the search that this query is performed on to the specified ip address.

Parameters

ip_address (*list*) – List of string ip addresses.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_name(*name*)

Restricts the search that this query is performed on to the specified name.

Parameters

name (*list*) – List of string names.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_os_architecture(*os_architecture*)

Restricts the search that this query is performed on to the specified os architecture.

Parameters

os_architecture (*list*) – List of string os architecture.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_os_description(*os_description*)

Restricts the search that this query is performed on to the specified os description.

Parameters

os_description (*list*) – List of string os description.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_os_type(*os_type*)

Restricts the search that this query is performed on to the specified os type.

Parameters

os_type (*list*) – List of string os type.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_registration_id(*registration_id*)

Restricts the search that this query is performed on to the specified registration ID.

Parameters

registration_id (*list*) – List of string registration IDs.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_uuid(*uuid*)

Restricts the search that this query is performed on to the specified uuid.

Parameters

uuid (*list*) – List of string uuid.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_vcenter_host_url(*vcenter_host_url*)

Restricts the search that this query is performed on to the specified vCenter host URL.

Parameters

vcenter_host_url (*list*) – List of string vCenter host URLs.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_vcenter_name(*vcenter_name*)

Restricts the search that this query is performed on to the specified vCenter name.

Parameters

vcenter_name (*list*) – List of string vCenter names.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_vcenter_uuid(*vcenter_uuid*)

Restricts the search that this query is performed on to the specified vCenter UUID.

Parameters

vcenter_uuid (*list*) – List of string vCenter UUIDs.

Returns

This instance.

Return type

VCenterComputeResourceQuery

set_vmwaretools_version(*vmwaretools_version*)

Restricts the search that this query is performed on to the specified VMware Tools version.

Parameters

vmwaretools_version (*list*) – List of string VMware Tools versions.

Returns

This instance.

Return type

VCenterComputeResourceQuery

sort_by(*key*, *direction*='ASC')

Sets the sorting behavior on a query's results.

Example

```
>>> cb.select(ComputeResource).sort_by("name")
```

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order.

Returns

This instance.

Return type

BaseComputeResourceQuery

update_criteria(*key, newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, QueryBuilder, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

```
log = <Logger cbc_sdk.workload.vm_workloads_search (WARNING)>
```

Workloads Search model

4.13 CBC SDK Package

4.13.1 Subpackages

Cache Package

LRU Module

LRU cache based on stucchio's py-lru-cache module

original copy at <https://github.com/stucchio/Python-LRU-cache> licensed under MIT

```
class LRUCacheDict(max_size=1024, expiration=900, thread_clear=False, concurrent=True)
```

Bases: object

A dictionary-like object, supporting LRU caching semantics.

```
>>> d = LRUCacheDict(max_size=3, expiration=3)
>>> d['foo'] = 'bar'
>>> d['foo']
'bar'
>>> import time
>>> time.sleep(4) # 4 seconds > 3 second cache expiry of d
>>> d['foo']
Traceback (most recent call last):
...
KeyError: 'foo'
>>> d['a'] = 'A'
>>> d['b'] = 'B'
>>> d['c'] = 'C'
>>> d['d'] = 'D'
>>> d['a'] # Should return value error, since we exceeded the max cache size
Traceback (most recent call last):
...
KeyError: 'a'
```

By default, this cache will only expire items whenever you poke it - all methods on this class will result in a cleanup. If the `thread_clear` option is specified, a background thread will clean it up every `thread_clear_min_check` seconds.

If this class must be used in a multithreaded environment, the option `concurrent` should be set to true. Note that the cache will always be concurrent if a background cleanup thread is used.

Initialize the LRUCacheDict object.

Parameters

- **max_size** (*int*) – Maximum number of elements in the cache.
- **expiration** (*int*) – Number of seconds an item can be in the cache before it expires.
- **thread_clear** (*bool*) – True if we want to use a background thread to keep the cache clear.
- **concurrent** (*bool*) – True to make access to the cache thread-safe.

```
class EmptyCacheThread(cache, peek_duration=60)
```

Bases: Thread

Background thread that expires elements out of the cache.

Initialize the EmptyCacheThread.

Parameters

- **cache** (`LRUCacheDict`) – The cache to be monitored.
- **peek_duration** (`int`) – The delay between “sweeps” of the cache.

getName()

Return a string used for identification purposes only.

This method is deprecated, use the name attribute instead.

property ident

Thread identifier of this thread or None if it has not been started.

This is a nonzero integer. See the `get_ident()` function. Thread identifiers may be recycled when a thread exits and another thread is created. The identifier is available even after the thread has exited.

isDaemon()

Return whether this thread is a daemon.

This method is deprecated, use the daemon attribute instead.

is_alive()

Return whether the thread is alive.

This method returns True just before the `run()` method starts until just after the `run()` method terminates. See also the module function `enumerate()`.

join(timeout=None)

Wait until the thread terminates.

This blocks the calling thread until the thread whose `join()` method is called terminates – either normally or through an unhandled exception or until the optional timeout occurs.

When the timeout argument is present and not None, it should be a floating point number specifying a timeout for the operation in seconds (or fractions thereof). As `join()` always returns None, you must call `is_alive()` after `join()` to decide whether a timeout happened – if the thread is still alive, the `join()` call timed out.

When the timeout argument is not present or None, the operation will block until the thread terminates.

A thread can be `join()`ed many times.

`join()` raises a `RuntimeError` if an attempt is made to join the current thread as that would cause a deadlock. It is also an error to `join()` a thread before it has been started and attempts to do so raises the same exception.

property name

A string used for identification purposes only.

It has no semantics. Multiple threads may be given the same name. The initial name is set by the constructor.

property native_id

Native integral thread ID of this thread, or None if it has not been started.

This is a non-negative integer. See the `get_native_id()` function. This represents the Thread ID as reported by the kernel.

run()

Execute the background cleanup.

setDaemon(daemonic)

Set whether this thread is a daemon.

This method is deprecated, use the `.daemon` property instead.

setName(name)

Set the name string for this thread.

This method is deprecated, use the `name` attribute instead.

start()

Start the thread's activity.

It must be called at most once per thread object. It arranges for the object's `run()` method to be invoked in a separate thread of control.

This method will raise a `RuntimeError` if called more than once on the same thread object.

class LRUCachedFunction(function, cache=None)

Bases: `object`

A memoized function, backed by an LRU cache.

```
>>> def f(x):
...     print "Calling f(" + str(x) + ")"
...     return x
>>> f = LRUCachedFunction(f, LRUCacheDict(max_size=3, expiration=3) )
>>> f(3)
Calling f(3)
3
>>> f(3)
3
>>> import time
>>> time.sleep(4) #Cache should now be empty, since expiration time is 3.
>>> f(3)
Calling f(3)
3
>>> f(4)
Calling f(4)
4
>>> f(5)
Calling f(5)
5
>>> f(3) #Still in cache, so no print statement. At this point, 4 is the least_
↪recently used.
3
>>> f(6)
Calling f(6)
6
>>> f(4) #No longer in cache - 4 is the least recently used, and there are at least_
↪3 others
items in cache [3,4,5,6].
Calling f(4)
4
```

Initialize the LRUCachedFunction object.

Parameters

- **function** (*func*) – The function to be used to create new items in the cache.
- **cache** (*LRUCacheDict*) – The internal cache structure.

lru_cache_function(*max_size=1024, expiration=900*)

Least recently used cache function

```
>>> @lru_cache_function(3, 1)
... def f(x):
...     print "Calling f(" + str(x) + ")"
...     return x
>>> f(3)
Calling f(3)
3
>>> f(3)
3
```

4.13.2 Submodules

Base Module

Models and Queries for the Base Carbon Black Cloud SDK

class ArrayFieldDescriptor(*field_name, coerce_to=None, default_value=None*)

Bases: *FieldDescriptor*

Field descriptor for fields of ‘array’ type.

Initialize the FieldDescriptor object.

Parameters

- **field_name** (*str*) – The name of the field.
- **coerce_to** (*class*) – The type to which the value should be coerced, or None.
- **default_value** (*Any*) – The default value of the field.

class AsyncQueryMixin

Bases: *object*

A mix-in which provides support for asynchronous queries.

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

class BaseQuery(*query=None*)

Bases: *object*

The base query for finding objects via the API.

Initializes the BaseQuery object.

Parameters

query (*solrq.Q*) – The parent query of this one.

class BinaryFieldDescriptor(*field_name, coerce_to=None, default_value=None*)

Bases: *FieldDescriptor*

Field descriptor for fields of 'byte' type.

Initialize the FieldDescriptor object.

Parameters

- **field_name** (*str*) – The name of the field.
- **coerce_to** (*class*) – The type to which the value should be coerced, or None.
- **default_value** (*Any*) – The default value of the field.

class CbMetaModel(*name, bases, clsdict*)

Bases: *type*

Meta-model for NewBaseModel and its subclasses.

Creates a new instance of a class, setting up the field descriptors based on the metafile.

Parameters

- **name** (*str*) – The name of the class.
- **bases** (*list*) – Base classes of the class to be created.
- **clsdict** (*dict*) – Elements defined in the new class.

mro()

Return a type's method resolution order.

class CreatableModelMixin

Bases: *object*

Mixin for all objects which are creatable.

class CriteriaBuilderSupportMixin

Bases: *object*

A mixin that supplies wrapper methods to access the criteria.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

update_criteria(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

class EpochDateTimeFieldDescriptor(*field_name*, *multiplier=1.0*)

Bases: [FieldDescriptor](#)

Field descriptor for fields of ‘epoch-ms-date-time’ type.

Initialize the EpochDateTimeFieldDescriptor object.

Parameters

- **field_name** (*str*) – The name of the field.
- **multiplier** (*float*) – Unused.

class ExclusionBuilderSupportMixin

Bases: `object`

A mixin that supplies wrapper methods to access the exclusions.

add_exclusions(*key*, *newlist*)

Add to the exclusions on this query with a custom exclusions key.

Will overwrite any existing exclusion for the specified key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*str* or *list[str]*) – Value or list of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).add_exclusions("type", ["WATCHLIST"])
>>> query = api.select(Alert).add_exclusions("type", "WATCHLIST")
```

update_exclusions(key, newlist)

Update the exclusion on this query with a custom exclusion key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*list*) – List of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).update_exclusions("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

class FacetQuery(cls, cb, query=None)

Bases: *BaseQuery*, *AsyncQueryMixin*, *QueryBuilderSupportMixin*, *CriteriaBuilderSupportMixin*, *ExclusionBuilderSupportMixin*

Query class for asynchronous Facet API calls.

These API calls return one result, and are not paginated or iterable.

Initialize the FacetQuery object.

add_criteria(key, newlist)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*str or list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

add_exclusions(*key*, *newlist*)

Add to the exclusions on this query with a custom exclusions key.

Will overwrite any existing exclusion for the specified key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*str* or *list[str]*) – Value or list of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).add_exclusions("type", ["WATCHLIST"])
>>> query = api.select(Alert).add_exclusions("type", "WATCHLIST")
```

add_facet_field(*field*)

Sets the facet fields to be received by this query.

Parameters

field (*str* or *[str]*) – Field(s) to be received.

Returns

The Query object that will receive the specified field(s).

Return type

Query (AsyncQuery)

Example

```
>>> cb.select(ProcessFacet).add_facet_field(["process_name", "process_username
↪"])
```

add_range(*range*)

Sets the facet ranges to be received by this query.

Parameters

range (*dict* or *[dict]*) – Range(s) to be received.

Returns

The Query object that will receive the specified range(s).

Return type

Query (AsyncQuery)

Note: The range parameter must be in this dictionary format:

```
{
```

```
    "bucket_size": "<object>",
    "start": "<object>",
    "end": "<object>",
    "field": "<string>"
},
```

where "bucket_size", "start", and "end" can be numbers or ISO 8601 timestamps.

Examples

```
>>> cb.select(ProcessFacet).add_range({"bucket_size": 5, "start": 0, "end": 10,
→ "field": "netconn_count"})
>>> cb.select(ProcessFacet).add_range({"bucket_size": "+1DAY", "start": "2020-
→ 11-01T00:00:00Z",
... "end": "2020-11-12T00:00:00Z", "field": "backend_timestamp"})
```

and_(*q=None*, ***kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

limit(*limit*)

Sets the maximum number of facets per category (i.e. any Process Search Fields in *self._fields*).

The default limit for Process Facet searches in the Carbon Black Cloud backend is 100.

Parameters

limit (*int*) – Maximum number of facets per category.

Returns

The Query object with new limit parameter.

Return type

Query (AsyncQuery)

Example

```
>>> cb.select(ProcessFacet).where(process_name="foo.exe").limit(50)
```

not_(*q=None, **kwargs*)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

property results

Save query results to `self._results` with `self._search()` method.

set_rows(*rows*)

Sets the number of facet results to return with the query.

Parameters

rows (*int*) – Number of rows to return.

Returns

The Query object with the new rows parameter.

Return type

Query (*AsyncQuery*)

Example

```
>>> cb.select(ProcessFacet).set_rows(50)
```

set_time_range(*start=None, end=None, window=None*)

Sets the 'time_range' query body parameter, determining a time window based on 'device_timestamp'.

Parameters

- **start** (*str in ISO 8601 timestamp*) – When to start the result search.
- **end** (*str in ISO 8601 timestamp*) – When to end the result search.

- **window** (*str*) – Time window to execute the result search, ending on the current time.
- **"-2w"** (*Should be in the form*) –
- **y=year** (*where*) –
- **w=week** –
- **d=day** –
- **h=hour** –
- **m=minute** –
- **s=second.** –

Note:

- *window* will take precedent over *start* and *end* if provided.

Examples

```
>>> query = api.select(Process).set_time_range(start="2020-10-20T20:34:07Z").
↳ where("query is required")
>>> second_query = api.select(Process).
...     set_time_range(start="2020-10-20T20:34:07Z", end="2020-10-30T20:34:07Z
↳ ").where("query is required")
>>> third_query = api.select(Process).set_time_range(window='-3d').where("query_
↳ is required")
```

timeout (*msecs*)

Sets the timeout on an AsyncQuery.

Parameters

msecs (*int*) – Timeout duration, in milliseconds. This value can never be greater than the configured default timeout. If this is 0, the configured default timeout value is used.

Returns

The Query object with new milliseconds parameter.

Return type

Query (AsyncQuery)

Example

```
>>> cb.select(ProcessFacet).where(process_name="foo.exe").timeout(5000)
```

update_criteria (*key, newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

`update_exclusions(key, newlist)`

Update the exclusion on this query with a custom exclusion key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*list*) – List of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).update_exclusions("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

`where(q=None, **kwargs)`

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, [QueryBuilder](#), or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

[Query](#)

`class FieldDescriptor(field_name, coerce_to=None, default_value=None)`

Bases: `object`

Object that describes a field within a model instance.

Initialize the FieldDescriptor object.

Parameters

- **field_name** (*str*) – The name of the field.
- **coerce_to** (*class*) – The type to which the value should be coerced, or `None`.
- **default_value** (*Any*) – The default value of the field.

class ForeignKeyFieldDescriptor(*field_name*, *join_model*, *join_field=None*)

Bases: [*FieldDescriptor*](#)

Field descriptor for fields that are foreign keys.

Initialize the ForeignKeyFieldDescriptor object.

Parameters

- **field_name** (*str*) – The name of the field.
- **join_model** (*class*) – The class for which this field value is a foreign key.
- **join_field** (*str*) – The name fo the field in the joined class for which this field value is a foreign key.

class IsoDateTimeFieldDescriptor(*field_name*)

Bases: [*FieldDescriptor*](#)

Field descriptor for fields of ‘iso-date-time’ type.

Initialize the IsoDateTimeFieldDescriptor object.

Parameters

- **field_name** (*str*) – The name of the field.

class IterableQueryMixin

Bases: object

A mix-in to provide iterability to a query.

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- [*MoreThanOneResultError*](#) – If the query returns more than one item
- [*ObjectNotFoundError*](#) – If the query returns zero items

class MutableBaseModel(*cb, model_unique_id=None, initial_data=None, force_init=False, full_doc=False*)

Bases: [*NewBaseModel*](#)

Base model for objects that can have properties changed and then saved back to the server.

Initialize the NewBaseModel object.

Parameters

- **cb** ([*CBCloudAPI*](#)) – A reference to the CBCloudAPI object.
- **model_unique_id** (*Any*) – The unique ID for this particular instance of the model object.
- **initial_data** (*dict*) – The data to use when initializing the model object.
- **force_init** (*bool*) – True to force object initialization.
- **full_doc** (*bool*) – True to mark the object as fully initialized.

delete()

Delete this object.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

is_dirty()

Returns whether or not any fields of this object have been changed.

Returns

True if any fields of this object have been changed, False if not.

Return type

bool

refresh()

Reload this object from the server.

reset()

Undo any changes made to this object's fields.

save()

Save any changes made to this object's fields.

Returns

This object.

Return type

[*MutableBaseModel*](#)

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

touch(*fulltouch=False*)

Force this object to be considered as changed.

validate()

Validates this object.

Returns

True if the object is validated.

Return type

bool

Raises

InvalidObjectError – If the object has missing fields.

class NewBaseModel(*cb, model_unique_id=None, initial_data=None, force_init=False, full_doc=False*)

Bases: object

Base class of all model objects within the Carbon Black Cloud SDK.

Initialize the NewBaseModel object.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.
- **model_unique_id** (*Any*) – The unique ID for this particular instance of the model object.
- **initial_data** (*dict*) – The data to use when initializing the model object.
- **force_init** (*bool*) – True to force object initialization.
- **full_doc** (*bool*) – True to mark the object as fully initialized.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

class ObjectFieldDescriptor(*field_name, coerce_to=None, default_value=None*)

Bases: *FieldDescriptor*

Field descriptor for fields of 'object' type.

Initialize the FieldDescriptor object.

Parameters

- **field_name** (*str*) – The name of the field.
- **coerce_to** (*class*) – The type to which the value should be coerced, or None.
- **default_value** (*Any*) – The default value of the field.

class PaginatedQuery(*cls, cb, query=None*)

Bases: *BaseQuery, IterableQueryMixin*

A query that returns objects in a paginated fashion.

Initialize the PaginatedQuery object.

Parameters

- **cls** (*class*) – The class of objects being returned by this query.
- **cb** (*CBCloudAPI*) – Reference to the CBCloudAPI object.
- **query** (*BaseQuery*) – The query that we are paginating.

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

batch_size(*new_batch_size*)

Set the batch size of the paginated query.

Parameters

new_batch_size (*int*) – The new batch size.

Returns

A new query with the updated batch size.

Return type

PaginatedQuery

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- ***MoreThanOneResultError*** – If the query returns more than one item
- ***ObjectNotFoundError*** – If the query returns zero items

class Query(*doc_class, cb*)

Bases: *PaginatedQuery, QueryBuilderSupportMixin, IterableQueryMixin, AsyncQueryMixin, CriteriaBuilderSupportMixin, ExclusionBuilderSupportMixin*

Represents a prepared query to the Carbon Black Cloud.

This object is returned as part of a *CBCCloudAPI.select* operation on models requested from the Carbon Black Cloud backend. You should not have to create this class yourself.

The query is not executed on the server until it's accessed, either as an iterator (where it will generate values on demand as they're requested) or as a list (where it will retrieve the entire result set and save to a list). You can also call the Python built-in `len()` on this object to retrieve the total number of items matching the query.

```
>>> from cbc_sdk import CBCloudAPI
>>> from cbc_sdk.enterprise_edr import Report
>>> cb = CBCloudAPI()
>>> query = cb.select(Report)
>>> query = query.where(report_id="ABCDEF1234")
>>> # alternatively:
>>> query = query.where("report_id:ABCDEF1234")
```

Notes

- The slicing operator only supports start and end parameters, but not step. `[1:-1]` is legal, but `[1:2:-1]` is not.
- You can chain where clauses together to create AND queries; only objects that match all where clauses will be returned.

Initialize the Query object.

Parameters

- **doc_class** (*class*) – The class of the model this query returns.
- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.

add_criteria(*key, newlist*)

Add to the criteria on this query with a custom criteria key.

Will overwrite any existing criteria for the specified key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.

- **newlist** (*str* or *list[str]*) – Value or list of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).add_criteria("type", ["CB_ANALYTIC", "WATCHLIST"])
>>> query = api.select(Alert).add_criteria("type", "CB_ANALYTIC")
```

add_exclusions(*key*, *newlist*)

Add to the exclusions on this query with a custom exclusions key.

Will overwrite any existing exclusion for the specified key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*str* or *list[str]*) – Value or list of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).add_exclusions("type", ["WATCHLIST"])
>>> query = api.select(Alert).add_exclusions("type", "WATCHLIST")
```

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*q=None*, ***kwargs*)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

batch_size(*new_batch_size*)

Set the batch size of the paginated query.

Parameters

new_batch_size (*int*) – The new batch size.

Returns

A new query with the updated batch size.

Return type

PaginatedQuery

execute_async()

Executes the current query in an asynchronous fashion.

Returns

A future representing the query and its results.

Return type

Future

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

not_(q=None, **kwargs)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

or_(q=None, **kwargs)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type*Query***set_fields(fields)**

Sets the fields to be returned with the response.

Parameters

fields (*str* or *list[str]*) – Field or list of fields to be returned.

set_rows(rows)

Sets the ‘rows’ query body parameter, determining how many rows of results to request.

Parameters

rows (*int*) – How many rows to request.

set_start(start)

Sets the ‘start’ query body parameter, determining where to begin retrieving results from.

Parameters

start (*int*) – Where to start results from.

set_time_range(start=None, end=None, window=None)

Sets the ‘time_range’ query body parameter, determining a time window based on ‘device_timestamp’.

Parameters

- **start** (*str* in ISO 8601 timestamp) – When to start the result search.
- **end** (*str* in ISO 8601 timestamp) – When to end the result search.
- **window** (*str*) – Time window to execute the result search, ending on the current time. Should be in the form “-2w”, where y=year, w=week, d=day, h=hour, m=minute, s=second.

Note:

- *window* will take precedent over *start* and *end* if provided.
-

Examples

```
>>> query = api.select(Process).set_time_range(start="2020-10-20T20:34:07Z").
↳ where("query is required")
>>> second_query = api.select(Process).
...     set_time_range(start="2020-10-20T20:34:07Z", end="2020-10-30T20:34:07Z
↳ ").where("query is required")
>>> third_query = api.select(Process).set_time_range(window='-3d').where("query_
↳ is required")
```

sort_by(key, direction='ASC')

Sets the sorting behavior on a query’s results.

Parameters

- **key** (*str*) – The key in the schema to sort by.
- **direction** (*str*) – The sort order, either “ASC” or “DESC”.

Returns

The query with sorting parameters.

Return type*Query***Example**

```
>>> cb.select(Process).where(process_name="cmd.exe").sort_by("device_timestamp")
```

update_criteria(*key*, *newlist*)

Update the criteria on this query with a custom criteria key.

Parameters

- **key** (*str*) – The key for the criteria item to be set.
- **newlist** (*list*) – List of values to be set for the criteria item.

Returns

The query object with specified custom criteria.

Example

```
>>> query = api.select(Alert).update_criteria("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

update_exclusions(*key*, *newlist*)

Update the exclusion on this query with a custom exclusion key.

Parameters

- **key** (*str*) – The key for the exclusion item to be set.
- **newlist** (*list*) – List of values to be set for the exclusion item.

Returns

The query object with specified custom exclusion.

Example

```
>>> query = api.select(Alert).update_exclusions("my.criteria.key", ["criteria_
↪value"])
```

Note: Use this method if there is no implemented method for your desired criteria.

where(*q=None*, ***kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, *QueryBuilder*, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

class QueryBuilder(kwargs)**

Bases: object

Provides a flexible interface for building prepared queries for the CB Cloud backend.

This object can be instantiated directly, or can be managed implicitly through the CBCloudAPI.select API.

Examples

```
>>> from cbc_sdk.base import QueryBuilder
>>> # build a query with chaining
>>> query = QueryBuilder().where(process_name="malicious.exe").and_(device_name=
↳ "suspect")
>>> # start with an initial query, and chain another condition to it
>>> query = QueryBuilder(device_os="WINDOWS").or_(process_username="root")
```

Initialize the QueryBuilder object.

Parameters

****kwargs** (*dict*) – If present, these are used to construct a SolrQ Query.

and_(*q*, **kwargs)

Adds a conjunctive filter to a QueryBuilder.

Parameters

- **q** (*object*) – Either a string or solr.Q object representing the query to be added.
- ****kwargs** (*dict*) – Arguments with which to construct a solr.Q object.

Returns

This object.

Return type

QueryBuilder

Raises

ApiError – If the q parameter is of an invalid type.

not_(*q*, **kwargs)

Adds a negative filter to a QueryBuilder.

Parameters

- **q** (*object*) – Either a string or solr.Q object representing the query to be added.
- ****kwargs** (*dict*) – Arguments with which to construct a solr.Q object.

Returns

This object.

Return type

QueryBuilder

Raises

ApiError – If the q parameter is of an invalid type.

or_(*q*, ****kwargs**)

Adds a disjunctive filter to a QueryBuilder.

Parameters

- **q** (*object*) – Either a string or solrq.Q object representing the query to be added.
- ****kwargs** (*dict*) – Arguments with which to construct a solrq.Q object.

Returns

This object.

Return type

QueryBuilder

Raises

ApiError – If the q parameter is of an invalid type.

where(*q*, ****kwargs**)

Adds a conjunctive filter to a QueryBuilder.

Parameters

- **q** (*object*) – Either a string or solrq.Q object representing the query to be added.
- ****kwargs** (*dict*) – Arguments with which to construct a solrq.Q object.

Returns

This object.

Return type

QueryBuilder

Raises

ApiError – If the q parameter is of an invalid type.

class QueryBuilderSupportMixin

Bases: object

A mixin that supplies wrapper methods to access the `_query_builder`.

and_(*q=None*, ****kwargs**)

Add a conjunctive filter to this query.

Parameters

- **q** (*Any*) – Query string or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

not_(*q=None*, ****kwargs**)

Adds a negated filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

or_(*q=None, **kwargs*)

Add a disjunctive filter to this query.

Parameters

- **q** (*solrq.Q*) – Query object.
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with.

Returns

This Query object.

Return type

Query

where(*q=None, **kwargs*)

Add a filter to this query.

Parameters

- **q** (*Any*) – Query string, *QueryBuilder*, or *solrq.Q* object
- ****kwargs** (*dict*) – Arguments to construct a *solrq.Q* with

Returns

This Query object.

Return type

Query

class SimpleQuery(*cls, cb, urlobject=None, returns_fulldoc=True*)

Bases: *BaseQuery*, *IterableQueryMixin*

A simple query object.

Initialize the SimpleQuery object.

Parameters

- **cls** (*class*) – Class of the object to be returned by the query.
- **cb** (*CBCloudAPI*) – Reference to the CBCloudAPI object.
- **urlobject** (*str*) – URL to be used in making the query.
- **returns_fulldoc** (*bool*) – Whether the result of the Query yields objects that have been fully initialized.

all()

Returns all the items of a query as a list.

Returns

List of query items

Return type

list

and_(*new_query*)

Add an additional “where” clause to this query.

Parameters

new_query (*object*) – The additional “where” clause, as a string or solrq.Q object.

Returns

A new query with the extra “where” clause specified.

Return type

SimpleQuery

first()

Returns the first item that would be returned as the result of a query.

Returns

First query item

Return type

obj

one()

Returns the only item that would be returned by a query.

Returns

Sole query return item

Return type

obj

Raises

- *MoreThanOneResultError* – If the query returns more than one item
- *ObjectNotFoundError* – If the query returns zero items

property results

Collect and return the results of this query.

Returns

The results of this query.

Return type

list

sort(*new_sort*)

Set the sorting for this query.

Parameters

new_sort (*object*) – The new sort criteria for this query.

Returns

A new query with the sort parameter specified.

Return type

SimpleQuery

where(*new_query*)

Add a “where” clause to this query.

Parameters

new_query (*object*) – The “where” clause, as a string or solrq.Q object.

Returns

A new query with the “where” clause specified.

Return type

SimpleQuery

class SwaggerLoader(*stream*)

Bases: `SafeLoader`

YAML loader class for loading Swagger metafiles.

Initialize the scanner.

check_state_key(*key*)

Block special attributes/methods from being set in a newly created object, to prevent user-controlled methods from being called during deserialization

class UnrefreshableModel(*cb, model_unique_id=None, initial_data=None, force_init=False, full_doc=False*)

Bases: *NewBaseModel*

Represents a model that can’t be refreshed, i.e. for which `reset()` is not a valid operation.

Initialize the NewBaseModel object.

Parameters

- **cb** (*CBCloudAPI*) – A reference to the CBCloudAPI object.
- **model_unique_id** (*Any*) – The unique ID for this particular instance of the model object.
- **initial_data** (*dict*) – The data to use when initializing the model object.
- **force_init** (*bool*) – True to force object initialization.
- **full_doc** (*bool*) – True to mark the object as fully initialized.

get(*attrname, default_val=None*)

Return an attribute of this object.

Parameters

- **attrname** (*str*) – Name of the attribute to be returned.
- **default_val** (*Any*) – Default value to be used if the attribute is not set.

Returns

The returned attribute value, which may be defaulted.

Return type

Any

refresh()

Reload this object from the server.

to_json()

Return a json object of the response.

Returns

The response dictionary representation.

Return type

Any

construct_include(*loader, node*)

Include the file referenced by the node.

Parameters

- **loader** (*yaml.Loader*) – YAML loader object.
- **node** (*yaml.Node*) – Current node being loaded.

Returns

The data to be included in the YAML loader output.

Return type

Any

log = <Logger cbc_sdk.base (WARNING)>

Base Models

Connection Module

Manages the CBC SDK connection to the server.

class BaseAPI(*args, **kwargs)

Bases: object

The base API object used by all CBC SDK objects to communicate with the server.

This class is not used directly, but most commonly via the CBCCloudAPI class.

Initialize the base API information.

Parameters

- ***args** (*list*) – Unused.
- ****kwargs** (*dict*) – Additional arguments.

Keyword Arguments

- **credential_file** (*str*) – The name of a credential file to be used by the default credential provider.
- **credential_provider** (*cbc_sdk.credentials.CredentialProvider*) – An alternate credential provider to use to find the credentials to be used when accessing the Carbon Black Cloud.
- **csp_api_token** (*str*) – The CSP API Token for Carbon Black Cloud.
- **csp_oauth_app_id** (*str*) – The CSP OAuth App ID for Carbon Black Cloud.
- **csp_oauth_app_secret** (*str*) – The CSP OAuth App Secret for Carbon Black Cloud.
- **integration_name** (*str*) – The name of the integration using this connection. This should be specified as a string in the format ‘name/version’
- **max_retries** (*int*) – The maximum number of times to retry failing API calls. Default is 5.
- **org_key** (*str*) – The organization key value to use when accessing the Carbon Black Cloud.
- **pool_block** (*bool*) – True if the connection pool should block when no free connections are available. Default is False.
- **pool_connections** (*int*) – Number of HTTP connections to be pooled for this instance. Default is 1.

- **pool_maxsize** (*int*) – Maximum size of the connection pool. Default is 10.
- **profile** (*str*) – Use the credentials in the named profile when connecting to the Carbon Black Cloud server. Uses the profile named ‘default’ when not specified.
- **proxy_session** (*requests.session.Session*) – Proxy session to be used for cookie persistence, connection pooling, and configuration. Default is *None* (use the standard session).
- **timeout** (*float*) – The timeout to use for API request connections. Default is *None* (no timeout).
- **token** (*str*) – The API token to use when accessing the Carbon Black Cloud.
- **url** (*str*) – The URL of the Carbon Black Cloud provider to use.

api_json_request(*method, uri, **kwargs*)

Submit a request to the server.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **method** (*str*) – HTTP method to use.
- **uri** (*str*) – URI to submit the request to.
- ****kwargs** (*dict*) – Additional arguments.

Keyword Arguments

- **data** (*object*) – Body data to be passed to the request, formatted as JSON.
- **headers** (*dict*) – Header names and values to pass to the request.

Returns

Result of the operation, as JSON

Return type

object

Raises

[*ServerError*](#) – If there’s an error output from the server.

api_request_iterate(*method, uri, **kwargs*)

Submit a request to the specified URI and iterate over the response as lines of text.

Should only be used for requests that can be expressed as large amounts of text that can be broken into lines.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **method** (*str*) – HTTP method to use.
- **uri** (*str*) – The URI to send the request to.
- ****kwargs** (*dict*) – Additional arguments for the request.

Keyword Arguments

- **data** (*object*) – Body data to be passed to the request, formatted as JSON.
- **headers** (*dict*) – Header names and values to pass to the request.

Yields

str – Each line of text in the returned data.

api_request_stream(*method*, *uri*, *stream_output*, ***kwargs*)

Submit a request to the specified URI and stream the results back into the given stream object.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **method** (*str*) – HTTP method to use.
- **uri** (*str*) – The URI to send the request to.
- **stream_output** (*RawIOBase*) – The output stream to write the data to.
- ****kwargs** (*dict*) – Additional arguments for the request.

Keyword Arguments

- **data** (*object*) – Body data to be passed to the request, formatted as JSON.
- **headers** (*dict*) – Header names and values to pass to the request.

Returns

The return data from the request.

Return type

object

create(*cls*, *data=None*)

Create a new object of a Model class.

Parameters

- **cls** (*class*) – The Model class (only some models can be created, for example, Feed, Notification, ...)
- **data** (*object*) – The data used to initialize the new object.

Returns

An empty instance of the model class.

Return type

Model

Raises

ApiError – If the Model cannot be created.

delete_object(*uri*)

Send a DELETE request to the specified URI.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

uri (*str*) – The URI to send the DELETE request to.

Returns

The return data from the DELETE request, as JSON.

Return type

object

get_object(*uri*, *query_parameters=None*, *default=None*)

Submit a GET request to the server and parse the result as JSON before returning.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **uri** (*str*) – The URI to send the GET request to.
- **query_parameters** (*dict*) – Parameters for the query.
- **default** (*object*) – What gets returned in the event of an empty response.

Returns

Result of the GET request, as JSON.

Return type

object

get_raw_data(*uri*, *query_parameters=None*, *default=None*, ***kwargs*)

Submit a GET request to the server and return the result without parsing it.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **uri** (*str*) – The URI to send the GET request to.
- **query_parameters** (*dict*) – Parameters for the query.
- **default** (*object*) – What gets returned in the event of an empty response.
- ****kwargs** (*dict*) – Additional arguments.

Keyword Arguments

headers (*dict*) – Header names and values to pass to the GET request.

Returns

Result of the GET request.

Return type

object

post_multipart(*uri*, *param_table*, ***kwargs*)

Send a POST request to the specified URI, with parameters sent as multipart/form-data.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **uri** (*str*) – The URI to send the POST request to.
- **param_table** (*dict*) – A dict of known parameters to the underlying method, each element of which is a parameter name mapped to a dict, which contains elements ‘filename’ and ‘type’ representing the pseudo-filename to be used for the data and the MIME type of the data.
- ****kwargs** (*dict*) – Arguments to pass to the API. Except for “headers,” these will all be added as parameters to the form data sent.

Keyword Arguments

headers (*dict*) – Header names and values to pass to the request.

Returns

The return data from the POST request.

Return type

object

post_object(*uri*, *body*, ***kwargs*)

Send a POST request to the specified URI.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **uri** (*str*) – The URI to send the POST request to.
- **body** (*object*) – The data to be sent in the body of the POST request, as JSON.
- ****kwargs** (*dict*) – Additional arguments for the HTTP POST.

Keyword Arguments

headers (*dict*) – Header names and values to pass to the request.

Returns

The return data from the POST request, as JSON.

Return type

object

put_object(*uri*, *body*, ***kwargs*)

Send a PUT request to the specified URI.

Normally only used by other SDK objects; used from user code only to submit a request to the server that is not currently implemented in the SDK.

Parameters

- **uri** (*str*) – The URI to send the PUT request to.
- **body** (*object*) – The data to be sent in the body of the PUT request.
- ****kwargs** (*dict*) – Additional arguments for the HTTP PUT.

Keyword Arguments

headers (*dict*) – Header names and values to pass to the request.

Returns

The return data from the PUT request, as JSON.

Return type

object

select(*cls*, *unique_id=None*, **args*, ***kwargs*)

Prepare a query against the Carbon Black data store.

Most objects returned by the SDK are returned via queries created using this method.

Parameters

- **cls** (*class* / *str*) – The Model class (for example, Computer, Process, Binary, FileInstance) to query
- **unique_id** (*Any*) – The unique id of the object to retrieve, to retrieve a single object by ID. Default is None (create a standard query).
- ***args** (*list*) – Additional arguments to pass to a created object.

- ****kwargs** (*dict*) – Additional arguments to pass to a created object or query.

Returns

An instance of the `Model` class if a `unique_id` is provided, otherwise a `Query` object.

Return type

object

property url

The connection URL.

class CBCSDKSessionAdapter(*verify_hostname=True, force_tls_1_2=False, max_retries=0, **pool_kwargs*)

Bases: `HTTPAdapter`

Adapter object used to handle TLS connections to the CB server.

Initialize the `CBCSDKSessionManager`.

Parameters

- **verify_hostname** (*boolean*) – True if we want to verify the hostname.
- **force_tls_1_2** (*boolean*) – True to force the use of TLS 1.2.
- **max_retries** (*int*) – Maximum number of retries.
- ****pool_kwargs** – Additional arguments.

Raises

[*ApiError*](#) – If the library versions are too old to force the use of TLS 1.2.

add_headers(*request, **kwargs*)

Add any headers needed by the connection. As of v2.0 this does nothing by default, but is left for overriding by users that subclass the `HTTPAdapter`.

This should not be called from user code, and is only exposed for use when subclassing the `HTTPAdapter`.

Parameters

- **request** – The `PreparedRequest` to add headers to.
- **kwargs** – The keyword arguments from the call to `send()`.

build_response(*req, resp*)

Builds a `Response` object from a `urllib3` response. This should not be called from user code, and is only exposed for use when subclassing the `HTTPAdapter`

Parameters

- **req** – The `PreparedRequest` used to generate the response.
- **resp** – The `urllib3` response object.

Return type

`requests.Response`

cert_verify(*conn, url, verify, cert*)

Verify a SSL certificate. This method should not be called from user code, and is only exposed for use when subclassing the `HTTPAdapter`.

Parameters

- **conn** – The `urllib3` connection object associated with the cert.
- **url** – The requested URL.

- **verify** – Either a boolean, in which case it controls whether we verify the server’s TLS certificate, or a string, in which case it must be a path to a CA bundle to use
- **cert** – The SSL certificate to verify.

close()

Disposes of any internal state.

Currently, this closes the PoolManager and any active ProxyManager, which closes any pooled connections.

get_connection(url, proxies=None)

Returns a urllib3 connection for the given URL. This should not be called from user code, and is only exposed for use when subclassing the HTTPAdapter.

Parameters

- **url** – The URL to connect to.
- **proxies** – (optional) A Requests-style dictionary of proxies used on this request.

Return type

urllib3.ConnectionPool

init_poolmanager(connections, maxsize, block=False, **pool_kwargs)

Initialize the connection pool manager.

Parameters

- **connections** (*int*) – Initial number of connections to be used.
- **maxsize** (*int*) – Maximum size of the connection pool.
- **block** (*object*) – Blocking policy.
- ****pool_kwargs** – Additional arguments for the connection pool.

Returns

None

proxy_headers(proxy)

Returns a dictionary of the headers to add to any request sent through a proxy. This works with urllib3 magic to ensure that they are correctly sent to the proxy, rather than in a tunnelled request if CONNECT is being used.

This should not be called from user code, and is only exposed for use when subclassing the HTTPAdapter.

Parameters

proxy – The url of the proxy being used for this request.

Return type

dict

proxy_manager_for(proxy, **proxy_kwargs)

Return urllib3 ProxyManager for the given proxy.

This method should not be called from user code, and is only exposed for use when subclassing the HTTPAdapter.

Parameters

- **proxy** – The proxy to return a urllib3 ProxyManager for.
- **proxy_kwargs** – Extra keyword arguments used to configure the Proxy Manager.

Returns

ProxyManager

Return type

urllib3.ProxyManager

request_url(*request, proxies*)

Obtain the url to use when making the final request.

If the message is being sent through a HTTP proxy, the full URL has to be used. Otherwise, we should only use the path portion of the URL.

This should not be called from user code, and is only exposed for use when subclassing the HTTPAdapter.

Parameters

- **request** – The PreparedRequest being sent.
- **proxies** – A dictionary of schemes or schemes and hosts to proxy URLs.

Return type

str

send(*request, stream=False, timeout=None, verify=True, cert=None, proxies=None*)

Sends PreparedRequest object. Returns Response object.

Parameters

- **request** – The PreparedRequest being sent.
- **stream** – (optional) Whether to stream the request content.
- **timeout** (*float or tuple or urllib3 Timeout object*) – (optional) How long to wait for the server to send data before giving up, as a float, or a (connect timeout, read timeout) tuple.
- **verify** – (optional) Either a boolean, in which case it controls whether we verify the server's TLS certificate, or a string, in which case it must be a path to a CA bundle to use
- **cert** – (optional) Any user-provided SSL certificate to be trusted.
- **proxies** – (optional) The proxies dictionary to apply to the request.

Return type

requests.Response

class Connection(*credentials, integration_name=None, timeout=None, max_retries=None, proxy_session=None, **pool_kwargs*)

Bases: object

Object that encapsulates the HTTP connection to the CB server.

Initialize the Connection object.

Parameters

- **credentials** (*object*) – The credentials to use for the connection.
- **integration_name** (*str*) – The integration name being used.
- **timeout** (*int*) – The timeout value to use for HTTP requests on this connection.
- **max_retries** (*int*) – The maximum number of times to retry a request.
- **proxy_session** (*requests.Session*) –
- ****pool_kwargs** – Additional arguments to be used to initialize connection pooling.

Raises

- **`ApiError`** – If there's an internal error initializing the connection.
- **`ConnectionError`** – If there's a problem with the credentials.

`delete(url, **kwargs)`

Submit a DELETE request on this connection.

Parameters

- **`url`** (*str*) – The URL to submit the request to.
- **`**kwargs`** – Additional arguments for the request.

Returns

Result of the HTTP request.

Return type

object

`get(url, **kwargs)`

Submit a GET request on this connection.

Parameters

- **`url`** (*str*) – The URL to submit the request to.
- **`**kwargs`** – Additional arguments for the request.

Returns

Result of the HTTP request.

Return type

object

`http_request(method, url, **kwargs)`

Submit a HTTP request to the server.

Parameters

- **`method`** (*str*) – The method name to use for the HTTP request.
- **`url`** (*str*) – The URL to submit the request to.
- **`**kwargs`** – Additional arguments for the request.

Returns

Result of the HTTP request.

Return type

object

Raises

- **`ApiError`** – An unknown problem was detected.
- **`ClientError`** – The server returned an error code in the 4xx range, indicating a problem with the request.
- **`ConnectionError`** – A problem was seen with the HTTP connection.
- **`ObjectNotFoundError`** – The specified object was not found on the server.
- **`QuerySyntaxError`** – The query passed in had invalid syntax.

- **`ServerError`** – The server returned an error code in the 5xx range, indicating a problem on the server side.
- **`TimeoutError`** – The HTTP request timed out.
- **`UnauthorizedError`** – The stored credentials do not permit access to the specified request.

`post(url, **kwargs)`

Submit a POST request on this connection.

Parameters

- **`url`** (*str*) – The URL to submit the request to.
- **`**kwargs`** – Additional arguments for the request.

Returns

Result of the HTTP request.

Return type

object

`put(url, **kwargs)`

Submit a PUT request on this connection.

Parameters

- **`url`** (*str*) – The URL to submit the request to.
- **`**kwargs`** – Additional arguments for the request.

Returns

Result of the HTTP request.

Return type

object

`check_python_tls_compatibility()`

Verify which level of TLS/SSL that this version of the code is compatible with.

Returns

The maximum level of TLS/SSL that this version is compatible with.

Return type

str

`select_class_instance(cls: str)`

Given a string class name of a model class, returns the corresponding Carbon Black Cloud SDK class.

Parameters

`cls` (*str*) – The class name represented in a string.

Returns

The class specified by `cls`.

Return type

class

Raises

`ModelNotFound` – The specified class could not be found.

try_json(*resp*)

Return a parsed JSON representation of the input.

Parameters

resp (*Response*) – Input to be parsed.

Returns

The parsed JSON result, or an empty dict if the value is not valid JSON.

Return type

object

Credentials Module

Credentials management for the CBC SDK.

class CredentialProvider

Bases: object

The interface implemented by a credential provider.

get_credentials(*section=None*)

Return a Credentials object containing the configured credentials.

Parameters

section (*str*) – The credential section to retrieve.

Returns

The credentials retrieved from that source.

Return type

Credentials

Raises

CredentialError – If there is any error retrieving the credentials.

class CredentialValue(*value, names=None, *, module=None, qualname=None, type=None, start=1, boundary=None*)

Bases: Enum

All possible credential values.

requires_boolean_value()

Return whether or not this credential requires a boolean value.

Returns

True if the credential requires a Boolean value, False if not.

Return type

bool

requires_integer_value()

Return whether or not this credential requires an integer value.

Returns

True if the credential requires an integer value, False if not.

Return type

bool

class `Credentials`(*values=None*)

Bases: `object`

The object that contains credentials retrieved from the credential provider.

Initialize the `Credentials` object.

Parameters

values (*dict*) – Dictionary containing values to be set in the credentials.

Raises

`CredentialError` – If the value is not correct for any credential of boolean type.

get_token()

Get token required to authenticate with VMware Carbon Black Cloud

Returns

Token string for VMware Carbon Black Cloud

Return type

`str`

get_token_type()

Get token type `API_KEY` or `BEARER`

Returns

The token type

Return type

`str`

get_value(key)

Get the value of a credential.

Parameters

key (*CredentialValues*) – The credential to be retrieved.

Returns

The credential's value, or a default value if the value was not explicitly set.

Return type

`object`

to_dict()

Serializes the credentials into a dictionary.

Returns

Dictionary with the credentials.

Return type

`dict`

Errors Module

Exceptions that are thrown by CBC SDK operations.

exception `ApiError(message=None, original_exception=None)`

Bases: `Exception`

Base class for all CBC SDK errors; also raised for generic internal errors.

Initialize the `ApiError`.

Parameters

- **message** (*str*) – The actual error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

add_note()

`Exception.add_note(note)` – add a note to the exception

with_traceback()

`Exception.with_traceback(tb)` – set `self.__traceback__` to `tb` and return `self`.

exception `ClientError(error_code, message, **kwargs)`

Bases: [`ApiError`](#)

A `ClientError` is raised when an HTTP 4xx error code is returned from the Carbon Black server.

Initialize the `ClientError`.

Parameters

- **error_code** (*int*) – The error code that was received from the server.
- **message** (*str*) – The actual error message.
- **kwargs** (*dict*) – Additional arguments, which may include ‘result’ (server operation result), ‘original_exception’ (exception causing this one to be raised), and ‘uri’ (URI being accessed when this error was raised).

add_note()

`Exception.add_note(note)` – add a note to the exception

with_traceback()

`Exception.with_traceback(tb)` – set `self.__traceback__` to `tb` and return `self`.

exception `ConnectionError(message=None, original_exception=None)`

Bases: [`ApiError`](#)

There was an error in the connection to the server.

Initialize the `ApiError`.

Parameters

- **message** (*str*) – The actual error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

add_note()

`Exception.add_note(note)` – add a note to the exception

with_traceback()

`Exception.with_traceback(tb)` – set `self.__traceback__` to `tb` and return `self`.

exception CredentialError(*message=None, original_exception=None*)

Bases: [ApiError](#)

The credentials had an unspecified error.

Initialize the ApiError.

Parameters

- **message** (*str*) – The actual error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception FunctionalityDecommissioned(*functionality_tag, alternate=None*)

Bases: [ApiError](#)

Raised when a piece of decommissioned functionality is used.

Initialize the FunctionalityDecommissioned exception.

Parameters

- **functionality_tag** (*str*) – Should indicate which functionality has been decommissioned.
- **alternate** (*str*) – Optional indication of what the replacement for this functionality is.

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception InvalidHashError

Bases: [Exception](#)

An invalid hash value was used.

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception InvalidObjectError(*message=None, original_exception=None*)

Bases: [ApiError](#)

An invalid object was received by the server.

Initialize the ApiError.

Parameters

- **message** (*str*) – The actual error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception ModelNotFound

Bases: Exception

Exception for not finding a model while selecting dynamically.

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception MoreThanOneResultError(message=None, original_exception=None, results=None)

Bases: [ApiError](#)

Only one object was requested, but multiple matches were found in the Carbon Black datastore.

Initialize the MoreThanOneResultError.

Parameters

- **message** (*str*) – The actual error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.
- **results** (*list*) – List of results returned

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception NSXJobError(message=None, original_exception=None)

Bases: [ApiError](#)

NSX remediation jobs were not started

Initialize the ApiError.

Parameters

- **message** (*str*) – The actual error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception NonQueryableModel(message=None, original_exception=None)

Bases: [ApiError](#)

A model that attempted to be queried which is not queryable

Initialize the ApiError.

Parameters

- **message** (*str*) – The actual error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception ObjectNotFoundError(*uri, message=None, original_exception=None*)

Bases: [ApiError](#)

The requested object could not be found in the Carbon Black datastore.

Initialize the ObjectNotFoundError.

Parameters

- **uri** (*str*) – The URI of the action that failed.
- **message** (*str*) – The error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception OperationCancelled(*message=None, original_exception=None*)

Bases: [ApiError](#)

An operation in the background was canceled.

Initialize the ApiError.

Parameters

- **message** (*str*) – The actual error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception QuerySyntaxError(*uri, message=None, original_exception=None*)

Bases: [ApiError](#)

The request contains a query with malformed syntax.

Initialize the QuerySyntaxError.

Parameters

- **uri** (*str*) – The URI of the action that failed.
- **message** (*str*) – The error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception ServerError(error_code, message, **kwargs)

Bases: [ApiError](#)

A ServerError is raised when an HTTP 5xx error code is returned from the Carbon Black server.

Initialize the ServerError.

Parameters

- **error_code** (*int*) – The error code that was received from the server.
- **message** (*str*) – The actual error message.
- **kwargs** (*dict*) – Additional arguments, which may include ‘result’ (server operation result), ‘original_exception’ (exception causing this one to be raised), and ‘uri’ (URI being accessed when this error was raised).

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception TimeoutError(uri=None, error_code=None, message=None, original_exception=None)

Bases: [ApiError](#)

A requested operation timed out.

Initialize the TimeoutError.

Parameters

- **uri** (*str*) – The URI of the action that timed out.
- **error_code** (*int*) – The error code that was received from the server.
- **message** (*str*) – The error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception UnauthorizedError(uri, message=None, action='read', original_exception=None)

Bases: [ApiError](#)

The action that was attempted was not authorized.

Initialize the UnauthorizedError.

Parameters

- **uri** (*str*) – The URI of the action that was not authorized.
- **message** (*str*) – The error message.

- **action** (*str*) – The action that was being performed that was not authorized.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

Helpers Module

Helper functions which are not strictly part of the SDK API, but which are used by many of the examples.

build_cli_parser(*description='Cb Example Script'*)

Build a basic CLI parser containing the arguments needed to create a CBCloudAPI. Additional arguments may be added.

Parameters

description (*str*) – Description of the script, for use in help messages.

Returns

The new argument parser.

Return type

ArgumentParser

disable_insecure_warnings()

Disable warnings about insecure URLs.

eprint(*args, **kwargs)

Print to standard error output.

Parameters

- ***args** (*list*) – Arguments to the print function.
- ****kwargs** (*dict*) – Keyword arguments to the print function.

get_cb_cloud_object(*args*)

Based on parsed command line arguments, create and return a CBCloudAPI object.

Parameters

args (*Namespace*) – Arguments parsed from the command line.

Returns

The CBCloudAPI object.

Return type

CBCloudAPI

get_object_by_name_or_id(*cb, cls, name_field='name', id=None, name=None*)

Locate an object in the API by either ID or name.

Parameters

- **cb** (*CBCloudAPI*) – Reference to the CBCloudAPI.
- **cls** (*class*) – Class of object to be found.
- **name_field** (*str*) – Name field to search on.
- **id** (*int*) – ID of object to search for. May be None to do name searching.

- **name** (*str*) – Object name to search on.
- **force_init** (*bool*) – True to force a new object found by ID to be initialized.

Returns

List of objects that match the search criteria.

Return type

list

read_iocs(*cb*, *file*=<_io.TextIOWrapper name='<stdin>' mode='r' encoding='utf-8'>)

Read indicators of compromise from standard input.

Parameters

- **cb** ([CBCloudAPI](#)) – Reference to the CBCloudAPI.
- **file** – Not used.

Returns

New report ID to be used. dict: The indicators of compromise that were read in.

Return type

str

Live Response API Module

The Live Response API and associated objects.

class CbLRManagerBase(*cb*, *timeout*=30, *keepalive_sessions*=False, *thread_pool_count*=5)

Bases: object

Live Response manager object.

Initialize the CbLRManagerBase object.

Parameters

- **cb** ([BaseAPI](#)) – The CBC SDK object reference.
- **timeout** (*int*) – Timeout to use for requests, in seconds.
- **keepalive_sessions** (*bool*) – If True, “ping” sessions occasionally to ensure they stay alive.
- **thread_pool_count** (*int*) – number of workers for async commands (optional)

close_session(*device_id*, *session_id*)

Close the specified Live Response session.

Parameters

- **device_id** (*int*) – ID of the device.
- **session_id** (*int*) – ID of the session.

request_session(*device_id*, *async_mode*=False)

Initiate a new Live Response session.

Parameters

- **device_id** (*int*) – The device ID to use.

Returns

The new Live Response session.

Return type*CbLRSessionBase***stop_keepalive_thread()**

Stops the keepalive thread.

submit_job(*job*, *device*)

Submit a new job to be executed as a Live Response.

Parameters

- **job** (*func*) – The job function to be scheduled.
- **device** (*int*) – ID of the device to use for job execution.

Returns

A reference to the running job.

Return type

Future

class CbLRSessionBase(*cbl_manager*, *session_id*, *device_id*, *session_data*=None, *thread_pool_count*=5)

Bases: object

A Live Response session that interacts with a remote machine.

Initialize the CbLRSessionBase.

Parameters

- **cbl_manager** (*CbLRManagerBase*) – The Live Response manager governing this session.
- **session_id** (*str*) – The ID of this session.
- **device_id** (*int*) – The ID of the device (remote machine) we’re connected to.
- **session_data** (*dict*) – Additional session data.
- **thread_pool_count** (*int*) – number of workers for async commands (optional)

cancel_command(*command_id*)

Cancel command if it is in status PENDING.

Parameters

command_id (*int*) – command_id

close()

Close the Live Response session.

command_status(*command_id*)

Check the status of async command

Parameters

command_id (*int*) – command_id

Returns

status of the command

create_directory(*dir_name*, *async_mode*=False)

Create a directory on the remote machine.

Parameters

- **dir_name** (*str*) – The new directory name.

- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

create_process(*command_string*, *wait_for_output=True*, *remote_output_file_name=None*, *working_directory=None*, *wait_timeout=30*, *wait_for_completion=True*, *async_mode=False*)

Create a new process on the remote machine with the specified command string.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
...     print(lr_session.create_process(r'cmd.exe /c "ping.exe 192.168.1.1"'))
Ping 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time<1ms TTL=64
```

Parameters

- **command_string** (*str*) – Command string used for the create process operation.
- **wait_for_output** (*bool*) – True to block on output from the new process (execute in foreground). This will also set *wait_for_completion* (below).
- **remote_output_file_name** (*str*) – The remote output file name used for process output.
- **working_directory** (*str*) – The working directory of the create process operation.
- **wait_timeout** (*int*) – Timeout used for this command.
- **wait_for_completion** (*bool*) – True to wait until the process is completed before returning.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async str: The output of the process.

create_registry_key(*regkey*, *async_mode=False*)

Create a new registry key on the remote machine.

Parameters

- **regkey** (*str*) – The registry key to create.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

delete_file(*filename*, *async_mode=False*)

Delete the specified file name on the remote machine.

Parameters

- **filename** (*str*) – Name of the file to be deleted.

- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

delete_registry_key(*regkey, async_mode=False*)

Delete a registry key on the remote machine.

Parameters

- **regkey** (*str*) – The registry key to delete.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

delete_registry_value(*regkey, async_mode=False*)

Delete a registry value on the remote machine.

Parameters

- **regkey** (*str*) – The registry value to delete.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

get_file(*file_name, timeout=None, delay=None, async_mode=False*)

Retrieve contents of the specified file on the remote machine.

Parameters

- **file_name** (*str*) – Name of the file to be retrieved.
- **timeout** (*int*) – Timeout for the operation.
- **delay** (*float*) – Delay in seconds to wait before command complete.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async str: Contents of the specified file.

get_raw_file(*file_name, timeout=None, delay=None, async_mode=False*)

Retrieve contents of the specified file on the remote machine.

Parameters

- **file_name** (*str*) – Name of the file to be retrieved.
- **timeout** (*int*) – Timeout for the operation.
- **delay** (*float*) – Delay in seconds to wait before command complete.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async or object: Contains the data of the file.

get_registry_value(*regkey*, *async_mode=False*)

Return the associated value of the specified registry key on the remote machine.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
>>>     pprint.pprint(lr_session.
...     get_registry_value('HKLM\\SYSTEM\\CurrentControlSet\\services\\ACPI\\
↳Start'))
{'u'value_data': 0, u'value_name': u'Start', u'value_type': u'REG_DWORD'}
```

Parameters

- **regkey** (*str*) – The registry key to retrieve.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async or dict: A dictionary with keys of: value_data, value_name, value_type.

kill_process(*pid*, *async_mode=False*)

Terminate a process on the remote machine.

Parameters

- **pid** (*int*) – Process ID to be terminated.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async bool: True if success, False if failure.

list_directory(*dir_name*, *async_mode=False*)

List the contents of a directory on the remote machine.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
...     pprint.pprint(lr_session.list_directory('C:\\\\temp\\\\'))
[{'u'attributes': [u'DIRECTORY'],
  u'create_time': 1471897244,
  u'filename': u'.',
  u'last_access_time': 1476390670,
  u'last_write_time': 1476390670,
  u'size': 0},
```

```
{u'attributes': [u'DIRECTORY'],
  u'create_time': 1471897244, u'filename': u'.', u'last_access_time': 1476390670,
  u'last_write_time': 1476390670, u'size': 0},
```

```
{u'attributes': [u'ARCHIVE'],
 u'create_time': 1476390668, u'filename': u'test.txt', u'last_access_time': 1476390668,
 u'last_write_time': 1476390668, u'size': 0}}
```

Parameters

- **dir_name** (*str*) – Directory to list. This parameter should end with the path separator.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async or list: A list of dicts, each one describing a directory entry.

list_processes(*async_mode=False*)

List currently running processes on the remote machine.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
...     print(lr_session.list_processes()[0])
{u'command_line': u'',
 u'create_time': 1476260500,
 u'parent': 0,
 u'parent_guid': u'000000001-0000-0000-0000-000000000000',
 u'path': u'',
 u'pid': 4,
 u'proc_guid': u'000000001-0000-0004-01d2-2461a85e4546',
 u'sid': u's-1-5-18',
 u'username': u'NT AUTHORITY\\SYSTEM'}
```

Parameters

- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async or list: A list of dicts describing the processes.

list_registry_keys_and_values(*regkey, async_mode=False*)

Enumerate subkeys and values of the specified registry key on the remote machine.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
>>> pprint.pprint(lr_session.
...     list_registry_keys_and_values('HKLM\\SYSTEM\\CurrentControlSet\\
↪services\\ACPI'))
{'sub_keys': [u'Parameters', u'Enum'],
 'values': [{u'value_data': 0,
             u'value_name': u'Start',
             u'value_type': u'REG_DWORD'},
            {u'value_data': 1,
```

(continues on next page)

(continued from previous page)

```

    u'value_name': u'Type',
    u'value_type': u'REG_DWORD'},
    {u'value_data': 3,
     u'value_name': u'ErrorControl',
     u'value_type': u'REG_DWORD'},
    {u'value_data': u'system32\\drivers\\ACPI.sys',
     u'value_name': u'ImagePath',
     u'value_type': u'REG_EXPAND_SZ'},
    {u'value_data': u'Microsoft ACPI Driver',
     u'value_name': u'DisplayName',
     u'value_type': u'REG_SZ'},
    {u'value_data': u'Boot Bus Extender',
     u'value_name': u'Group',
     u'value_type': u'REG_SZ'},
    {u'value_data': u'acpi.inf_x86_neutral_ddd3c514822f1b21',
     u'value_name': u'DriverPackageId',
     u'value_type': u'REG_SZ'},
    {u'value_data': 1,
     u'value_name': u'Tag',
     u'value_type': u'REG_DWORD'}}}]

```

Parameters

- **regkey** (*str*) – The registry key to enumerate.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

or

dict: A dictionary with two keys, 'sub_keys' (a list of subkey names) and 'values' (a list of dicts containing value data, name, and type).

list_registry_values(*regkey, async_mode=False*)

Enumerate all registry values from the specified registry key on the remote machine.

Parameters

- **regkey** (*str*) – The registry key to enumerate.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async or list: List of values for the registry key.

memdump(*local_filename, remote_filename=None, compress=False, async_mode=False*)

Perform a memory dump operation on the remote machine.

Parameters

- **local_filename** (*str*) – Name of the file the memory dump will be transferred to on the local machine.

- **remote_filename** (*str*) – Name of the file the memory dump will be stored in on the remote machine.
- **compress** (*bool*) – True to compress the file on the remote system.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

put_file(*infp*, *remote_filename*, *async_mode=False*)

Create a new file on the remote machine with the specified data.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
...     lr_session.put_file(open("test.txt", "rb"), r"c:\test.txt")
```

Parameters

- **infp** (*object*) – Python file-like containing data to upload to the remote endpoint.
- **remote_filename** (*str*) – File name to create on the remote endpoint.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

set_registry_value(*regkey*, *value*, *overwrite=True*, *value_type=None*, *async_mode=False*)

Set a registry value on the specified registry key on the remote machine.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
...     lr_session.
...     set_registry_value('HKLM\\SYSTEM\\CurrentControlSet\\services\\
↪ACPI\\testvalue', 1)
```

Parameters

- **regkey** (*str*) – The registry key to set.
- **value** (*object*) – The value data.
- **overwrite** (*bool*) – If True, any existing value will be overwritten.
- **value_type** (*str*) – The type of value. Examples: REG_DWORD, REG_MULTI_SZ, REG_SZ
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

start_memdump(*remote_filename=None, compress=True*)

Start a memory dump operation on the remote machine.

Parameters

- **remote_filename** (*str*) – Name of the file the memory dump will be stored in on the remote machine.
- **compress** (*bool*) – True to compress the file on the remote system.

Returns

Controlling object for the memory dump operation.

Return type

LiveResponseMemdump

walk(*top, topdown=True, onerror=None, followlinks=False*)

Perform a full directory walk with recursion into subdirectories on the remote machine.

Note: walk does not support `async_mode` due to its behaviour, it can only be invoked synchronously

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
...     for entry in lr_session.walk(directory_name):
...         print(entry)
('C:\\temp\\', [u'dir1', u'dir2'], [u'file1.txt'])
```

Parameters

- **top** (*str*) – Directory to recurse on.
- **topdown** (*bool*) – If True, start output from top level directory.
- **onerror** (*func*) – Callback if an error occurs. This function is called with one argument (the exception that occurred).
- **followlinks** (*bool*) – True to follow symbolic links.

Returns

List of tuples containing directory name, subdirectory names, file names.

Return type

list

class CompletionNotification(*device_id*)

Bases: object

The notification that an operation is complete.

Initialize the CompletionNotification.

Parameters

device_id (*int*) – The device ID this notification is for.

class GetFileJob(*file_name*)

Bases: object

Object that retrieves a file via Live Response.

Initialize the GetFileJob.

Parameters

file_name (*str*) – The name of the file to be fetched.

run(*session*)

Execute the file transfer.

Parameters

session (*CbLRSessionBase*) – The Live Response session being used.

Returns

The contents of the file being retrieved.

Return type

str

class JobWorker(*cb, device_id, result_queue*)

Bases: *Thread*

Thread object that executes individual Live Response jobs.

Initialize the JobWorker.

Parameters

- **cb** (*BaseAPI*) – The CBC SDK object reference.
- **device_id** (*int*) – The ID of the device being used.
- **result_queue** (*Queue*) – The queue where results are placed.

property daemon

A boolean value indicating whether this thread is a daemon thread.

This must be set before `start()` is called, otherwise `RuntimeError` is raised. Its initial value is inherited from the creating thread; the main thread is not a daemon thread and therefore all threads created in the main thread default to `daemon = False`.

The entire Python program exits when only daemon threads are left.

getName()

Return a string used for identification purposes only.

This method is deprecated, use the `name` attribute instead.

property ident

Thread identifier of this thread or `None` if it has not been started.

This is a nonzero integer. See the `get_ident()` function. Thread identifiers may be recycled when a thread exits and another thread is created. The identifier is available even after the thread has exited.

isDaemon()

Return whether this thread is a daemon.

This method is deprecated, use the `daemon` attribute instead.

is_alive()

Return whether the thread is alive.

This method returns `True` just before the `run()` method starts until just after the `run()` method terminates. See also the module function `enumerate()`.

join(*timeout=None*)

Wait until the thread terminates.

This blocks the calling thread until the thread whose `join()` method is called terminates – either normally or through an unhandled exception or until the optional timeout occurs.

When the timeout argument is present and not `None`, it should be a floating point number specifying a timeout for the operation in seconds (or fractions thereof). As `join()` always returns `None`, you must call `is_alive()` after `join()` to decide whether a timeout happened – if the thread is still alive, the `join()` call timed out.

When the timeout argument is not present or `None`, the operation will block until the thread terminates.

A thread can be `join()`ed many times.

`join()` raises a `RuntimeError` if an attempt is made to join the current thread as that would cause a deadlock. It is also an error to `join()` a thread before it has been started and attempts to do so raises the same exception.

property name

A string used for identification purposes only.

It has no semantics. Multiple threads may be given the same name. The initial name is set by the constructor.

property native_id

Native integral thread ID of this thread, or `None` if it has not been started.

This is a non-negative integer. See the `get_native_id()` function. This represents the Thread ID as reported by the kernel.

run()

Execute the job worker.

run_job(*work_item*)

Execute an individual `WorkItem`.

Parameters

work_item (`WorkItem`) – The work item to execute.

setDaemon(*daemonic*)

Set whether this thread is a daemon.

This method is deprecated, use the `.daemon` property instead.

setName(*name*)

Set the name string for this thread.

This method is deprecated, use the `name` attribute instead.

start()

Start the thread's activity.

It must be called at most once per thread object. It arranges for the object's `run()` method to be invoked in a separate thread of control.

This method will raise a `RuntimeError` if called more than once on the same thread object.

exception LiveResponseError(*details*)

Bases: `Exception`

Exception raised for errors with Live Response.

Initialize the `LiveResponseError`.

Parameters

details (*object*) – Details of the specific error.

add_note()

Exception.add_note(note) – add a note to the exception

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

class LiveResponseJobScheduler(*cb, max_workers=10*)

Bases: Thread

Thread that schedules Live Response jobs.

Initialize the LiveResponseJobScheduler.

Parameters

- **cb** ([BaseAPI](#)) – The CBC SDK object reference.
- **max_workers** (*int*) – Maximum number of JobWorker threads to use.

getName()

Return a string used for identification purposes only.

This method is deprecated, use the name attribute instead.

property ident

Thread identifier of this thread or None if it has not been started.

This is a nonzero integer. See the get_ident() function. Thread identifiers may be recycled when a thread exits and another thread is created. The identifier is available even after the thread has exited.

isDaemon()

Return whether this thread is a daemon.

This method is deprecated, use the daemon attribute instead.

is_alive()

Return whether the thread is alive.

This method returns True just before the run() method starts until just after the run() method terminates. See also the module function enumerate().

join(*timeout=None*)

Wait until the thread terminates.

This blocks the calling thread until the thread whose join() method is called terminates – either normally or through an unhandled exception or until the optional timeout occurs.

When the timeout argument is present and not None, it should be a floating point number specifying a timeout for the operation in seconds (or fractions thereof). As join() always returns None, you must call is_alive() after join() to decide whether a timeout happened – if the thread is still alive, the join() call timed out.

When the timeout argument is not present or None, the operation will block until the thread terminates.

A thread can be join()ed many times.

join() raises a RuntimeError if an attempt is made to join the current thread as that would cause a deadlock. It is also an error to join() a thread before it has been started and attempts to do so raises the same exception.

property name

A string used for identification purposes only.

It has no semantics. Multiple threads may be given the same name. The initial name is set by the constructor.

property native_id

Native integral thread ID of this thread, or None if it has not been started.

This is a non-negative integer. See the `get_native_id()` function. This represents the Thread ID as reported by the kernel.

run()

Execute the job scheduler.

setDaemon(*daemonic*)

Set whether this thread is a daemon.

This method is deprecated, use the `.daemon` property instead.

setName(*name*)

Set the name string for this thread.

This method is deprecated, use the `name` attribute instead.

start()

Start the thread's activity.

It must be called at most once per thread object. It arranges for the object's `run()` method to be invoked in a separate thread of control.

This method will raise a `RuntimeError` if called more than once on the same thread object.

submit_job(*work_item*)

Submit a new job to be processed.

Parameters

work_item ([WorkItem](#)) – New job to be processed.

class LiveResponseMemdump(*lr_session*, *memdump_id*, *remote_filename*)

Bases: `object`

Object managing a memory dump on a remote machine.

Initialize the `LiveResponseMemdump`.

Parameters

- **lr_session** ([Session](#)) – The Live Response session to the machine doing the memory dump.
- **memdump_id** (*str*) – The ID of the memory dump being performed.
- **remote_filename** (*str*) – The file name the memory dump will be stored in on the remote machine.

delete()

Delete the memory dump file.

get(*local_filename*)

Retrieve the remote memory dump to a local file.

Parameters

local_filename (*str*) – Filename locally that will receive the memory dump.

wait()

Wait for the remote memory dump to complete.

class LiveResponseSession(*cbl_r_manager, session_id, device_id, session_data=None*)

Bases: [*CbLRSessionBase*](#)

Public face of the Live Response session object.

Initializes the LiveResponseSession.

Parameters

- **cbl_r_manager** ([*LiveResponseSessionManager*](#)) – Reference to the session manager.
- **session_id** (*str*) – The ID of this session.
- **device_id** (*int*) – The ID of the device (remote machine) we’re connected to.
- **session_data** (*dict*) – Additional session data.

cancel_command(*command_id*)

Cancel command if it is in status PENDING.

Parameters

command_id (*int*) – command_id

close()

Close the Live Response session.

command_status(*command_id*)

Check the status of async command

Parameters

command_id (*int*) – command_id

Returns

status of the command

create_directory(*dir_name, async_mode=False*)

Create a directory on the remote machine.

Parameters

- **dir_name** (*str*) – The new directory name.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

create_process(*command_string, wait_for_output=True, remote_output_file_name=None, working_directory=None, wait_timeout=30, wait_for_completion=True, async_mode=False*)

Create a new process on the remote machine with the specified command string.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
...     print(lr_session.create_process(r'cmd.exe /c "ping.exe 192.168.1.1"'))
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time<1ms TTL=64
```

Parameters

- **command_string** (*str*) – Command string used for the create process operation.
- **wait_for_output** (*bool*) – True to block on output from the new process (execute in foreground). This will also set wait_for_completion (below).
- **remote_output_file_name** (*str*) – The remote output file name used for process output.
- **working_directory** (*str*) – The working directory of the create process operation.
- **wait_timeout** (*int*) – Timeout used for this command.
- **wait_for_completion** (*bool*) – True to wait until the process is completed before returning.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async str: The output of the process.

create_registry_key(*regkey, async_mode=False*)

Create a new registry key on the remote machine.

Parameters

- **regkey** (*str*) – The registry key to create.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

delete_file(*filename, async_mode=False*)

Delete the specified file name on the remote machine.

Parameters

- **filename** (*str*) – Name of the file to be deleted.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

delete_registry_key(*regkey, async_mode=False*)

Delete a registry key on the remote machine.

Parameters

- **regkey** (*str*) – The registry key to delete.

- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

delete_registry_value(*regkey, async_mode=False*)

Delete a registry value on the remote machine.

Parameters

- **regkey** (*str*) – The registry value to delete.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

get_file(*file_name, timeout=None, delay=None, async_mode=False*)

Retrieve contents of the specified file on the remote machine.

Parameters

- **file_name** (*str*) – Name of the file to be retrieved.
- **timeout** (*int*) – Timeout for the operation.
- **delay** (*float*) – Delay in seconds to wait before command complete.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async str: Contents of the specified file.

get_raw_file(*file_name, timeout=None, delay=None, async_mode=False*)

Retrieve contents of the specified file on the remote machine.

Parameters

- **file_name** (*str*) – Name of the file to be retrieved.
- **timeout** (*int*) – Timeout for the operation.
- **delay** (*float*) – Delay in seconds to wait before command complete.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async or object: Contains the data of the file.

get_registry_value(*regkey, async_mode=False*)

Return the associated value of the specified registry key on the remote machine.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
>>>     pprint.pprint(lr_session.
...     get_registry_value('HKLM\\SYSTEM\\CurrentControlSet\\services\\ACPI\\
↪Start'))
{'u'value_data': 0, u'value_name': u'Start', u'value_type': u'REG_DWORD'}
```

Parameters

- **regkey** (*str*) – The registry key to retrieve.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async or dict: A dictionary with keys of: value_data, value_name, value_type.

kill_process(*pid, async_mode=False*)

Terminate a process on the remote machine.

Parameters

- **pid** (*int*) – Process ID to be terminated.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async bool: True if success, False if failure.

list_directory(*dir_name, async_mode=False*)

List the contents of a directory on the remote machine.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
...     pprint.pprint(lr_session.list_directory('C:\\\\temp\\\\'))
[{u'attributes': [u'DIRECTORY'],
  u'create_time': 1471897244,
  u'filename': u'..',
  u'last_access_time': 1476390670,
  u'last_write_time': 1476390670,
  u'size': 0},
```

```
  {u'attributes': [u'DIRECTORY'],
    u'create_time': 1471897244, u'filename': u'..', u'last_access_time': 1476390670,
    u'last_write_time': 1476390670, u'size': 0},
```

```
  {u'attributes': [u'ARCHIVE'],
    u'create_time': 1476390668, u'filename': u'test.txt', u'last_access_time': 1476390668,
    u'last_write_time': 1476390668, u'size': 0}]
```

Parameters

- **dir_name** (*str*) – Directory to list. This parameter should end with the path separator.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async or list: A list of dicts, each one describing a directory entry.

list_processes(*async_mode=False*)

List currently running processes on the remote machine.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
...     print(lr_session.list_processes()[0])
{'command_line': u'',
 'create_time': 1476260500,
 'parent': 0,
 'parent_guid': u'000000001-0000-0000-0000-000000000000',
 'path': u'',
 'pid': 4,
 'proc_guid': u'000000001-0000-0004-01d2-2461a85e4546',
 'sid': u's-1-5-18',
 'username': u'NT AUTHORITY\\SYSTEM'}
```

Parameters

async_mode (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async or list: A list of dicts describing the processes.

list_registry_keys_and_values(*regkey, async_mode=False*)

Enumerate subkeys and values of the specified registry key on the remote machine.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
>>> pprint.pprint(lr_session.
...     list_registry_keys_and_values('HKLM\\SYSTEM\\CurrentControlSet\\
↳ services\\ACPI'))
{'sub_keys': [u'Parameters', u'Enum'],
 'values': [{u'value_data': 0,
             u'value_name': u'Start',
             u'value_type': u'REG_DWORD'},
            {u'value_data': 1,
             u'value_name': u'Type',
             u'value_type': u'REG_DWORD'},
            {u'value_data': 3,
             u'value_name': u'ErrorControl',
             u'value_type': u'REG_DWORD'},
            {u'value_data': u'system32\\drivers\\ACPI.sys',
```

(continues on next page)

(continued from previous page)

```

    u'value_name': u'ImagePath',
    u'value_type': u'REG_EXPAND_SZ'},
    {u'value_data': u'Microsoft ACPI Driver',
     u'value_name': u'DisplayName',
     u'value_type': u'REG_SZ'},
    {u'value_data': u'Boot Bus Extender',
     u'value_name': u'Group',
     u'value_type': u'REG_SZ'},
    {u'value_data': u'acpi.inf_x86_neutral_ddd3c514822f1b21',
     u'value_name': u'DriverPackageId',
     u'value_type': u'REG_SZ'},
    {u'value_data': 1,
     u'value_name': u'Tag',
     u'value_type': u'REG_DWORD'}}}]

```

Parameters

- **regkey** (*str*) – The registry key to enumerate.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

or

dict: A dictionary with two keys, 'sub_keys' (a list of subkey names) and 'values' (a list of dicts containing value data, name, and type).

list_registry_values(*regkey*, *async_mode=False*)

Enumerate all registry values from the specified registry key on the remote machine.

Parameters

- **regkey** (*str*) – The registry key to enumerate.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async or list: List of values for the registry key.

memdump(*local_filename*, *remote_filename=None*, *compress=False*, *async_mode=False*)

Perform a memory dump operation on the remote machine.

Parameters

- **local_filename** (*str*) – Name of the file the memory dump will be transferred to on the local machine.
- **remote_filename** (*str*) – Name of the file the memory dump will be stored in on the remote machine.
- **compress** (*bool*) – True to compress the file on the remote system.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

put_file(*infp*, *remote_filename*, *async_mode=False*)

Create a new file on the remote machine with the specified data.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
...     lr_session.put_file(open("test.txt", "rb"), r"c:\test.txt")
```

Parameters

- **infp** (*object*) – Python file-like containing data to upload to the remote endpoint.
- **remote_filename** (*str*) – File name to create on the remote endpoint.
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

set_registry_value(*regkey*, *value*, *overwrite=True*, *value_type=None*, *async_mode=False*)

Set a registry value on the specified registry key on the remote machine.

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
...     lr_session.
...     set_registry_value('HKLM\\SYSTEM\\CurrentControlSet\\services\\
...     ↪ACPI\\testvalue', 1)
```

Parameters

- **regkey** (*str*) – The registry key to set.
- **value** (*object*) – The value data.
- **overwrite** (*bool*) – If True, any existing value will be overwritten.
- **value_type** (*str*) – The type of value. Examples: REG_DWORD, REG_MULTI_SZ, REG_SZ
- **async_mode** (*bool*) – Flag showing whether the command should be executed asynchronously

Returns

command_id, future if ran async

start_memdump(*remote_filename=None*, *compress=True*)

Start a memory dump operation on the remote machine.

Parameters

- **remote_filename** (*str*) – Name of the file the memory dump will be stored in on the remote machine.

- **compress** (*bool*) – True to compress the file on the remote system.

Returns

Controlling object for the memory dump operation.

Return type

LiveResponseMemdump

walk(*top*, *topdown=True*, *onerror=None*, *followlinks=False*)

Perform a full directory walk with recursion into subdirectories on the remote machine.

Note: walk does not support *async_mode* due to its behaviour, it can only be invoked synchronously

Example

```
>>> with c.select(Device, 1).lr_session() as lr_session:
...     for entry in lr_session.walk(directory_name):
...         print(entry)
('C:\\temp\\', [u'dir1', u'dir2'], [u'file1.txt'])
```

Parameters

- **top** (*str*) – Directory to recurse on.
- **topdown** (*bool*) – If True, start output from top level directory.
- **onerror** (*func*) – Callback if an error occurs. This function is called with one argument (the exception that occurred).
- **followlinks** (*bool*) – True to follow symbolic links.

Returns

List of tuples containing directory name, subdirectory names, file names.

Return type

list

class LiveResponseSessionManager(*cb*, *timeout=30*, *keepalive_sessions=False*)

Bases: *CbLRManagerBase*

Session manager for Live Response sessions.

Initialize the LiveResponseSessionManager - only needed to format *cbldr_base*

cbldr_session_cls

alias of *LiveResponseSession*

close_session(*device_id*, *session_id*)

Close the specified Live Response session.

Parameters

- **device_id** (*int*) – ID of the device.
- **session_id** (*int*) – ID of the session.

request_session(*device_id*, *async_mode=False*)

Initiate a new Live Response session.

Parameters

- **device_id** (*int*) – The device ID to use.

Returns

The new Live Response session.

Return type

CbLRSessionBase

session_status(*session_id*)

Check the status of a lr session

Parameters

session_id (*str*) – The id of the session.

Returns

Status of the session

Return type

str

stop_keepalive_thread()

Stops the keepalive thread.

submit_job(*job, device*)

Submit a job for execution by the job scheduler.

Parameters

- **job** (*func*) – The job function to be executed.
- **device** (*object*) – The device ID or Device object the job will be executed on.

Returns

A Future that will allow waiting until the job is complete.

Return type

Future

class WorkItem(*fn, device_id*)

Bases: object

Work item for scheduling.

Initialize the WorkItem.

Parameters

- **fn** (*func*) – The function to be called to do the actual work.
- **device_id** (*object*) – The device ID or Device object the work item is directed for.

class WorkerStatus(*device_id, status='READY', exception=None*)

Bases: object

Holds the status of an individual worker.

Initialize the WorkerStatus.

Parameters

- **device_id** (*int*) – The device ID this status is for.
- **status** (*str*) – The current status value.
- **exception** (*Exception*) – Any exception that happened.

jobrunner(*callable*, *cb*, *device_id*)

Wrap a callable object with a live response session.

Parameters

- **callable** (*object*) – The object to be wrapped.
- **cb** ([BaseAPI](#)) – The CBC SDK object reference.
- **device_id** (*int*) – The device ID to use to get the session.

Returns

The wrapped object.

Return type

object

poll_status(*cb*, *url*, *desired_status*='COMPLETE', *timeout*=None, *delay*=None)

Poll the status of a Live Response query.

Parameters

- **cb** ([BaseAPI](#)) – The CBC SDK object reference.
- **url** (*str*) – The URL to poll.
- **desired_status** (*str*) – The status we're looking for.
- **timeout** (*int*) – The timeout value in seconds.
- **delay** (*float*) – The delay between attempts in seconds.

Returns

The result of the Live Response query that has the desired status.

Return type

object

Raises

[LiveResponseError](#) – If an error response was encountered.

Utils Module

Utility functions for use within the CBC SDK.

convert_from_cb(*s*)

Parse a date and time value into a datetime object.

Parameters

s (*str*) – The date and time string to parse. If this is None, we use the UNIX epoch timestamp.

Returns

The parsed date and time.

Return type

datetime

convert_to_cb(*dt*)

Convert a date and time to a string in the Carbon Black format.

Parameters

dt (*datetime*) – The date and time to be converted.

Returns

The date and time as a string.

Return type

str

WinError Module

Error related constants for win32

Generated by h2py from winerror.h

class CommDlgError

Bases: [ErrorBaseClass](#)

Collects all the common dialog error codes.

classmethod lookup_error(*error_code*)

Look up an error code by value.

Parameters

error_code (*int*) – The error code to be looked up.

Returns

The error code name.

Return type

str

class DirectoryStorageError

Bases: [ErrorBaseClass](#)

Collects all the directory storage error codes.

classmethod lookup_error(*error_code*)

Look up an error code by value.

Parameters

error_code (*int*) – The error code to be looked up.

Returns

The error code name.

Return type

str

class ErrorBaseClass

Bases: object

Base class for repositories of error codes.

classmethod lookup_error(*error_code*)

Look up an error code by value.

Parameters

error_code (*int*) – The error code to be looked up.

Returns

The error code name.

Return type

str

class ErrorMetaClass(*name, bases, clsdict*)

Bases: `type`

Metaclass which establishes an easy means of looking up error codes in a collection.

Creates a new instance of a class, setting up the dict to make it easy to look up error codes.

Parameters

- **name** (*str*) – The name of the class.
- **bases** (*list*) – Base classes of the class to be created.
- **clsdict** (*dict*) – Elements defined in the new class.

mro()

Return a type's method resolution order.

FAILED(*Status*)

Return True iff a HRESULT/SCODE status represents failure.

class Facility

Bases: [*ErrorBaseClass*](#)

Collects all known facility codes.

classmethod lookup_error(*error_code*)

Look up an error code by value.

Parameters

error_code (*int*) – The error code to be looked up.

Returns

The error code name.

Return type

`str`

GetScode(*hr*)

Turn a HRESULT into a SCODE.

HRESULT_CODE(*hr*)

Return the error code field of a HRESULT.

HRESULT_FACILITY(*hr*)

Return the facility field of a HRESULT.

HRESULT_FROM_NT(*x*)

Turn an NT error code into a HRESULT.

HRESULT_FROM_WIN32(*scode*)

Return the HRESULT corresponding to a Win32 error code.

HRESULT_SEVERITY(*hr*)

Return the severity field of a HRESULT.

class RawErrorCode

Bases: [*ErrorBaseClass*](#)

Collects all known error codes defined as raw SCODEs (from COM, OLE, etc.)

classmethod `lookup_error(error_code)`

Look up an error code by value.

Parameters

error_code (*int*) – The error code to be looked up.

Returns

The error code name.

Return type

str

ResultFromScore(*sc*)

Turn a SCORE into a HRESULT.

SCORE_CODE(*sc*)

Return the error code field of a SCORE.

SCORE_FACILITY(*sc*)

Return the facility field of a SCORE.

SCORE_SEVERITY(*sc*)

Return the severity field of a SCORE.

SUCCEEDED(*Status*)

Return True iff a HRESULT/SCORE status represents success.

class `Win32Error`

Bases: [*ErrorBaseClass*](#)

Collects all the Win32 error codes.

classmethod `lookup_error(error_code)`

Look up an error code by value.

Parameters

error_code (*int*) – The error code to be looked up.

Returns

The error code name.

Return type

str

decode_hresult(*hresult*)

Look up a Win32 error code based on the error code in a HRESULT.

4.14 Logging & Diagnostics

The `cbc_sdk` provides extensive logging facilities to track down issues communicating with the REST API and understand potential performance bottlenecks.

4.14.1 Enabling Logging

The `cbc_sdk` uses Python's standard logging module for logging. To enable debug logging for the `cbc_sdk`, you can do the following:

```
>>> import logging
>>> logging.basicConfig(level=logging.DEBUG)
```

All REST API calls, including the API endpoint, any data sent via POST or PUT, and the time it took for the call to complete:

```
>>> devices = [ device for device in cb.select(Device) ]
DEBUG:cbc_sdk.connection:Sending HTTP POST /appservices/v6/orgs/ABCD1234/devices/_search_
↳with {"criteria": {}, "exclusions": {}, "query": ""}
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): defense-eap01.
↳conferdeploy.net:443
DEBUG:urllib3.connectionpool:https://defense-eap01.conferdeploy.net:443 "POST /
↳appservices/v6/orgs/ABCD1234/devices/_search HTTP/1.1" 200 None
DEBUG:cbc_sdk.connection:HTTP POST /appservices/v6/orgs/ABCD1234/devices/_search took 0.
↳409s (response 200)
```

4.15 Testing

This document will provide information about how to run the functional tests for the CBC Python SDK in Linux and Windows platforms.

These instructions assume you already have the CBC SDK sources present locally. If not, they can be checked out from GitHub using the URL <https://github.com/carbonblack/carbon-black-cloud-sdk-python>; doing so will require you to either have Git installed or download the source tree packed as a zip archive from GitHub and then unarchive it.

4.15.1 Running the tests on Microsoft Windows

Install Python

From <http://python.org>, download the installer for the most recent Python 3.8 version (as of this writing, version 3.8.6 is the latest).

Fix the Execution PATH

Go to the Environment Variables dialog (System Control Panel or Properties page for My Computer/This PC, then select **Advanced system settings** and then the **Environment Variables** button). Ensure that the first two components of the user PATH environment variable are `%USERPROFILE%\AppData\Local\Programs\Python\Python38` and `%USERPROFILE%\AppData\Local\Programs\Python\Python38\Scripts`.

To test this, open a command window and use the command: `python --version`

It should run Python and show that you are running Python 3.8.

Install CBC Python SDK Requirements

From the top-level CBC SDK source directory, execute the following commands:

```
pip install -r requirements.txt
```

This will ensure that all required python modules are installed.

Execute the Functional Tests

From the top-level CBC SDK source directory, execute the following command:

```
pytest
```

The tests should return that they all completed successfully.

4.15.2 Running the tests on Linux

Carbon Black Cloud Python SDK provides a number of Dockerfiles inside the docker folder of the source root. Those contain the necessary instructions to build docker images containing a number of distributions with CBC Python SDK preinstalled in /app directory (relative to image root).

Build the docker image

Currently the following Dockerfiles are available:

- docker/amazon/Dockerfile - Amazon Linux (latest) image
- docker/ubuntu/Dockerfile - Ubuntu 18.04 image
- docker/rhel/Dockerfile - RHEL8 UBI image
- docker/suse/Dockerfile - OpenSUSE Leap (latest) image

Building the images should be done from the CBC SDK root directory by explicitly providing the path to the Dockerfile to be built, e.g for the RHEL one, the build command would be:

```
docker build -t cbc-sdk-python=rhel -f docker/rhel/Dockerfile .
```

By default, the docker Unix socket is owned by root user / docker group. In case you are running the build as a non-root user that isn't member of docker group, sudo should be used:

```
sudo docker build -t cbc-sdk-python-rhel -f docker/rhel/Dockerfile .
```

Run the container and execute the test

When the docker image builds, it should be started, e.g:

```
docker run -it cbc-sdk-python-rhel
```

This will run the container and spawn an interactive shell running in it. CBC Python SDK is installed in the /app directory, so pytest needs to be executed from there:

```
cd /app && pytest
```

4.16 Changelog

4.16.1 CBC SDK 1.5.1 - Released January 30, 2024

New Features:

- Asset Groups - Added management of asset groups:
 - Create, delete, and update asset groups (either with manual or dynamic membership)
 - Retrieve asset groups by ID
 - Search for asset groups, retrieve list of all asset groups
 - Add/remove members, get all members in a group
 - Get statistics for a group
 - Helper functions for `Device` to retrieve and maintain group membership
 - Preview changes to effective policy for device(s) as a result of a number of different potential changes
 - Full documentation and new Guide page
- Alerts v7 Enhancements - Added additional functionality to Alerts v7 as implemented in version 1.5.0:
 - Search Grouped Alerts, including faceting and retrieval of all alerts for a group
 - Get list of watchlists on an alert
 - Network threat metadata helper function
 - Full update to Alerts guide in documentation
- Command line deobfuscation added to Processes, Alerts, and Observations, allowing visualization of PowerShell command lines that have been deliberately obfuscated by attackers.
- New `scroll()` method added to Live Query search results.
- New helper methods added to `Policy` to enable or disable XDR data collection and auth event data collection.
- New `export()` and `scroll()` methods added to `DeviceSearchQuery`.

Updates:

- Python 3.7 has been re-added as “unofficially” supported, since certain integrations that use the SDK still use it.
- Added `deployment_type` as part of the facets available in `DeviceSearchQuery`.

Bug Fixes:

- Search jobs that allow setting a timeout now default that timeout to 5 minutes. The timeout may be lowered from that point, but *never* raised beyond it. This eliminates a problem of “hung” searches.

Documentation:

- ReadTheDocs generation has been improved to show the inherited methods. There are some helper functions on `SearchQuery` classes such as `add_criteria()` inherited from `CriteriaBuilderSupportMixin` and `first()` inherited from `IterableQueryMixin`.

4.16.2 CBC SDK 1.5.0 - Released October 24, 2023

Alerts Update to use V7 API

The new Alerts V7 API will improve alert management and allow for easier management, consumption, and triage of alerts in the Carbon Black Cloud. Alerts v7 API extends the capabilities with improved methods of retrieving alerts and added functionality to manage alert workflow.

N.B.: This change involves breaking changes to the SDK involving the core Alerts workflow. Please check your existing code carefully before deploying this SDK upgrade.

Breaking Changes:

- Alerts V7: Certain changes are not compatible with code written to the old V6 API. For details, please see the [Alert Migration Guide](#). Breaking changes include:
 - Default Search Time Period is reduced to two weeks.
 - For fields that do not exist in the Alerts V7 API, a `FunctionalityDecommissioned` exception is raised.
 - `get_events()` method has been removed.
 - All facet terms match the field names.
 - Workflow has been rebuilt.
 - Create Note returns a single `Note` instance instead of a list.
- Official support for Python 3.7 has been dropped, since that version is now end-of-life. Added explicit testing support for Python version 3.12. **N.B.:** End users should update their Python version to 3.8.x or greater.

New Features:

- Alerts V7:
 - Extended alert schema with additional metadata such as process command line and username, parent and child process information, netconn data, additional device fields, MITRE categorization when available, and more
 - Ability to mark alerts as “In Progress”
 - Ability to mark alerts as True Positive or False Positive
 - Additional fields available for both searching and faceting
 - Enhanced note management with the ability to add notes to both individual alerts and threats (alerts grouped by threat)
 - Observed Alerts have been removed from the Alerts API as these events are not considered actionable threats. They can now be retrieved via the Observations API.
- External Devices: Added External Device Export and External Device Approvals Export.

Updates:

- Audit log requests have moved from `CBCloudAPI` into their own function entry point in the `platform` package. The old function has been deprecated.
- Process search validation has been changed to use the V2 POST API rather than the old V1 GET API.
- `CBCloudAPI.get_notifications()` and `CBCloudAPI.notification_listener()` have been marked as deprecated.

Documentation:

- Added example script to poll for audit logs.

- CBCloudAPI documentation has been pulled out into its own page.
- Authentication, Getting Started, and Guides pages have been updated.
- Concepts page has been removed, and the information it contained has moved to other pages.
- New *Searching guide* added.
- Update to left-hand sidebar to allow the Guides sub-listing to be collapsed.
- Porting guide has been updated to reflect the latest APIs.
- Live Response migration guide has been updated with links.
- README.md has been updated with better instructions for generating docs locally.
- CBCloudAPI and Devices documentation have been updated to better conform to new style guide for docstrings.

4.16.3 CBC SDK 1.4.3 - Released June 26, 2023

New Features:

- Policy Rule Configurations - support for additional rule configuration types:
 - Host-Based Firewall - addresses the protection of assets based on rules governing network and application behavior.
 - Data Collection - control over what data is uploaded to the Carbon Black Cloud. Specifically, can enable or disable auth events collection.

Updates:

- Added an example script for manipulating core prevention rule configuration and data collection status on a policy.
- Changed pymox dependency to the latest version, which eliminates warning messages on unit test and provides compatibility with Python 3.11 and later.
- Added specific testing support for Python 3.11.
- Added additional UAT tests for authentication events.
- Many exception classes now carry a `uri` field which holds the URI of the API being accessed that caused the exception to be raised.

Bug Fixes:

- Fixed link validation for reports and IOCs to accept IPv4 addresses, domain names, or URIs.

Documentation:

- Documentation has been reorganized for ease of reference; guides have been added to the main menu, the menu has been reordered, and various modules have been renamed.
- Fixed typo in workload guide.

4.16.4 CBC SDK 1.4.2 - Released March 22, 2023

New Features:

- Policy Rule Configurations - allows users to make adjustments to Carbon Black-defined rules.
- Core Prevention Rule Configurations - controls settings for core prevention rules as supplied by Carbon Black.
- Observations - search through all the noteworthy, searchable activity that was reported by your organization's sensors.
- Auth Events - visibility into authentication events on Windows endpoints.

Updates:

- Remove use of v1 status URL from process search, which now depends entirely on v2 operations.
- Vulnerabilities can now be dismissed and undismissed, and have dismissals edited.

Bug Fixes:

- User creation: raise error if the API object is not passed as the first parameter to `User.create()`.
- Live Response: pass failed session exception back up to the `WorkItem` future objects.
- Improved query string parameter handling in API calls.

Documentation:

- New example script showing how to retrieve container alerts.
- New example script allows exporting users with grant and role information.
- Bug fixed in `policy_service_crud_operations.py` example script affecting iteration over rules.
- Update clarifying alert filtering by fields that take an empty list.
- Sample script added for retrieving alerts for multiple organizations.

4.16.5 CBC SDK 1.4.1 - Released October 21, 2022

New Features:

- AWS workloads now supported in VM Workloads Search.
- Live Query Differential Analysis functionality.

Updates:

- VM Workloads Search updated to use new v2 APIs
- Added the `alertable` field to feeds.
- Devices API now supports faceting on three additional (public cloud related) fields.
- Added a user acceptance test script for the policy function updates.

Documentation:

- Added information on OAuth authentication to docs.

4.16.6 CBC SDK 1.4.0 - Released July 26,2022

Breaking Changes:

- Policy object has been moved from `cbc_sdk.endpoint_standard` to `cbc_sdk.platform`, as it now uses the new Policy Services API rather than the old APIs through Integration Services.
 - **N.B.:** This change means that you *must* use a custom API key with permissions under `org.policies` to manage policies, rather than an older “API key.”
 - To enable time to update integration logic, the `cbc_sdk.endpoint_standard` Policy object may still be imported from the old package, and supports operations that are backwards-compatible with the old one.
 - When developing a new integration, or updating an existing one `cbc_sdk.platform` should be used. There is a utility class `PolicyBuilder`, and as features are added to the Carbon Black Cloud, they will be added to this module.
- Official support for Python 3.6 has been dropped, since that version is now end-of-life. Added explicit testing support for Python versions 3.9 and 3.10. **N.B.:** End users should update their Python version to 3.7.x or greater.

New Features:

- Credentials handler now supports OAuth tokens.
- Added support for querying a single `Report` from a `Feed`.
- Added support for alert notes (create, delete, get, refresh).

Updates:

- Removed the (unused) `revoked` property from `Grant` objects.
- Increased the asynchronous query thread pool to 3 threads by default.
- Required version of `lxml` is now 4.9.1.
- Added a user acceptance test script for Alerts.

Bug Fixes:

- Added `max_rows` to USB device query, fixing pagination.
- Fixed an off-by-one error in Alerts Search resulting un duplicate alerts showing up in results.
- Fixed an error in alert faceting operations due to sending excess input to the server.

Documentation:

- Watchlists, Feeds, and Reports guide has been updated with additional clarification and examples.
- Updated description for some `Device` fields that are never populated.
- Additional sensor states added to `Device` documentation.
- Fixed the description of `BaseAlertSearchQuery.set_types` so that it mentions all valid alert types.
- Threat intelligence example has been deprecated.

4.16.7 CBC SDK 1.3.6 - Released April 19, 2022

New Features:

- Support for Device Facet API.
- Dynamic reference of query classes—now you can do `api.select("Device")` in addition to `api.select(Device)`.
- Support for Container Runtime Alerts.
- NSX Remediation functionality - set the NSX remediation state for workloads which support it.

Updates:

- Endpoint Standard specific Events have been decommissioned and removed.
- SDK now uses Watchlist Manager APIs v3 instead of v2. v2 APIs are being decommissioned.

Documentation:

- Added a CONTRIBUTING link to the README.md file.
- Change to Watchlist/Report documentation to properly reflect how to update a Report in a Watchlist.
- Cleaned up formatting.

4.16.8 CBC SDK 1.3.5 - Released January 26, 2022

New Features:

- Added asynchronous query support to Live Query.
- Added the ability to export query results from Live Query, either synchronously or asynchronously (via the Job object and the Jobs API). Synchronous exports include full-file export, line-by-line export, and ZIP file export. Asynchronous exports include full-file export and line-by-line export.
- Added a CredentialProvider that uses AWS Secrets Manager to store credential information.

Updates:

- Added `WatchlistAlert.get_process()` method to return the Process of a WatchlistAlert.
- Added several helpers to Live Query support to make it easier to get runs from a template, or results, device summaries, or facets from a run.
- Optimized API requests when performing query slicing.
- Updated pretty-printing of objects containing dict members.
- lxml dependency updated to version 4.6.5.

Bug Fixes:

- `User.delete()` now checks for an outstanding access grant on the user, and deletes it first if it exists.
- Fixed handling of URL when attaching a new IOC to a Feed.
- Getting and setting of Report ignore status is now supported even if that Report is part of a Feed.

Documentation:

- Information added about the target audience for the SDK.
- Improper reference to a credential property replaced in the Authentication guide.
- Broken example updated in Authentication guide.

- Added SDK guides for Vulnerabilities and Live Query APIs.
- Updated documentation for ProcessFacet model to better indicate support for full query string.

4.16.9 CBC SDK 1.3.4 - Released October 12, 2021

New Features:

- New CredentialProvider supporting Keychain storage of credentials (Mac OS only).
- Recommendations API - suggested reputation overrides for policy configuration.

Updates:

- Improved string representation of objects through `__str__()` mechanism.

Bug Fixes:

- Ensure proper `TimeoutError` is raised in several places where the wrong exception was being raised.
- Fix to allowed categories when performing alert queries.

Documentation Changes:

- Added guide page for alerts.
- Live Response documentation updated to note use of custom API keys.
- Clarified query examples in Concepts.
- Note that vulnerability assessment has been moved from `workload` to `platform`.
- Small typo fixes in watchlists, feeds, UBS, and reports guide.

4.16.10 CBC SDK 1.3.3 - Released August 10, 2021

Bug Fixes:

- Dependency fix on schema library.

4.16.11 CBC SDK 1.3.2 - Released August 10, 2021

New Features:

- Added asynchronous query options to Live Response APIs.
- Added functionality for Watchlists, Reports, and Feeds to simplify developer interaction.

Updates:

- Added documentation on the mapping between permissions and Live Response commands.

Bug Fixes:

- Fixed an error using the STIX/TAXII example with Cabby.
- Fixed a potential infinite loop in getting detailed search results for enriched events and processes.
- Comparison now case-insensitive on UBS download.

4.16.12 CBC SDK 1.3.1 - Released June 15, 2021

New Features:

- Allow the SDK to accept a pre-configured Session object to be used for access, to get around unusual configuration requirements.

Bug Fixes:

- Fix functions in Grant object for adding a new access profile to a user access grant.

4.16.13 CBC SDK 1.3.0 - Released June 8, 2021

New Features

- Add User Management, Grants, Access Profiles, Permitted Roles
- Move Vulnerability models to Platform package in preparation for supporting Endpoints and Workloads
- Refactor Vulnerability models
 - `VulnerabilitySummary.get_org_vulnerability_summary` static function changed to `Vulnerability.OrgSummary` model with query class
 - `VulnerabilitySummary` model moved inside `Vulnerability` to `Vulnerability.AssetView` sub model
 - `OrganizationalVulnerability` and `Vulnerability` consolidated into a single model to include Carbon Black Cloud context and CVE information together
 - `Vulnerability(cb, CVE_ID)` returns Carbon Black Cloud context and CVE information
 - `DeviceVulnerability.get_vulnerability_summary_per_device` static function moved to `get_vulnerability_summary` function on `Device` model
 - `affected_assets(os_product_id)` function changed to `get_affected_assets()` function and no longer requires `os_product_id`
- Add dashboard export examples
- Live Response migrated from v3 to v6 ([migration guide](#))
 - Live Response uses API Keys of type Custom
- Add function to get Enriched Events for Alert

Bug Fixes

- Fix validate query from dropping `sort_by` for Query class
- Fix the ability to set expiration for binary download URL
- Fix bug in helpers `read_iocs` functionality
- Fix `install_sensor` and `bulk_install` on `ComputeResource` to use `id` instead of `uuid`
- Fix `DeviceSearchQuery` from duplicating `Device` due to base index of 1

4.16.14 CBC SDK 1.2.3 - Released April 19, 2021

Bug Fixes

- Prevent alert query from retrieving past 10k limit

4.16.15 CBC SDK 1.2.3 - Released April 19, 2021

Bug Fixes

- Prevent alert query from retrieving past 10k limit

4.16.16 CBC SDK 1.2.2 - Released April 5, 2021

Bug Fixes

- Add support for full credential property loading through BaseAPI constructor

4.16.17 CBC SDK 1.2.1 - Released March 31, 2021

New Features

- Add `__str__` functions for `Process.Tree` and `Process.Summary`
- Add `get_details` for `Process`
- Add `set_max_rows` to `DeviceQuery`

Bug Fixes

- Modify base class for `EnrichedEventQuery` to `Query` from `cbc_sdk.base` to support entire feature set for searching
- Document fixes for changelog and Workload
- Fix `_spawn_new_workers` to correctly find active devices for Carbon Black Cloud

4.16.18 CBC SDK 1.2.0 - Released March 9, 2021

New Features

- VMware Carbon Black Cloud Workload support for managing workloads:
 - Vulnerability Assessment
 - Sensor Lifecycle Management
 - VM Workloads Search
- Add tutorial for Reputation Override

Bug Fixes

- Fix to initialization of `ReputationOverride` objects

4.16.19 CBC SDK 1.1.1 - Released February 2, 2021

New Features

- Add easy way to add single approvals and blocks
- Add Device Control Alerts
- Add deployment_type support to the Device model

Bug Fixes

- Fix error when updating iocs in a Report model
- Set max_retries to None to use Connection init logic for retries

4.16.20 CBC SDK 1.1.0 - Released January 27, 2021

New Features

- Reputation Overrides for Endpoint Standard with Enterprise EDR support coming soon
- Device Control for Endpoint Standard
- Live Query Templates/Scheduled Runs and Template History
- Add set_time_range for Alert query

Bug Fixes

- Refactored code base to reduce query inheritance complexity
- Limit Live Query results to 10k cap to prevent 400 Bad Request
- Add missing criteria for Live Query RunHistory to search on template ids
- Add missing args.orgkey to get_cb_cloud_object to prevent exception from being thrown
- Refactor add and update criteria to use CriteriaBuilderSupportMixin

4.16.21 CBC SDK 1.0.1 - Released December 17, 2020

Bug Fixes

- Fix readme links
- Few ReadTheDocs fixes

4.16.22 CBC SDK 1.0.0 - Released December 16, 2020

New Features

- Enriched Event searches for Endpoint Standard
- Aggregation search added for Enriched Event Query
- Add support for fetching additional details for an Enriched Event
- Facet query support for Enriched Events, Processes, and Process Events
- Addition of Python Futures to support asynchronous calls for customers who want to leverage that feature , while continuing to also provide the simplified experience which hides the multiple calls required.

- Added translation support for MISP threat intel to cbc_sdk threat intel example

Updates

- Improved information and extra calls for Audit and Remediation (Live Query)
- Great test coverage – create extensions and submit PRs with confidence
- Process and Process Event searches updated to latest APIs and moved to platform package
- Flake8 formatting applied to all areas of the code
- Converted old docstrings to use google format docstrings
- Migrated STIX/TAXII Threat Intel module from cbapi to cbc_sdk examples

Bug Fixes

- Fixed off by one error for process event pagination
- Added support for default profile using CBCloudAPI()
- Retry limit to Process Event search to prevent infinite loop

4.17 Exceptions

If an error occurs, the API attempts to roll the error into an appropriate Exception class.

4.17.1 Exception Classes

exception ApiError(*message=None, original_exception=None*)

Base class for all CBC SDK errors; also raised for generic internal errors.

Initialize the ApiError.

Parameters

- **message** (*str*) – The actual error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

exception CredentialError(*message=None, original_exception=None*)

The credentials had an unspecified error.

Initialize the ApiError.

Parameters

- **message** (*str*) – The actual error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

exception ServerError(*error_code, message, **kwargs*)

A ServerError is raised when an HTTP 5xx error code is returned from the Carbon Black server.

Initialize the ServerError.

Parameters

- **error_code** (*int*) – The error code that was received from the server.
- **message** (*str*) – The actual error message.

- **kwargs** (*dict*) – Additional arguments, which may include ‘result’ (server operation result), ‘original_exception’ (exception causing this one to be raised), and ‘uri’ (URI being accessed when this error was raised).

exception ObjectNotFoundError(*uri, message=None, original_exception=None*)

The requested object could not be found in the Carbon Black datastore.

Initialize the ObjectNotFoundError.

Parameters

- **uri** (*str*) – The URI of the action that failed.
- **message** (*str*) – The error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

exception MoreThanOneResultError(*message=None, original_exception=None, results=None*)

Only one object was requested, but multiple matches were found in the Carbon Black datastore.

Initialize the MoreThanOneResultError.

Parameters

- **message** (*str*) – The actual error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.
- **results** (*list*) – List of results returned

exception InvalidObjectError(*message=None, original_exception=None*)

An invalid object was received by the server.

Initialize the ApiError.

Parameters

- **message** (*str*) – The actual error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

exception TimeoutError(*uri=None, error_code=None, message=None, original_exception=None*)

A requested operation timed out.

Initialize the TimeoutError.

Parameters

- **uri** (*str*) – The URI of the action that timed out.
- **error_code** (*int*) – The error code that was received from the server.
- **message** (*str*) – The error message.
- **original_exception** (*Exception*) – The exception that caused this one to be raised.

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

C

- `cbc_sdk.audit_remediation.base`, 127
- `cbc_sdk.audit_remediation.differential`, 163
- `cbc_sdk.base`, 556
- `cbc_sdk.cache.lru`, 553
- `cbc_sdk.connection`, 579
- `cbc_sdk.credential_providers.aws_sm_credential_provider`, 169
- `cbc_sdk.credential_providers.default`, 168
- `cbc_sdk.credential_providers.enviro_n_credential_provider`, 169
- `cbc_sdk.credential_providers.file_credential_provider`, 170
- `cbc_sdk.credential_providers.keychain_credential_provider`, 170
- `cbc_sdk.credential_providers.registry_credential_provider`, 171
- `cbc_sdk.credentials`, 589
- `cbc_sdk.endpoint_standard.base`, 172
- `cbc_sdk.endpoint_standard.recommendation`, 180
- `cbc_sdk.endpoint_standard.usb_device_control`, 189
- `cbc_sdk.enterprise_edr.auth_events`, 204
- `cbc_sdk.enterprise_edr.threat_intelligence`, 216
- `cbc_sdk.enterprise_edr.ubs`, 242
- `cbc_sdk.errors`, 591
- `cbc_sdk.helpers`, 596
- `cbc_sdk.live_response_api`, 597
- `cbc_sdk.platform.alerts`, 246
- `cbc_sdk.platform.asset_groups`, 353
- `cbc_sdk.platform.audit`, 364
- `cbc_sdk.platform.base`, 245
- `cbc_sdk.platform.devices`, 365
- `cbc_sdk.platform.events`, 385
- `cbc_sdk.platform.grants`, 398
- `cbc_sdk.platform.jobs`, 406
- `cbc_sdk.platform.legacy_alerts`, 410
- `cbc_sdk.platform.network_threat_metadata`, 423
- `cbc_sdk.platform.observations`, 424
- `cbc_sdk.platform.policies`, 435
- `cbc_sdk.platform.policy_ruleconfigs`, 451
- `cbc_sdk.platform.previewer`, 464
- `cbc_sdk.platform.processes`, 465
- `cbc_sdk.platform.reputation`, 480
- `cbc_sdk.platform.users`, 484
- `cbc_sdk.platform.vulnerability_assessment`, 491
- `cbc_sdk.utils`, 619
- `cbc_sdk.winerror`, 620
- `cbc_sdk.workload.nsx_remediation`, 515
- `cbc_sdk.workload.sensor_lifecycle`, 516
- `cbc_sdk.workload.vm_workloads_search`, 520

A

- accept() (*Recommendation method*), 184
- add_criteria() (*AffectedAssetQuery method*), 491
- add_criteria() (*AlertSearchQuery method*), 257
- add_criteria() (*AssetGroupQuery method*), 361
- add_criteria() (*AsyncProcessQuery method*), 466
- add_criteria() (*AuthEventQuery method*), 210
- add_criteria() (*AWSComputeResourceQuery method*), 522
- add_criteria() (*BaseComputeResourceQuery method*), 533
- add_criteria() (*CriteriaBuilderSupportMixin method*), 557
- add_criteria() (*DeviceSearchQuery method*), 376
- add_criteria() (*DifferentialQuery method*), 165
- add_criteria() (*EnrichedEventQuery method*), 175
- add_criteria() (*EventFacetQuery method*), 388
- add_criteria() (*EventQuery method*), 393
- add_criteria() (*FacetQuery method*), 130, 559
- add_criteria() (*GroupedAlertSearchQuery method*), 305
- add_criteria() (*ObservationQuery method*), 429
- add_criteria() (*Query method*), 569
- add_criteria() (*RecommendationQuery method*), 186
- add_criteria() (*ResultQuery method*), 139
- add_criteria() (*RunHistoryQuery method*), 150
- add_criteria() (*SensorKitQuery method*), 517
- add_criteria() (*TemplateHistoryQuery method*), 161
- add_criteria() (*USBDeviceApprovalQuery method*), 194
- add_criteria() (*USBDeviceQuery method*), 199
- add_criteria() (*VCenterComputeResourceQuery method*), 540
- add_criteria() (*VulnerabilityAssetViewQuery method*), 501
- add_criteria() (*VulnerabilityQuery method*), 508
- add_descriptions() (*PolicyQuery method*), 447
- add_directory_action_rule() (*Policy.PolicyBuilder method*), 436
- add_exclusions() (*AlertSearchQuery method*), 257
- add_exclusions() (*AsyncProcessQuery method*), 466
- add_exclusions() (*AuthEventQuery method*), 210
- add_exclusions() (*EnrichedEventQuery method*), 176
- add_exclusions() (*EventFacetQuery method*), 388
- add_exclusions() (*EventQuery method*), 394
- add_exclusions() (*ExclusionBuilderSupportMixin method*), 558
- add_exclusions() (*FacetQuery method*), 560
- add_exclusions() (*GroupedAlertSearchQuery method*), 305
- add_exclusions() (*ObservationQuery method*), 430
- add_exclusions() (*Query method*), 570
- add_facet_field() (*EventFacetQuery method*), 389
- add_facet_field() (*FacetQuery method*), 560
- add_grant_profile() (*User.UserBuilder method*), 485
- add_headers() (*CBCSDKSessionAdapter method*), 584
- add_ioc() (*Report.ReportBuilder method*), 228
- add_members() (*AssetGroup method*), 354
- add_names() (*PolicyQuery method*), 447
- add_note() (*ApiError method*), 591
- add_note() (*ClientError method*), 591
- add_note() (*ConnectionError method*), 591
- add_note() (*CredentialError method*), 592
- add_note() (*FunctionalityDecommissioned method*), 592
- add_note() (*InvalidHashError method*), 592
- add_note() (*InvalidObjectError method*), 592
- add_note() (*LiveResponseError method*), 608
- add_note() (*ModelNotFound method*), 593
- add_note() (*MoreThanOneResultError method*), 593
- add_note() (*NonQueryableModel method*), 594
- add_note() (*NSXJobError method*), 593
- add_note() (*ObjectNotFoundError method*), 594
- add_note() (*OperationCancelled method*), 594
- add_note() (*QuerySyntaxError method*), 594
- add_note() (*ServerError method*), 595
- add_note() (*TimeoutError method*), 595
- add_note() (*UnauthorizedError method*), 596
- add_org() (*Grant.ProfileBuilder method*), 402
- add_policy_ids() (*PolicyQuery method*), 447
- add_principal() (*GrantQuery method*), 405
- add_priorities() (*PolicyQuery method*), 448
- add_profiles() (*User method*), 486
- add_range() (*EventFacetQuery method*), 389

`add_range()` (*FacetQuery* method), 560
`add_report_ids()` (*Watchlist* method), 237
`add_report_ids()` (*Watchlist.WatchlistBuilder* method), 236
`add_reports()` (*Feed.FeedBuilder* method), 216
`add_reports()` (*Watchlist* method), 237
`add_reports()` (*Watchlist.WatchlistBuilder* method), 236
`add_role()` (*Grant.GrantBuilder* method), 399
`add_role()` (*Grant.ProfileBuilder* method), 402
`add_rule()` (*Policy* method), 441
`add_rule()` (*Policy.PolicyBuilder* method), 436
`add_rule_config()` (*Policy.PolicyBuilder* method), 437
`add_rule_config_copy()` (*Policy.PolicyBuilder* method), 437
`add_rule_copy()` (*Policy.PolicyBuilder* method), 437
`add_sensor_kit_type()` (*SensorKitQuery* method), 518
`add_sensor_setting()` (*Policy.PolicyBuilder* method), 437
`add_tag()` (*Report.ReportBuilder* method), 228
`add_threat_tags()` (*Alert* method), 252
`add_threat_tags()` (*CBAAnalyticsAlert* method), 282
`add_threat_tags()` (*ContainerRuntimeAlert* method), 291
`add_threat_tags()` (*DeviceControlAlert* method), 298
`add_threat_tags()` (*HostBasedFirewallAlert* method), 329
`add_threat_tags()` (*IntrusionDetectionSystemAlert* method), 339
`add_threat_tags()` (*WatchlistAlert* method), 348
`add_time_criteria()` (*AlertSearchQuery* method), 258
`add_time_criteria()` (*GroupedAlertSearchQuery* method), 305
`add_to_groups()` (*Device* method), 367
`add_to_groups_by_id()` (*Device* method), 367
`AffectedAssetQuery` (class in *cbc_sdk.platform.vulnerability_assessment*), 491
`aggregation()` (*EnrichedEventQuery* method), 176
`Alert` (class in *cbc_sdk.platform.alerts*), 246
`Alert.Note` (class in *cbc_sdk.platform.alerts*), 251
`alert_search_suggestions()` (*CBCloudAPI* method), 118
`AlertSearchQuery` (class in *cbc_sdk.platform.alerts*), 257
`all()` (*AffectedAssetQuery* method), 491
`all()` (*AlertSearchQuery* method), 258
`all()` (*AssetGroupQuery* method), 362
`all()` (*AsyncProcessQuery* method), 467
`all()` (*AuthEventQuery* method), 211
`all()` (*AWSComputeResourceQuery* method), 522
`all()` (*BaseComputeResourceQuery* method), 533
`all()` (*DeviceSearchQuery* method), 376
`all()` (*DifferentialQuery* method), 165
`all()` (*EnrichedEventQuery* method), 176
`all()` (*EventQuery* method), 394
`all()` (*FacetQuery* method), 130
`all()` (*FeedQuery* method), 221
`all()` (*GrantQuery* method), 405
`all()` (*GroupedAlertSearchQuery* method), 306
`all()` (*IterableQueryMixin* method), 565
`all()` (*JobQuery* method), 409
`all()` (*ObservationQuery* method), 430
`all()` (*PaginatedQuery* method), 568
`all()` (*PolicyQuery* method), 448
`all()` (*Query* method), 570
`all()` (*RecommendationQuery* method), 186
`all()` (*ReportQuery* method), 233
`all()` (*ReputationOverrideQuery* method), 482
`all()` (*ResultQuery* method), 139
`all()` (*RunHistoryQuery* method), 150
`all()` (*SensorKitQuery* method), 518
`all()` (*SimpleQuery* method), 576
`all()` (*TemplateHistoryQuery* method), 161
`all()` (*USBDeviceApprovalQuery* method), 194
`all()` (*USBDeviceBlockQuery* method), 199
`all()` (*USBDeviceQuery* method), 200
`all()` (*UserQuery* method), 490
`all()` (*VCenterComputeResourceQuery* method), 540
`all()` (*VulnerabilityAssetViewQuery* method), 501
`all()` (*VulnerabilityQuery* method), 508
`all()` (*WatchlistQuery* method), 240
`allowed_orgs` (*Grant.Profile* property), 400
`and_()` (*AffectedAssetQuery* method), 491
`and_()` (*AlertSearchQuery* method), 258
`and_()` (*AssetGroupQuery* method), 362
`and_()` (*AsyncProcessQuery* method), 467
`and_()` (*AuthEventQuery* method), 211
`and_()` (*AWSComputeResourceQuery* method), 522
`and_()` (*BaseComputeResourceQuery* method), 533
`and_()` (*DeviceSearchQuery* method), 376
`and_()` (*EnrichedEventQuery* method), 176
`and_()` (*EventFacetQuery* method), 390
`and_()` (*EventQuery* method), 394
`and_()` (*FacetQuery* method), 130, 561
`and_()` (*FeedQuery* method), 222
`and_()` (*GroupedAlertSearchQuery* method), 306
`and_()` (*ObservationQuery* method), 430
`and_()` (*Query* method), 570
`and_()` (*QueryBuilder* method), 574
`and_()` (*QueryBuilderSupportMixin* method), 575
`and_()` (*ReportQuery* method), 233
`and_()` (*ReputationOverrideQuery* method), 482
`and_()` (*ResultQuery* method), 139
`and_()` (*RunHistoryQuery* method), 150

- `and_()` (*SimpleQuery* method), 576
 - `and_()` (*SummaryQuery* method), 478
 - `and_()` (*TemplateHistoryQuery* method), 161
 - `and_()` (*USBDeviceApprovalQuery* method), 194
 - `and_()` (*USBDeviceQuery* method), 200
 - `and_()` (*VCenterComputeResourceQuery* method), 540
 - `and_()` (*VulnerabilityAssetViewQuery* method), 501
 - `and_()` (*VulnerabilityQuery* method), 508
 - `and_()` (*WatchlistQuery* method), 241
 - `api_json_request()` (*BaseAPI* method), 580
 - `api_json_request()` (*CBCloudAPI* method), 118
 - `api_request_iterate()` (*BaseAPI* method), 580
 - `api_request_iterate()` (*CBCloudAPI* method), 118
 - `api_request_stream()` (*BaseAPI* method), 581
 - `api_request_stream()` (*CBCloudAPI* method), 119
 - `ApiError`, 591, 635
 - `append()` (*Vulnerability.AssetView* method), 498
 - `append_iocs()` (*Report* method), 229
 - `append_reports()` (*Feed* method), 217
 - `append_reports_rawdata()` (*Feed* method), 218
 - `append_rule()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup* method), 458
 - `append_rule_group()` (*HostBasedFirewallRuleConfig* method), 459
 - `application_` (*Recommendation.RecommendationNewRule* property), 183
 - `approve()` (*USBDevice* method), 189
 - `approve_process_sha256()` (*EnrichedEvent* method), 172
 - `approve_process_sha256()` (*Process* method), 473
 - `ArrayFieldDescriptor` (class in *cbc_sdk.base*), 556
 - `asset_count` (*DevicePolicyChangePreview* property), 465
 - `asset_query` (*DevicePolicyChangePreview* property), 465
 - `AssetGroup` (class in *cbc_sdk.platform.asset_groups*), 353
 - `AssetGroupQuery` (class in *cbc_sdk.platform.asset_groups*), 361
 - `assets` (*DevicePolicyChangePreview* property), 465
 - `async_await_result()` (*NSXRemediationJob* method), 515
 - `async_export()` (*DifferentialQuery* method), 165
 - `async_export()` (*ResultQuery* method), 139
 - `ASYNC_RATE_LIMIT` (in module *cbc_sdk.audit_remediation.differential*), 163
 - `AsyncProcessQuery` (class in *cbc_sdk.platform.processes*), 466
 - `AsyncQueryMixin` (class in *cbc_sdk.base*), 556
 - `audit_remediation()` (*CBCloudAPI* method), 119
 - `audit_remediation_history()` (*CBCloudAPI* method), 119
 - `AuditLog` (class in *cbc_sdk.platform.audit*), 364
 - `AuthEvent` (class in *cbc_sdk.enterprise_edr.auth_events*), 204
 - `AuthEventFacet` (class in *cbc_sdk.enterprise_edr.auth_events*), 207
 - `AuthEventFacet.Ranges` (class in *cbc_sdk.enterprise_edr.auth_events*), 207
 - `AuthEventFacet.Terms` (class in *cbc_sdk.enterprise_edr.auth_events*), 208
 - `AuthEventGroup` (class in *cbc_sdk.enterprise_edr.auth_events*), 209
 - `AuthEventQuery` (class in *cbc_sdk.enterprise_edr.auth_events*), 210
 - `await_completion()` (*Job* method), 407
 - `await_result()` (*NSXRemediationJob* method), 515
 - `AWSComputeResource` (class in *cbc_sdk.workload.vm_workloads_search*), 520
 - `AWSComputeResourceQuery` (class in *cbc_sdk.workload.vm_workloads_search*), 521
 - `AWSCredentialProvider` (class in *cbc_sdk.credential_providers.aws_sm_credential_provider*), 169
- ## B
- `background_scan()` (*Device* method), 368
 - `background_scan()` (*DeviceSearchQuery* method), 376
 - `ban_process_sha256()` (*EnrichedEvent* method), 172
 - `ban_process_sha256()` (*Process* method), 473
 - `BaseAPI` (class in *cbc_sdk.connection*), 579
 - `BaseComputeResource` (class in *cbc_sdk.workload.vm_workloads_search*), 531
 - `BaseComputeResourceQuery` (class in *cbc_sdk.workload.vm_workloads_search*), 533
 - `BaseQuery` (class in *cbc_sdk.base*), 556
 - `batch_size()` (*AsyncProcessQuery* method), 467
 - `batch_size()` (*AuthEventQuery* method), 211
 - `batch_size()` (*EnrichedEventQuery* method), 176
 - `batch_size()` (*EventQuery* method), 395
 - `batch_size()` (*ObservationQuery* method), 430
 - `batch_size()` (*PaginatedQuery* method), 568
 - `batch_size()` (*Query* method), 570
 - `Binary` (class in *cbc_sdk.enterprise_edr.ubs*), 242
 - `Binary.Summary` (class in *cbc_sdk.enterprise_edr.ubs*), 242
 - `BinaryFieldDescriptor` (class in *cbc_sdk.base*), 557
 - `build()` (*Feed.FeedBuilder* method), 216
 - `build()` (*Grant.GrantBuilder* method), 399
 - `build()` (*Grant.ProfileBuilder* method), 402
 - `build()` (*Policy.PolicyBuilder* method), 438
 - `build()` (*Report.ReportBuilder* method), 228

`build()` (*User.UserBuilder method*), 485
`build()` (*Watchlist.WatchlistBuilder method*), 236
`build_cli_parser()` (in module *cbc_sdk.helpers*), 596
`build_response()` (*CBCSDKSessionAdapter method*), 584
`bulk_add_profiles()` (*User class method*), 486
`bulk_create()` (*USBDeviceApproval class method*), 191
`bulk_create()` (*USBDeviceBlock class method*), 197
`bulk_create()` (*User class method*), 486
`bulk_create_csv()` (*USBDeviceApproval class method*), 192
`bulk_delete()` (*ReputationOverride class method*), 480
`bulk_delete()` (*User class method*), 487
`bulk_disable_all_access()` (*User class method*), 487
`bulk_disable_profiles()` (*User class method*), 487
`bulk_get_details()` (*AuthEvent static method*), 205
`bulk_get_details()` (*Observation static method*), 425
`bulk_install()` (*AWSComputeResource class method*), 520
`bulk_install()` (*BaseComputeResource class method*), 531
`bulk_install()` (*VCenterComputeResource class method*), 538
`bulk_install_by_id()` (*AWSComputeResource class method*), 520
`bulk_install_by_id()` (*BaseComputeResource class method*), 531
`bulk_install_by_id()` (*VCenterComputeResource class method*), 539
`bulk_threat_dismiss()` (*CBCloudAPI method*), 119
`bulk_threat_update()` (*CBCloudAPI method*), 120
`bypass()` (*Device method*), 368
`bypass()` (*DeviceSearchQuery method*), 377

C

`cancel_command()` (*CbLRSessionBase method*), 598
`cancel_command()` (*LiveResponseSession method*), 610
`CBAAnalyticsAlert` (class in *cbc_sdk.platform.alerts*), 278
`CBAAnalyticsAlert.Note` (class in *cbc_sdk.platform.alerts*), 282
`cbc_sdk.audit_remediation.base` module, 127
`cbc_sdk.audit_remediation.differential` module, 163
`cbc_sdk.base` module, 556
`cbc_sdk.cache.lru` module, 553
`cbc_sdk.connection` module, 579
`cbc_sdk.credential_providers.aws_sm_credential_provider`

module, 169
`cbc_sdk.credential_providers.default` module, 168
`cbc_sdk.credential_providers.environ_credential_provider` module, 169
`cbc_sdk.credential_providers.file_credential_provider` module, 170
`cbc_sdk.credential_providers.keychain_credential_provider` module, 170
`cbc_sdk.credential_providers.registry_credential_provider` module, 171
`cbc_sdk.credentials` module, 589
`cbc_sdk.endpoint_standard.base` module, 172
`cbc_sdk.endpoint_standard.recommendation` module, 180
`cbc_sdk.endpoint_standard.usb_device_control` module, 189
`cbc_sdk.enterprise_edr.auth_events` module, 204
`cbc_sdk.enterprise_edr.threat_intelligence` module, 216
`cbc_sdk.enterprise_edr.ubs` module, 242
`cbc_sdk.errors` module, 591
`cbc_sdk.helpers` module, 596
`cbc_sdk.live_response_api` module, 597
`cbc_sdk.platform.alerts` module, 246
`cbc_sdk.platform.asset_groups` module, 353
`cbc_sdk.platform.audit` module, 364
`cbc_sdk.platform.base` module, 245
`cbc_sdk.platform.devices` module, 365
`cbc_sdk.platform.events` module, 385
`cbc_sdk.platform.grants` module, 398
`cbc_sdk.platform.jobs` module, 406
`cbc_sdk.platform.legacy_alerts` module, 410
`cbc_sdk.platform.network_threat_metadata` module, 423
`cbc_sdk.platform.observations` module, 424
`cbc_sdk.platform.policies`

- module, 435
- cbc_sdk.platform.policy_ruleconfigs
 - module, 451
- cbc_sdk.platform.previewer
 - module, 464
- cbc_sdk.platform.processes
 - module, 465
- cbc_sdk.platform.reputation
 - module, 480
- cbc_sdk.platform.users
 - module, 484
- cbc_sdk.platform.vulnerability_assessment
 - module, 491
- cbc_sdk.utils
 - module, 619
- cbc_sdk.winerror
 - module, 620
- cbc_sdk.workload.nsx_remediation
 - module, 515
- cbc_sdk.workload.sensor_lifecycle
 - module, 516
- cbc_sdk.workload.vm_workloads_search
 - module, 520
- CBCloudAPI (class in cbc_sdk.rest_api), 117
- CBCSDKSessionAdapter (class in cbc_sdk.connection), 584
- cbldr_session_cls (LiveResponseSessionManager attribute), 617
- CbLRManagerBase (class in cbc_sdk.live_response_api), 597
- CbLRSessionBase (class in cbc_sdk.live_response_api), 598
- CbMetaModel (class in cbc_sdk.base), 557
- cert_verify() (CBCSDKSessionAdapter method), 584
- change_role() (User method), 487
- check_python_tls_compatibility() (in module cbc_sdk.connection), 588
- check_state_key() (SwaggerLoader method), 578
- children (Process property), 473
- classifier_ (Watchlist property), 237
- clear() (Vulnerability.AssetView method), 498
- ClientError, 591
- close() (Alert method), 252
- close() (AlertSearchQuery method), 259
- close() (CBAnalyticsAlert method), 283
- close() (CBCSDKSessionAdapter method), 585
- close() (CbLRSessionBase method), 598
- close() (ContainerRuntimeAlert method), 291
- close() (DeviceControlAlert method), 298
- close() (GroupedAlertSearchQuery method), 306
- close() (HostBasedFirewallAlert method), 330
- close() (IntrusionDetectionSystemAlert method), 339
- close() (LiveResponseSession method), 610
- close() (WatchlistAlert method), 348
- close_session() (CbLRManagerBase method), 597
- close_session() (LiveResponseSessionManager method), 617
- command_status() (CbLRSessionBase method), 598
- command_status() (LiveResponseSession method), 610
- CommDlgError (class in cbc_sdk.winerror), 620
- CompletionNotification (class in cbc_sdk.live_response_api), 605
- ComputeResourceFacet (class in cbc_sdk.workload.vm_workloads_search), 537
- ComputeResourceFacet.ComputeResourceFacetValue (class in cbc_sdk.workload.vm_workloads_search), 537
- config_params() (SensorKitQuery method), 518
- Connection (class in cbc_sdk.connection), 586
- ConnectionError, 591
- construct_include() (in module cbc_sdk.base), 578
- ContainerRuntimeAlert (class in cbc_sdk.platform.alerts), 288
- ContainerRuntimeAlert.Note (class in cbc_sdk.platform.alerts), 290
- convert_feed_query() (CBCloudAPI method), 120
- convert_from_cb() (in module cbc_sdk.utils), 619
- convert_to_cb() (in module cbc_sdk.utils), 619
- copy() (Vulnerability.AssetView method), 498
- copy_rules() (HostBasedFirewallRuleConfig method), 459
- core_prevention_rule_configs (Policy property), 441
- core_prevention_rule_configs_list (Policy property), 441
- CorePreventionRuleConfig (class in cbc_sdk.platform.policy_ruleconfigs), 451
- count() (Vulnerability.AssetView method), 498
- count_only() (DifferentialQuery method), 166
- CreatableModelMixin (class in cbc_sdk.base), 557
- create() (BaseAPI method), 581
- create() (CBCloudAPI method), 120
- create() (Feed class method), 218
- create() (Grant class method), 403
- create() (Policy class method), 442
- create() (Report class method), 229
- create() (ReputationOverride class method), 481
- create() (USBDeviceBlock class method), 198
- create() (User class method), 487
- create() (Watchlist class method), 237
- create_directory() (CbLRSessionBase method), 598
- create_directory() (LiveResponseSession method), 610
- create_equality() (IOC_V2 class method), 224
- create_from_feed() (Watchlist class method), 238
- create_from_usb_device() (USBDeviceApproval class method), 192

- `create_group()` (*AssetGroup* class method), 355
- `create_note()` (*Alert* method), 253
- `create_note()` (*CBAnalyticsAlert* method), 283
- `create_note()` (*ContainerRuntimeAlert* method), 292
- `create_note()` (*DeviceControlAlert* method), 299
- `create_note()` (*HostBasedFirewallAlert* method), 330
- `create_note()` (*IntrusionDetectionSystemAlert* method), 340
- `create_note()` (*WatchlistAlert* method), 349
- `create_process()` (*CbLRSessionBase* method), 599
- `create_process()` (*LiveResponseSession* method), 610
- `create_profile()` (*Grant* method), 403
- `create_profile()` (*Grant.GrantBuilder* method), 399
- `create_query()` (*IOC_V2* class method), 225
- `create_regex()` (*IOC_V2* class method), 225
- `create_registry_key()` (*CbLRSessionBase* method), 599
- `create_registry_key()` (*LiveResponseSession* method), 611
- `CredentialError`, 591, 635
- `CredentialProvider` (class in *cbc_sdk.credentials*), 589
- `Credentials` (class in *cbc_sdk.credentials*), 589
- `CredentialValue` (class in *cbc_sdk.credentials*), 589
- `CriteriaBuilderSupportMixin` (class in *cbc_sdk.base*), 557
- `current_policy` (*DevicePolicyChangePreview* property), 465
- `current_policy_id` (*DevicePolicyChangePreview* property), 465
- `current_policy_position` (*DevicePolicyChangePreview* property), 465
- `custom_severities` (*CBCloudAPI* property), 121
- `custom_severity` (*Report* property), 230
- D**
- `daemon` (*JobWorker* property), 606
- `data_collection_rule_configs` (*Policy* property), 442
- `data_collection_rule_configs_list` (*Policy* property), 442
- `DataCollectionRuleConfig` (class in *cbc_sdk.platform.policy_ruleconfigs*), 453
- `decode_hresult()` (in module *cbc_sdk.winerror*), 622
- `default_action` (*HostBasedFirewallRuleConfig* property), 460
- `default_credential_provider()` (in module *cbc_sdk.credential_providers.default*), 169
- `DefaultProvider` (class in *cbc_sdk.credential_providers.default*), 168
- `delete()` (*Alert.Note* method), 251
- `delete()` (*AssetGroup* method), 355
- `delete()` (*CBAnalyticsAlert.Note* method), 282
- `delete()` (*Connection* method), 587
- `delete()` (*ContainerRuntimeAlert.Note* method), 290
- `delete()` (*CorePreventionRuleConfig* method), 451
- `delete()` (*DataCollectionRuleConfig* method), 454
- `delete()` (*DeviceControlAlert.Note* method), 298
- `delete()` (*Feed* method), 218
- `delete()` (*FeedModel* method), 220
- `delete()` (*Grant* method), 404
- `delete()` (*Grant.Profile* method), 400
- `delete()` (*HostBasedFirewallAlert.Note* method), 329
- `delete()` (*HostBasedFirewallRuleConfig* method), 460
- `delete()` (*HostBasedFirewallRuleConfig.FirewallRule* method), 456
- `delete()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup* method), 458
- `delete()` (*IntrusionDetectionSystemAlert.Note* method), 338
- `delete()` (*IOC* method), 223
- `delete()` (*IOC_V2* method), 225
- `delete()` (*LiveResponseMemdump* method), 609
- `delete()` (*MutableBaseModel* method), 566
- `delete()` (*Policy* method), 442
- `delete()` (*PolicyRule* method), 449
- `delete()` (*PolicyRuleConfig* method), 463
- `delete()` (*Report* method), 230
- `delete()` (*ReportSeverity* method), 234
- `delete()` (*ReputationOverride* method), 481
- `delete()` (*Run* method), 146
- `delete()` (*RunHistory* method), 148
- `delete()` (*Template* method), 157
- `delete()` (*TemplateHistory* method), 159
- `delete()` (*USBDeviceApproval* method), 192
- `delete()` (*USBDeviceBlock* method), 198
- `delete()` (*User* method), 487
- `delete()` (*Watchlist* method), 238
- `delete()` (*WatchlistAlert.Note* method), 348
- `delete_file()` (*CbLRSessionBase* method), 599
- `delete_file()` (*LiveResponseSession* method), 611
- `delete_object()` (*BaseAPI* method), 581
- `delete_object()` (*CBCloudAPI* method), 121
- `delete_registry_key()` (*CbLRSessionBase* method), 600
- `delete_registry_key()` (*LiveResponseSession* method), 611
- `delete_registry_value()` (*CbLRSessionBase* method), 600
- `delete_registry_value()` (*LiveResponseSession* method), 612
- `delete_rule()` (*Policy* method), 442
- `delete_rule_config()` (*Policy* method), 442
- `delete_sensor()` (*Device* method), 368
- `delete_sensor()` (*DeviceSearchQuery* method), 377
- `delete_threat_tag()` (*Alert* method), 253
- `delete_threat_tag()` (*CBAnalyticsAlert* method), 284

- [delete_threat_tag\(\)](#) (*ContainerRuntimeAlert method*), 292
[delete_threat_tag\(\)](#) (*DeviceControlAlert method*), 299
[delete_threat_tag\(\)](#) (*HostBasedFirewallAlert method*), 331
[delete_threat_tag\(\)](#) (*IntrusionDetectionSystemAlert method*), 340
[delete_threat_tag\(\)](#) (*WatchlistAlert method*), 349
[deobfuscate_cmdline\(\)](#) (*Alert method*), 253
[deobfuscate_cmdline\(\)](#) (*CBAalyticsAlert method*), 284
[deobfuscate_cmdline\(\)](#) (*ContainerRuntimeAlert method*), 292
[deobfuscate_cmdline\(\)](#) (*DeviceControlAlert method*), 299
[deobfuscate_cmdline\(\)](#) (*HostBasedFirewallAlert method*), 331
[deobfuscate_cmdline\(\)](#) (*IntrusionDetectionSystemAlert method*), 340
[deobfuscate_cmdline\(\)](#) (*Observation method*), 425
[deobfuscate_cmdline\(\)](#) (*Process method*), 473
[deobfuscate_cmdline\(\)](#) (*WatchlistAlert method*), 349
[Device](#) (*class in cbc_sdk.platform.devices*), 365
[device_](#) (*Result property*), 136
[device_background_scan\(\)](#) (*CBCloudAPI method*), 121
[device_bypass\(\)](#) (*CBCloudAPI method*), 121
[device_delete_sensor\(\)](#) (*CBCloudAPI method*), 121
[device_ids\(\)](#) (*RunQuery method*), 153
[device_quarantine\(\)](#) (*CBCloudAPI method*), 122
[device_types\(\)](#) (*RunQuery method*), 153
[device_uninstall_sensor\(\)](#) (*CBCloudAPI method*), 122
[device_update_policy\(\)](#) (*CBCloudAPI method*), 122
[device_update_sensor_version\(\)](#) (*CBCloudAPI method*), 122
[DeviceControlAlert](#) (*class in cbc_sdk.platform.alerts*), 296
[DeviceControlAlert.Note](#) (*class in cbc_sdk.platform.alerts*), 297
[DeviceFacet](#) (*class in cbc_sdk.platform.devices*), 373
[DeviceFacet.DeviceFacetValue](#) (*class in cbc_sdk.platform.devices*), 374
[deviceId](#) (*Device property*), 368
[DevicePolicyChangePreview](#) (*class in cbc_sdk.platform.previewer*), 464
[DeviceSearchQuery](#) (*class in cbc_sdk.platform.devices*), 375
[DeviceSummary](#) (*class in cbc_sdk.audit_remediation.base*), 127
[DeviceSummary.Metrics](#) (*class in cbc_sdk.audit_remediation.base*), 127
[DeviceSummaryFacet](#) (*class in cbc_sdk.audit_remediation.base*), 128
[DeviceSummaryFacet.Values](#) (*class in cbc_sdk.audit_remediation.base*), 129
[Differential](#) (*class in cbc_sdk.audit_remediation.differential*), 163
[DifferentialQuery](#) (*class in cbc_sdk.audit_remediation.differential*), 164
[DirectoryStorageError](#) (*class in cbc_sdk.winerror*), 620
[disable_alerts\(\)](#) (*Watchlist method*), 238
[disable_all_access\(\)](#) (*User method*), 487
[disable_insecure_warnings\(\)](#) (*in module cbc_sdk.helpers*), 596
[disable_profiles\(\)](#) (*User method*), 488
[disable_tags\(\)](#) (*Watchlist method*), 238
[dismiss_threat\(\)](#) (*Alert method*), 253
[dismiss_threat\(\)](#) (*CBAalyticsAlert method*), 284
[dismiss_threat\(\)](#) (*ContainerRuntimeAlert method*), 292
[dismiss_threat\(\)](#) (*DeviceControlAlert method*), 300
[dismiss_threat\(\)](#) (*HostBasedFirewallAlert method*), 331
[dismiss_threat\(\)](#) (*IntrusionDetectionSystemAlert method*), 340
[dismiss_threat\(\)](#) (*WatchlistAlert method*), 350
[download\(\)](#) (*AWSComputeResourceQuery method*), 522
[download\(\)](#) (*BaseComputeResourceQuery method*), 534
[download\(\)](#) (*DeviceSearchQuery method*), 377
[download\(\)](#) (*VCenterComputeResourceQuery method*), 541
[download_url\(\)](#) (*Binary method*), 243
[Downloads](#) (*class in cbc_sdk.enterprise_edr.ubs*), 244
[Downloads.FoundItem](#) (*class in cbc_sdk.enterprise_edr.ubs*), 244
- ## E
- [email_addresses\(\)](#) (*UserQuery method*), 490
[enable_alerts\(\)](#) (*Watchlist method*), 238
[enable_tags\(\)](#) (*Watchlist method*), 238
[enabled](#) (*HostBasedFirewallRuleConfig property*), 460
[EnrichedEvent](#) (*class in cbc_sdk.endpoint_standard.base*), 172
[EnrichedEventFacet](#) (*class in cbc_sdk.endpoint_standard.base*), 173
[EnrichedEventFacet.Ranges](#) (*class in cbc_sdk.endpoint_standard.base*), 173
[EnrichedEventFacet.Terms](#) (*class in cbc_sdk.endpoint_standard.base*), 174
[EnrichedEventQuery](#) (*class in cbc_sdk.endpoint_standard.base*), 175
[EnvironCredentialProvider](#) (*class in cbc_sdk.credential_providers.environ_credential_provider*),

- 169
- EpochDateTimeFieldDescriptor (class in *cbc_sdk.base*), 558
- eprint() (in module *cbc_sdk.helpers*), 596
- ErrorBaseClass (class in *cbc_sdk.winerror*), 620
- ErrorMetaClass (class in *cbc_sdk.winerror*), 620
- Event (class in *cbc_sdk.endpoint_standard.base*), 180
- Event (class in *cbc_sdk.platform.events*), 385
- EventFacet (class in *cbc_sdk.platform.events*), 386
- EventFacet.Ranges (class in *cbc_sdk.platform.events*), 386
- EventFacet.Terms (class in *cbc_sdk.platform.events*), 387
- EventFacetQuery (class in *cbc_sdk.platform.events*), 388
- EventQuery (class in *cbc_sdk.platform.events*), 393
- events() (Process method), 473
- exclude_appliance_uuid() (VCenterComputeResourceQuery method), 541
- exclude_auto_scaling_group_name() (AWSComputeResourceQuery method), 523
- exclude_availability_zone() (AWSComputeResourceQuery method), 523
- exclude_cloud_provider_account_id() (AWSComputeResourceQuery method), 523
- exclude_cloud_provider_resource_id() (AWSComputeResourceQuery method), 524
- exclude_cloud_provider_tags() (AWSComputeResourceQuery method), 524
- exclude_cluster_name() (VCenterComputeResourceQuery method), 541
- exclude_datacenter_name() (VCenterComputeResourceQuery method), 542
- exclude_device_guid() (VCenterComputeResourceQuery method), 542
- exclude_eligibility() (VCenterComputeResourceQuery method), 542
- exclude_eligibility_code() (VCenterComputeResourceQuery method), 542
- exclude_esx_host_name() (VCenterComputeResourceQuery method), 542
- exclude_esx_host_uuid() (VCenterComputeResourceQuery method), 543
- exclude_host_name() (VCenterComputeResourceQuery method), 543
- exclude_id() (AWSComputeResourceQuery method), 524
- exclude_installation_status() (AWSComputeResourceQuery method), 524
- exclude_installation_status() (VCenterComputeResourceQuery method), 543
- exclude_installation_type() (VCenterComputeResourceQuery method), 543
- exclude_ip_address() (VCenterComputeResourceQuery method), 543
- exclude_name() (AWSComputeResourceQuery method), 524
- exclude_name() (VCenterComputeResourceQuery method), 543
- exclude_os_architecture() (VCenterComputeResourceQuery method), 544
- exclude_os_description() (VCenterComputeResourceQuery method), 544
- exclude_os_type() (VCenterComputeResourceQuery method), 544
- exclude_platform() (AWSComputeResourceQuery method), 525
- exclude_platform_details() (AWSComputeResourceQuery method), 525
- exclude_region() (AWSComputeResourceQuery method), 525
- exclude_registration_id() (VCenterComputeResourceQuery method), 544
- exclude_subnet_id() (AWSComputeResourceQuery method), 525
- exclude_uuid() (VCenterComputeResourceQuery method), 544
- exclude_vcenter_host_url() (VCenterComputeResourceQuery method), 545
- exclude_vcenter_name() (VCenterComputeResourceQuery method), 545
- exclude_vcenter_uuid() (VCenterComputeResourceQuery method), 545
- exclude_virtual_private_cloud_id() (AWSComputeResourceQuery method), 525
- exclude_vmwaretools_version() (VCenterComputeResourceQuery method), 545
- ExclusionBuilderSupportMixin (class in *cbc_sdk.base*), 558
- execute_async() (AffectedAssetQuery method), 492
- execute_async() (AssetGroupQuery method), 362
- execute_async() (AsyncProcessQuery method), 467
- execute_async() (AsyncQueryMixin method), 556
- execute_async() (AuthEventQuery method), 211
- execute_async() (AWSComputeResourceQuery method), 526
- execute_async() (BaseComputeResourceQuery method), 534
- execute_async() (DeviceSearchQuery method), 377
- execute_async() (EnrichedEventQuery method), 177
- execute_async() (EventFacetQuery method), 390
- execute_async() (EventQuery method), 395
- execute_async() (FacetQuery method), 130, 561
- execute_async() (GrantQuery method), 406
- execute_async() (JobQuery method), 409
- execute_async() (ObservationQuery method), 430
- execute_async() (PolicyQuery method), 448
- execute_async() (Query method), 571

- execute_async() (*RecommendationQuery* method), 186
- execute_async() (*ReputationOverrideQuery* method), 482
- execute_async() (*ResultQuery* method), 140
- execute_async() (*RunHistoryQuery* method), 150
- execute_async() (*RunQuery* method), 153
- execute_async() (*SensorKitQuery* method), 518
- execute_async() (*SummaryQuery* method), 478
- execute_async() (*TemplateHistoryQuery* method), 161
- execute_async() (*USBDeviceApprovalQuery* method), 194
- execute_async() (*USBDeviceBlockQuery* method), 199
- execute_async() (*USBDeviceQuery* method), 200
- execute_async() (*UserQuery* method), 490
- execute_async() (*VCenterComputeResourceQuery* method), 545
- execute_async() (*VulnerabilityAssetViewQuery* method), 501
- execute_async() (*VulnerabilityQuery* method), 509
- expires() (*SensorKitQuery* method), 519
- export() (*AffectedAssetQuery* method), 492
- export() (*DeviceSearchQuery* method), 378
- export() (*USBDeviceApprovalQuery* method), 195
- export() (*USBDeviceQuery* method), 200
- export() (*VulnerabilityAssetViewQuery* method), 502
- export() (*VulnerabilityQuery* method), 509
- export_csv_as_file() (*ResultQuery* method), 140
- export_csv_as_lines() (*ResultQuery* method), 140
- export_csv_as_stream() (*ResultQuery* method), 140
- export_csv_as_string() (*ResultQuery* method), 140
- export_rules() (*HostBasedFirewallRuleConfig* method), 460
- export_zipped_csv() (*ResultQuery* method), 141
- extend() (*Vulnerability.AssetView* method), 498
- ## F
- facet() (*AWSCloudComputeResourceQuery* method), 526
- facet() (*BaseComputeResourceQuery* method), 534
- facet() (*VCenterComputeResourceQuery* method), 546
- facet_field() (*FacetQuery* method), 131
- FacetQuery (class in *cbc_sdk.audit_remediation.base*), 130
- FacetQuery (class in *cbc_sdk.base*), 559
- facets (*AuthEventFacet.Ranges* property), 207
- facets (*AuthEventFacet.Terms* property), 208
- facets (*EnrichedEventFacet.Ranges* property), 174
- facets (*EnrichedEventFacet.Terms* property), 174
- facets (*EventFacet.Ranges* property), 386
- facets (*EventFacet.Terms* property), 387
- facets (*ObservationFacet.Ranges* property), 427
- facets (*ObservationFacet.Terms* property), 428
- facets (*ProcessFacet.Ranges* property), 476
- facets (*ProcessFacet.Terms* property), 476
- facets() (*AlertSearchQuery* method), 259
- facets() (*DeviceSearchQuery* method), 378
- facets() (*GroupedAlertSearchQuery* method), 307
- facets() (*Process* method), 474
- facets() (*USBDeviceQuery* method), 201
- Facility (class in *cbc_sdk.winerror*), 621
- FAILED() (in module *cbc_sdk.winerror*), 621
- Feed (class in *cbc_sdk.enterprise_edr.threat_intelligence*), 216
- feed (*Watchlist* property), 239
- Feed.FeedBuilder (class in *cbc_sdk.enterprise_edr.threat_intelligence*), 216
- FeedModel (class in *cbc_sdk.enterprise_edr.threat_intelligence*), 220
- FeedQuery (class in *cbc_sdk.enterprise_edr.threat_intelligence*), 221
- fetch_process_queries() (*CBCCloudAPI* method), 123
- FieldDescriptor (class in *cbc_sdk.base*), 564
- fields (*AuthEventFacet.Ranges* property), 207
- fields (*AuthEventFacet.Terms* property), 208
- fields (*EnrichedEventFacet.Ranges* property), 174
- fields (*EnrichedEventFacet.Terms* property), 174
- fields (*EventFacet.Ranges* property), 386
- fields (*EventFacet.Terms* property), 387
- fields (*ObservationFacet.Ranges* property), 427
- fields (*ObservationFacet.Terms* property), 428
- fields (*ProcessFacet.Ranges* property), 476
- fields (*ProcessFacet.Terms* property), 477
- fields_ (*Result* property), 136
- FileCredentialProvider (class in *cbc_sdk.credential_providers.file_credential_provider*), 170
- first() (*AffectedAssetQuery* method), 492
- first() (*AlertSearchQuery* method), 259
- first() (*AssetGroupQuery* method), 362
- first() (*AsyncProcessQuery* method), 467
- first() (*AuthEventQuery* method), 211
- first() (*AWSCloudComputeResourceQuery* method), 526
- first() (*BaseComputeResourceQuery* method), 535
- first() (*DeviceSearchQuery* method), 378
- first() (*DifferentialQuery* method), 166
- first() (*EnrichedEventQuery* method), 177
- first() (*EventQuery* method), 395
- first() (*FacetQuery* method), 131
- first() (*FeedQuery* method), 222
- first() (*GrantQuery* method), 406
- first() (*GroupedAlertSearchQuery* method), 307
- first() (*IterableQueryMixin* method), 565
- first() (*JobQuery* method), 409
- first() (*ObservationQuery* method), 431
- first() (*PaginatedQuery* method), 568

`first()` (*PolicyQuery* method), 448
`first()` (*Query* method), 571
`first()` (*RecommendationQuery* method), 186
`first()` (*ReportQuery* method), 233
`first()` (*ReputationOverrideQuery* method), 482
`first()` (*ResultQuery* method), 141
`first()` (*RunHistoryQuery* method), 150
`first()` (*SensorKitQuery* method), 519
`first()` (*SimpleQuery* method), 577
`first()` (*TemplateHistoryQuery* method), 161
`first()` (*USBDeviceApprovalQuery* method), 195
`first()` (*USBDeviceBlockQuery* method), 199
`first()` (*USBDeviceQuery* method), 201
`first()` (*UserQuery* method), 490
`first()` (*VCenterComputeResourceQuery* method), 546
`first()` (*VulnerabilityAssetViewQuery* method), 502
`first()` (*VulnerabilityQuery* method), 509
`first()` (*WatchlistQuery* method), 241
`ForeignKeyFieldDescriptor` (class in *cbc_sdk.base*), 564
`found` (*Downloads* property), 244
`from_type()` (*SensorKit* class method), 516
`FunctionalityDecommissioned`, 592

G

`get()` (*Alert* method), 254
`get()` (*Alert.Note* method), 251
`get()` (*AssetGroup* method), 355
`get()` (*AuditLog* method), 364
`get()` (*AuthEvent* method), 205
`get()` (*AuthEventFacet* method), 208
`get()` (*AuthEventFacet.Ranges* method), 208
`get()` (*AuthEventFacet.Terms* method), 208
`get()` (*AWSComputeResource* method), 521
`get()` (*BaseComputeResource* method), 532
`get()` (*Binary* method), 243
`get()` (*Binary.Summary* method), 243
`get()` (*CBAnalyticsAlert* method), 284
`get()` (*CBAnalyticsAlert.Note* method), 282
`get()` (*ComputeResourceFacet* method), 538
`get()` (*ComputeResourceFacet.ComputeResourceFacetValue* method), 537
`get()` (*Connection* method), 587
`get()` (*ContainerRuntimeAlert* method), 293
`get()` (*ContainerRuntimeAlert.Note* method), 290
`get()` (*CorePreventionRuleConfig* method), 451
`get()` (*DataCollectionRuleConfig* method), 454
`get()` (*Device* method), 368
`get()` (*DeviceControlAlert* method), 300
`get()` (*DeviceControlAlert.Note* method), 298
`get()` (*DeviceFacet* method), 375
`get()` (*DeviceFacet.DeviceFacetValue* method), 374
`get()` (*DeviceSummary* method), 128

`get()` (*DeviceSummary.Metrics* method), 127
`get()` (*DeviceSummaryFacet* method), 129
`get()` (*DeviceSummaryFacet.Values* method), 129
`get()` (*Differential* method), 164
`get()` (*Downloads* method), 245
`get()` (*Downloads.FoundItem* method), 244
`get()` (*EnrichedEvent* method), 172
`get()` (*EnrichedEventFacet* method), 175
`get()` (*EnrichedEventFacet.Ranges* method), 174
`get()` (*EnrichedEventFacet.Terms* method), 174
`get()` (*Event* method), 385
`get()` (*EventFacet* method), 387
`get()` (*EventFacet.Ranges* method), 386
`get()` (*EventFacet.Terms* method), 387
`get()` (*Feed* method), 218
`get()` (*FeedModel* method), 220
`get()` (*Grant* method), 404
`get()` (*Grant.Profile* method), 400
`get()` (*GroupedAlert* method), 304
`get()` (*HostBasedFirewallAlert* method), 331
`get()` (*HostBasedFirewallAlert.Note* method), 329
`get()` (*HostBasedFirewallRuleConfig* method), 460
`get()` (*HostBasedFirewallRuleConfig.FirewallRule* method), 456
`get()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup* method), 458
`get()` (*IntrusionDetectionSystemAlert* method), 340
`get()` (*IntrusionDetectionSystemAlert.Note* method), 338
`get()` (*IOC* method), 223
`get()` (*IOC_V2* method), 225
`get()` (*Job* method), 407
`get()` (*LiveResponseMemdump* method), 609
`get()` (*MutableBaseModel* method), 566
`get()` (*NetworkThreatMetadata* method), 424
`get()` (*NewBaseModel* method), 567
`get()` (*Observation* method), 425
`get()` (*ObservationFacet* method), 428
`get()` (*ObservationFacet.Ranges* method), 427
`get()` (*ObservationFacet.Terms* method), 428
`get()` (*PlatformModel* method), 245
`get()` (*Policy* method), 442
`get()` (*PolicyRule* method), 450
`get()` (*PolicyRuleConfig* method), 463
`get()` (*Process* method), 474
`get()` (*Process.Summary* method), 472
`get()` (*Process.Tree* method), 472
`get()` (*ProcessFacet* method), 477
`get()` (*ProcessFacet.Ranges* method), 476
`get()` (*ProcessFacet.Terms* method), 477
`get()` (*Recommendation* method), 184
`get()` (*Recommendation.RecommendationApplication* method), 181
`get()` (*Recommendation.RecommendationImpact* method), 182

- get() (*Recommendation.RecommendationNewRule method*), 183
- get() (*Recommendation.RecommendationWorkflow method*), 184
- get() (*Report method*), 230
- get() (*ReportSeverity method*), 234
- get() (*ReputationOverride method*), 481
- get() (*Result method*), 136
- get() (*Result.Device method*), 135
- get() (*Result.Fields method*), 135
- get() (*Result.Metrics method*), 136
- get() (*ResultFacet method*), 138
- get() (*ResultFacet.Values method*), 138
- get() (*Run method*), 146
- get() (*RunHistory method*), 148
- get() (*SensorKit method*), 517
- get() (*Template method*), 157
- get() (*TemplateHistory method*), 159
- get() (*UnrefreshableModel method*), 578
- get() (*USBDevice method*), 190
- get() (*USBDeviceApproval method*), 193
- get() (*USBDeviceBlock method*), 198
- get() (*User method*), 488
- get() (*VCenterComputeResource method*), 539
- get() (*Vulnerability method*), 500
- get() (*Vulnerability.OrgSummary method*), 499
- get() (*Watchlist method*), 239
- get() (*WatchlistAlert method*), 350
- get() (*WatchlistAlert.Note method*), 348
- get_affected_assets() (*Vulnerability method*), 500
- get_alert_search_query() (*GroupedAlert method*), 304
- get_alert_search_query() (*GroupedAlertSearch-Query method*), 307
- get_alerts() (*GroupedAlert method*), 304
- get_all_groups() (*AssetGroup class method*), 355
- get_asset_group_ids() (*Device method*), 369
- get_asset_groups() (*Device method*), 369
- get_asset_groups_for_devices() (*Device class method*), 369
- get_assignment_mode() (*CorePreventionRuleConfig method*), 452
- get_auditlogs() (*AuditLog static method*), 365
- get_auditlogs() (*CBCloudAPI method*), 123
- get_auth_events_descriptions() (*AuthEvent static method*), 205
- get_cb_cloud_object() (*in module cbc_sdk.helpers*), 596
- get_config_template() (*SensorKit class method*), 517
- get_connection() (*CBBCSDKSessionAdapter method*), 585
- get_credentials() (*AWSCredentialProvider method*), 169
- get_credentials() (*CredentialProvider method*), 589
- get_credentials() (*EnvironCredentialProvider method*), 169
- get_credentials() (*FileCredentialProvider method*), 170
- get_credentials() (*KeychainCredentialProvider method*), 171
- get_credentials() (*RegistryCredentialProvider method*), 171
- get_default_provider() (*DefaultProvider method*), 168
- get_details() (*AuthEvent method*), 206
- get_details() (*EnrichedEvent method*), 173
- get_details() (*Observation method*), 425
- get_details() (*Process method*), 474
- get_endpoints() (*USBDevice method*), 190
- get_events() (*CBAnalyticsAlert method*), 285
- get_file() (*CbLRSessionBase method*), 600
- get_file() (*LiveResponseSession method*), 612
- get_group_results() (*ObservationQuery method*), 431
- get_history() (*Alert method*), 254
- get_history() (*CBAnalyticsAlert method*), 285
- get_history() (*ContainerRuntimeAlert method*), 293
- get_history() (*DeviceControlAlert method*), 300
- get_history() (*HostBasedFirewallAlert method*), 332
- get_history() (*IntrusionDetectionSystemAlert method*), 341
- get_history() (*WatchlistAlert method*), 350
- get_network_threat_metadata() (*IntrusionDetectionSystemAlert method*), 341
- get_network_threat_metadata() (*Observation method*), 426
- get_notifications() (*CBCloudAPI method*), 123
- get_object() (*BaseAPI method*), 581
- get_object() (*CBCloudAPI method*), 123
- get_object_by_name_or_id() (*in module cbc_sdk.helpers*), 596
- get_observations() (*Alert method*), 254
- get_observations() (*CBAnalyticsAlert method*), 285
- get_observations() (*ContainerRuntimeAlert method*), 293
- get_observations() (*DeviceControlAlert method*), 300
- get_observations() (*HostBasedFirewallAlert method*), 332
- get_observations() (*IntrusionDetectionSystemAlert method*), 341
- get_observations() (*WatchlistAlert method*), 350
- get_output_as_file() (*Job method*), 407
- get_output_as_lines() (*Job method*), 408
- get_output_as_stream() (*Job method*), 408
- get_output_as_string() (*Job method*), 408

- `get_parameter()` (*CorePreventionRuleConfig* method), 452
 - `get_parameter()` (*DataCollectionRuleConfig* method), 454
 - `get_parameter()` (*HostBasedFirewallRuleConfig* method), 460
 - `get_parameter()` (*PolicyRuleConfig* method), 463
 - `get_permitted_role_urns()` (*Grant* class method), 404
 - `get_policy_ruleconfig_parameter_schema()` (*CBCloudAPI* method), 124
 - `get_process()` (*Alert* method), 254
 - `get_process()` (*CBAntivirusAlert* method), 286
 - `get_process()` (*ContainerRuntimeAlert* method), 293
 - `get_process()` (*DeviceControlAlert* method), 301
 - `get_process()` (*HostBasedFirewallAlert* method), 332
 - `get_process()` (*IntrusionDetectionSystemAlert* method), 342
 - `get_process()` (*WatchlistAlert* method), 351
 - `get_progress()` (*Job* method), 408
 - `get_raw_data()` (*BaseAPI* method), 582
 - `get_raw_data()` (*CBCloudAPI* method), 124
 - `get_raw_file()` (*CbLRSessionBase* method), 600
 - `get_raw_file()` (*LiveResponseSession* method), 612
 - `get_registry_value()` (*CbLRSessionBase* method), 600
 - `get_registry_value()` (*LiveResponseSession* method), 612
 - `get_ruleconfig_parameter_schema()` (*Policy* method), 443
 - `get_statistics()` (*AssetGroup* method), 355
 - `get_threat_tags()` (*Alert* method), 255
 - `get_threat_tags()` (*CBAntivirusAlert* method), 286
 - `get_threat_tags()` (*ContainerRuntimeAlert* method), 294
 - `get_threat_tags()` (*DeviceControlAlert* method), 301
 - `get_threat_tags()` (*HostBasedFirewallAlert* method), 332
 - `get_threat_tags()` (*IntrusionDetectionSystemAlert* method), 342
 - `get_threat_tags()` (*WatchlistAlert* method), 351
 - `get_token()` (*Credentials* method), 590
 - `get_token_type()` (*Credentials* method), 590
 - `get_value()` (*Credentials* method), 590
 - `get_vendors_and_products_seen()` (*USBDevice* class method), 190
 - `get_vulnerability_summary()` (*Device* method), 370
 - `get_vulnerabilities()` (*Device* method), 370
 - `get_watchlist_objects()` (*WatchlistAlert* method), 351
 - `GetFileJob` (class in *cbc_sdk.live_response_api*), 605
 - `getName()` (*JobWorker* method), 606
 - `getName()` (*LiveResponseJobScheduler* method), 608
 - `getName()` (*LRUCacheDict.EmptyCacheThread* method), 554
 - `GetScore()` (in module *cbc_sdk.winerror*), 621
 - `Grant` (class in *cbc_sdk.platform.grants*), 398
 - `grant()` (*User* method), 488
 - `Grant.GrantBuilder` (class in *cbc_sdk.platform.grants*), 399
 - `Grant.Profile` (class in *cbc_sdk.platform.grants*), 400
 - `Grant.ProfileBuilder` (class in *cbc_sdk.platform.grants*), 401
 - `GrantQuery` (class in *cbc_sdk.platform.grants*), 405
 - `group_results()` (*AuthEventQuery* method), 211
 - `GroupedAlert` (class in *cbc_sdk.platform.alerts*), 303
 - `GroupedAlertSearchQuery` (class in *cbc_sdk.platform.alerts*), 305
- ## H
- `host_based_firewall_rule_config` (*Policy* property), 443
 - `HostBasedFirewallAlert` (class in *cbc_sdk.platform.alerts*), 325
 - `HostBasedFirewallAlert.Note` (class in *cbc_sdk.platform.alerts*), 329
 - `HostBasedFirewallRuleConfig` (class in *cbc_sdk.platform.policy_ruleconfigs*), 455
 - `HostBasedFirewallRuleConfig.FirewallRule` (class in *cbc_sdk.platform.policy_ruleconfigs*), 456
 - `HostBasedFirewallRuleConfig.FirewallRuleGroup` (class in *cbc_sdk.platform.policy_ruleconfigs*), 457
 - `HRESULT_CODE()` (in module *cbc_sdk.winerror*), 621
 - `HRESULT_FACILITY()` (in module *cbc_sdk.winerror*), 621
 - `HRESULT_FROM_NT()` (in module *cbc_sdk.winerror*), 621
 - `HRESULT_FROM_WIN32()` (in module *cbc_sdk.winerror*), 621
 - `HRESULT_SEVERITY()` (in module *cbc_sdk.winerror*), 621
 - `http_request()` (*Connection* method), 587
- ## I
- `ident` (*JobWorker* property), 606
 - `ident` (*LiveResponseJobScheduler* property), 608
 - `ident` (*LRUCacheDict.EmptyCacheThread* property), 554
 - `ignore()` (*IOC_V2* method), 226
 - `ignore()` (*Report* method), 230
 - `ignored` (*IOC_V2* property), 226
 - `ignored` (*Report* property), 230
 - `impact_` (*Recommendation* property), 184
 - `index()` (*Vulnerability.AssetView* method), 498
 - `init_poolmanager()` (*CBSCSDKSessionAdapter* method), 585

- [insert\(\) \(Vulnerability.AssetView method\), 498](#)
[install_sensor\(\) \(AWSComputeResource method\), 521](#)
[install_sensor\(\) \(BaseComputeResource method\), 532](#)
[install_sensor\(\) \(VCenterComputeResource method\), 539](#)
[IntrusionDetectionSystemAlert \(class in cbc_sdk.platform.alerts\), 334](#)
[IntrusionDetectionSystemAlert.Note \(class in cbc_sdk.platform.alerts\), 338](#)
[InvalidHashError, 592](#)
[InvalidObjectError, 592, 636](#)
[IOC \(class in cbc_sdk.enterprise_edr.threat_intelligence\), 222](#)
[IOC_V2 \(class in cbc_sdk.enterprise_edr.threat_intelligence\), 224](#)
[iocs_ \(Report property\), 231](#)
[ipv6_equality_format\(\) \(IOC_V2 class method\), 226](#)
[is_alive\(\) \(JobWorker method\), 606](#)
[is_alive\(\) \(LiveResponseJobScheduler method\), 608](#)
[is_alive\(\) \(LRUCacheDict.EmptyCacheThread method\), 554](#)
[is_deleted \(PolicyRule property\), 450](#)
[is_dirty\(\) \(AssetGroup method\), 356](#)
[is_dirty\(\) \(CorePreventionRuleConfig method\), 452](#)
[is_dirty\(\) \(DataCollectionRuleConfig method\), 454](#)
[is_dirty\(\) \(Feed method\), 218](#)
[is_dirty\(\) \(FeedModel method\), 220](#)
[is_dirty\(\) \(Grant method\), 404](#)
[is_dirty\(\) \(Grant.Profile method\), 400](#)
[is_dirty\(\) \(HostBasedFirewallRuleConfig method\), 461](#)
[is_dirty\(\) \(HostBasedFirewallRuleConfig.FirewallRule method\), 457](#)
[is_dirty\(\) \(HostBasedFirewallRuleConfig.FirewallRuleGroup method\), 458](#)
[is_dirty\(\) \(IOC method\), 223](#)
[is_dirty\(\) \(IOC_V2 method\), 226](#)
[is_dirty\(\) \(MutableBaseModel method\), 566](#)
[is_dirty\(\) \(Policy method\), 443](#)
[is_dirty\(\) \(PolicyRule method\), 450](#)
[is_dirty\(\) \(PolicyRuleConfig method\), 463](#)
[is_dirty\(\) \(Report method\), 231](#)
[is_dirty\(\) \(ReportSeverity method\), 235](#)
[is_dirty\(\) \(USBDeviceApproval method\), 193](#)
[is_dirty\(\) \(User method\), 488](#)
[is_dirty\(\) \(Watchlist method\), 239](#)
[isDaemon\(\) \(JobWorker method\), 606](#)
[isDaemon\(\) \(LiveResponseJobScheduler method\), 608](#)
[isDaemon\(\) \(LRUCacheDict.EmptyCacheThread method\), 554](#)
[IsoDateTimeFieldDescriptor \(class in cbc_sdk.base\), 565](#)
[IterableQueryMixin \(class in cbc_sdk.base\), 565](#)
- ## J
- [Job \(class in cbc_sdk.platform.jobs\), 406](#)
[JobQuery \(class in cbc_sdk.platform.jobs\), 409](#)
[jobrunner\(\) \(in module cbc_sdk.live_response_api\), 618](#)
[JobWorker \(class in cbc_sdk.live_response_api\), 606](#)
[join\(\) \(JobWorker method\), 606](#)
[join\(\) \(LiveResponseJobScheduler method\), 608](#)
[join\(\) \(LRUCacheDict.EmptyCacheThread method\), 554](#)
- ## K
- [KeychainCredentialProvider \(class in cbc_sdk.credential_providers.keychain_credential_provider\), 170](#)
[kill_process\(\) \(CbLRSessionBase method\), 601](#)
[kill_process\(\) \(LiveResponseSession method\), 613](#)
- ## L
- [latestRevision \(Policy property\), 443](#)
[LegacyAlertSearchQueryCriterionMixin \(class in cbc_sdk.platform.legacy_alerts\), 410](#)
[limit\(\) \(EventFacetQuery method\), 390](#)
[limit\(\) \(FacetQuery method\), 561](#)
[list_directory\(\) \(CbLRSessionBase method\), 601](#)
[list_directory\(\) \(LiveResponseSession method\), 613](#)
[list_member_ids\(\) \(AssetGroup method\), 356](#)
[list_members\(\) \(AssetGroup method\), 356](#)
[list_processes\(\) \(CbLRSessionBase method\), 602](#)
[list_processes\(\) \(LiveResponseSession method\), 614](#)
[list_registry_keys_and_values\(\) \(CbLRSessionBase method\), 602](#)
[list_registry_keys_and_values\(\) \(LiveResponseSession method\), 614](#)
[list_registry_values\(\) \(CbLRSessionBase method\), 603](#)
[list_registry_values\(\) \(LiveResponseSession method\), 615](#)
[live_response \(CBCloudAPI property\), 124](#)
[LiveResponseError, 607](#)
[LiveResponseJobScheduler \(class in cbc_sdk.live_response_api\), 608](#)
[LiveResponseMemdump \(class in cbc_sdk.live_response_api\), 609](#)
[LiveResponseSession \(class in cbc_sdk.live_response_api\), 610](#)
[LiveResponseSessionManager \(class in cbc_sdk.live_response_api\), 617](#)
[log \(in module cbc_sdk.base\), 579](#)
[log \(in module cbc_sdk.endpoint_standard.base\), 180](#)
[log \(in module cbc_sdk.endpoint_standard.recommendation\), 188](#)

[log \(in module `cbc_sdk.endpoint_standard.usb_device_control`\)](#), [cbc_sdk.credentials](#), 589
[204](#)
[log \(in module `cbc_sdk.enterprise_edr.threat_intelligence`\)](#), [cbc_sdk.endpoint_standard.base](#), 172
[242](#)
[log \(in module `cbc_sdk.platform.base`\)](#), 246
[log \(in module `cbc_sdk.platform.devices`\)](#), 384
[log \(in module `cbc_sdk.platform.grants`\)](#), 406
[log \(in module `cbc_sdk.platform.users`\)](#), 491
[log \(in module `cbc_sdk.platform.vulnerability_assessment`\)](#), [cbc_sdk.endpoint_standard.recommendation](#),
[514](#)
[180](#)
[log \(in module `cbc_sdk.workload.vm_workloads_search`\)](#), [cbc_sdk.endpoint_standard.usb_device_control](#),
[552](#)
[189](#)
[lookup_error\(\) \(CommDlgError class method\)](#), 620
[lookup_error\(\) \(DirectoryStorageError class method\)](#), [cbc_sdk.enterprise_edr.auth_events](#), 204
[620](#)
[lookup_error\(\) \(ErrorBaseClass class method\)](#), 620
[lookup_error\(\) \(Facility class method\)](#), 621
[lookup_error\(\) \(RawErrorCode class method\)](#), 621
[lookup_error\(\) \(Win32Error class method\)](#), 622
[lr_session\(\) \(Device method\)](#), 370
[lru_cache_function\(\) \(in module `cbc_sdk.cache.lru`\)](#), [cbc_sdk.enterprise_edr.threat_intelligence](#),
[556](#)
[216](#)
[LRUCacheFunction \(class in `cbc_sdk.cache.lru`\)](#), 555
[LRUCacheDict \(class in `cbc_sdk.cache.lru`\)](#), 553
[LRUCacheDict.EmptyCacheThread \(class in `cbc_sdk.cache.lru`\)](#), 553

M

[matches_template\(\) \(Grant.Profile method\)](#), 401
[MAX_RESULTS_LIMIT \(in module `cbc_sdk.audit_remediation.base`\)](#), 134
[memdump\(\) \(CbLRSessionBase method\)](#), 603
[memdump\(\) \(LiveResponseSession method\)](#), 615
[metrics_ \(DeviceSummary property\)](#), 128
[metrics_ \(Result property\)](#), 136
[ModelNotFound](#), 593
[module](#)
[\[cbc_sdk.audit_remediation.base\]\(#\)](#), 127
[\[cbc_sdk.audit_remediation.differential\]\(#\)](#),
[163](#)
[\[cbc_sdk.base\]\(#\)](#), 556
[\[cbc_sdk.cache.lru\]\(#\)](#), 553
[\[cbc_sdk.connection\]\(#\)](#), 579
[\[cbc_sdk.credential_providers.aws_sm_credential_provider\]\(#\)](#),
[169](#)
[\[cbc_sdk.credential_providers.default\]\(#\)](#), 168
[\[cbc_sdk.credential_providers.envron_credential_provider\]\(#\)](#),
[169](#)
[\[cbc_sdk.credential_providers.file_credential_provider\]\(#\)](#),
[170](#)
[\[cbc_sdk.credential_providers.keychain_credential_provider\]\(#\)](#),
[170](#)
[\[cbc_sdk.credential_providers.registry_credential_provider\]\(#\)](#),
[171](#)

[cbc_sdk.endpoint_standard.base](#), 172
[cbc_sdk.endpoint_standard.recommendation](#),
[180](#)
[cbc_sdk.endpoint_standard.usb_device_control](#),
[189](#)
[cbc_sdk.enterprise_edr.auth_events](#), 204
[cbc_sdk.enterprise_edr.threat_intelligence](#),
[216](#)
[cbc_sdk.enterprise_edr.ubs](#), 242
[cbc_sdk.errors](#), 591
[cbc_sdk.helpers](#), 596
[cbc_sdk.live_response_api](#), 597
[cbc_sdk.platform.alerts](#), 246
[cbc_sdk.platform.asset_groups](#), 353
[cbc_sdk.platform.audit](#), 364
[cbc_sdk.platform.base](#), 245
[cbc_sdk.platform.devices](#), 365
[cbc_sdk.platform.events](#), 385
[cbc_sdk.platform.grants](#), 398
[cbc_sdk.platform.jobs](#), 406
[cbc_sdk.platform.legacy_alerts](#), 410
[cbc_sdk.platform.network_threat_metadata](#),
[423](#)
[cbc_sdk.platform.observations](#), 424
[cbc_sdk.platform.policies](#), 435
[cbc_sdk.platform.policy_ruleconfigs](#), 451
[cbc_sdk.platform.previewer](#), 464
[cbc_sdk.platform.processes](#), 465
[cbc_sdk.platform.reputation](#), 480
[cbc_sdk.platform.users](#), 484
[cbc_sdk.platform.vulnerability_assessment](#),
[491](#)
[cbc_sdk.utils](#), 619
[cbc_sdk.winerror](#), 620
[cbc_sdk.workload.nsx_remediation](#), 515
[cbc_sdk.workload.sensor_lifecycle](#), 516
[cbc_sdk.workload.vm_workloads_search](#), 520
[MoreThanOneResultError](#), 593, 636
[most_recent_alert_ \(GroupedAlert property\)](#), 304
[mro\(\) \(CbMetaModel method\)](#), 557
[mro\(\) \(ErrorMetaClass method\)](#), 621
[MutableBaseModel \(class in `cbc_sdk.base`\)](#), 565

N

[name \(JobWorker property\)](#), 607
[name \(LiveResponseJobScheduler property\)](#), 608
[name \(LRUCacheDict.EmptyCacheThread property\)](#), 554
[name \(Query method\)](#), 153
[native_id \(JobWorker property\)](#), 607
[native_id \(LiveResponseJobScheduler property\)](#), 609
[native_id \(LRUCacheDict.EmptyCacheThread prop-](#)
[erty\)](#), 554

NetworkThreatMetadata (class in *cbc_sdk.platform.network_threat_metadata*), 423
 new_policy (DevicePolicyChangePreview property), 465
 new_policy_id (DevicePolicyChangePreview property), 465
 new_policy_position (DevicePolicyChangePreview property), 465
 new_rule_ (Recommendation property), 185
 NewBaseModel (class in *cbc_sdk.base*), 567
 newer_run_id() (DifferentialQuery method), 166
 NonQueryableModel, 593
 normalize_org() (in module *cbc_sdk.platform.grants*), 406
 normalize_profile_list() (in module *cbc_sdk.platform.users*), 491
 not_() (AffectedAssetQuery method), 492
 not_() (AlertSearchQuery method), 260
 not_() (AssetGroupQuery method), 362
 not_() (AsyncProcessQuery method), 467
 not_() (AuthEventQuery method), 212
 not_() (AWSComputeResourceQuery method), 526
 not_() (BaseComputeResourceQuery method), 535
 not_() (DeviceSearchQuery method), 379
 not_() (EnrichedEventQuery method), 177
 not_() (EventFacetQuery method), 390
 not_() (EventQuery method), 395
 not_() (FacetQuery method), 131, 562
 not_() (GroupedAlertSearchQuery method), 307
 not_() (ObservationQuery method), 431
 not_() (Query method), 571
 not_() (QueryBuilder method), 574
 not_() (QueryBuilderSupportMixin method), 575
 not_() (ReputationOverrideQuery method), 482
 not_() (ResultQuery method), 141
 not_() (RunHistoryQuery method), 151
 not_() (SummaryQuery method), 478
 not_() (TemplateHistoryQuery method), 161
 not_() (USBDeviceApprovalQuery method), 195
 not_() (USBDeviceQuery method), 201
 not_() (VCenterComputeResourceQuery method), 546
 not_() (VulnerabilityAssetViewQuery method), 502
 not_() (VulnerabilityQuery method), 509
 notes_() (Alert method), 255
 notes_() (CBAnalyticsAlert method), 286
 notes_() (ContainerRuntimeAlert method), 294
 notes_() (DeviceControlAlert method), 301
 notes_() (HostBasedFirewallAlert method), 332
 notes_() (IntrusionDetectionSystemAlert method), 342
 notes_() (WatchlistAlert method), 351
 notification_listener() (CBCloudAPI method), 124
 notify_on_finish() (RunQuery method), 153
 nsx_available (Device property), 370
 nsx_remediation() (Device method), 370
 NSXJobError, 593
 NSXRemediationJob (class in *cbc_sdk.workload.nsx_remediation*), 515

O

object_rule_configs (Policy property), 443
 object_rule_configs_list (Policy property), 443
 object_rules (Policy property), 444
 ObjectFieldDescriptor (class in *cbc_sdk.base*), 568
 ObjectNotFoundError, 594, 636
 Observation (class in *cbc_sdk.platform.observations*), 424
 ObservationFacet (class in *cbc_sdk.platform.observations*), 427
 ObservationFacet.Ranges (class in *cbc_sdk.platform.observations*), 427
 ObservationFacet.Terms (class in *cbc_sdk.platform.observations*), 428
 ObservationGroup (class in *cbc_sdk.platform.observations*), 429
 ObservationQuery (class in *cbc_sdk.platform.observations*), 429
 older_run_id() (DifferentialQuery method), 167
 one() (AffectedAssetQuery method), 493
 one() (AlertSearchQuery method), 260
 one() (AssetGroupQuery method), 362
 one() (AsyncProcessQuery method), 468
 one() (AuthEventQuery method), 212
 one() (AWSComputeResourceQuery method), 527
 one() (BaseComputeResourceQuery method), 535
 one() (DeviceSearchQuery method), 379
 one() (DifferentialQuery method), 167
 one() (EnrichedEventQuery method), 177
 one() (EventQuery method), 395
 one() (FacetQuery method), 131
 one() (FeedQuery method), 222
 one() (GrantQuery method), 406
 one() (GroupedAlertSearchQuery method), 307
 one() (IterableQueryMixin method), 565
 one() (JobQuery method), 409
 one() (ObservationQuery method), 431
 one() (PaginatedQuery method), 568
 one() (PolicyQuery method), 448
 one() (Query method), 571
 one() (RecommendationQuery method), 187
 one() (ReportQuery method), 233
 one() (ReputationOverrideQuery method), 483
 one() (ResultQuery method), 141
 one() (RunHistoryQuery method), 151
 one() (SensorKitQuery method), 519
 one() (SimpleQuery method), 577
 one() (TemplateHistoryQuery method), 162

- `one()` (*USBDeviceApprovalQuery* method), 195
- `one()` (*USBDeviceBlockQuery* method), 199
- `one()` (*USBDeviceQuery* method), 201
- `one()` (*UserQuery* method), 490
- `one()` (*VCenterComputeResourceQuery* method), 546
- `one()` (*VulnerabilityAssetViewQuery* method), 502
- `one()` (*VulnerabilityQuery* method), 510
- `one()` (*WatchlistQuery* method), 241
- `OpenKey()` (in module `cbc_sdk.credential_providers.registry_credential_providers`), 171
- `OperationCancelled`, 594
- `or_()` (*AffectedAssetQuery* method), 493
- `or_()` (*AlertSearchQuery* method), 260
- `or_()` (*AssetGroupQuery* method), 363
- `or_()` (*AsyncProcessQuery* method), 468
- `or_()` (*AuthEventQuery* method), 212
- `or_()` (*AWSComputeResourceQuery* method), 527
- `or_()` (*BaseComputeResourceQuery* method), 535
- `or_()` (*DeviceSearchQuery* method), 379
- `or_()` (*EnrichedEventQuery* method), 177
- `or_()` (*EventFacetQuery* method), 391
- `or_()` (*EventQuery* method), 395
- `or_()` (*FacetQuery* method), 132, 562
- `or_()` (*GroupedAlertSearchQuery* method), 308
- `or_()` (*ObservationQuery* method), 432
- `or_()` (*Query* method), 571
- `or_()` (*QueryBuilder* method), 575
- `or_()` (*QueryBuilderSupportMixin* method), 576
- `or_()` (*ReputationOverrideQuery* method), 483
- `or_()` (*ResultQuery* method), 141
- `or_()` (*RunHistoryQuery* method), 151
- `or_()` (*SummaryQuery* method), 478
- `or_()` (*TemplateHistoryQuery* method), 162
- `or_()` (*USBDeviceApprovalQuery* method), 195
- `or_()` (*USBDeviceQuery* method), 201
- `or_()` (*VCenterComputeResourceQuery* method), 547
- `or_()` (*VulnerabilityAssetViewQuery* method), 502
- `or_()` (*VulnerabilityQuery* method), 510
- `org_urn` (*CBCloudAPI* property), 125
- `org_urn` (*User* property), 488
- P**
- `PaginatedQuery` (class in `cbc_sdk.base`), 568
- `parameter_names` (*CorePreventionRuleConfig* property), 452
- `parameter_names` (*DataCollectionRuleConfig* property), 454
- `parameter_names` (*HostBasedFirewallRuleConfig* property), 461
- `parameter_names` (*PolicyRuleConfig* property), 463
- `parents` (*Process* property), 474
- `perform_action()` (*Vulnerability* method), 500
- `PlatformModel` (class in `cbc_sdk.platform.base`), 245
- `Policy` (class in `cbc_sdk.platform.policies`), 435
- `policy` (*Policy* property), 444
- `Policy.PolicyBuilder` (class in `cbc_sdk.platform.policies`), 436
- `policy_id()` (*RunQuery* method), 154
- `PolicyQuery` (class in `cbc_sdk.platform.policies`), 447
- `PolicyRule` (class in `cbc_sdk.platform.policies`), 449
- `PolicyRuleConfig` (class in `cbc_sdk.platform.policy_ruleconfigs`), 462
- `policy_status()` (in module `cbc_sdk.live_response_api`), 619
- `pop()` (*Vulnerability.AssetView* method), 499
- `post()` (*Connection* method), 588
- `post_multipart()` (*BaseAPI* method), 582
- `post_multipart()` (*CBCloudAPI* method), 125
- `post_object()` (*BaseAPI* method), 583
- `post_object()` (*CBCloudAPI* method), 125
- `preview_add_members()` (*AssetGroup* method), 357
- `preview_add_members_to_groups()` (*AssetGroup* class method), 357
- `preview_add_policy_override()` (*Policy* method), 444
- `preview_add_policy_override_for_devices()` (*Device* class method), 371
- `preview_create_asset_group()` (*AssetGroup* class method), 357
- `preview_delete()` (*AssetGroup* method), 358
- `preview_delete_asset_groups()` (*AssetGroup* class method), 358
- `preview_policy_rank_changes()` (*Policy* class method), 444
- `preview_rank_change()` (*Policy* method), 445
- `preview_remove_members()` (*AssetGroup* method), 358
- `preview_remove_members_from_groups()` (*AssetGroup* class method), 359
- `preview_remove_policy_override()` (*Device* method), 371
- `preview_remove_policy_override_for_devices()` (*Device* class method), 371
- `preview_save()` (*AssetGroup* method), 359
- `preview_update_asset_groups()` (*AssetGroup* class method), 359
- `priorityLevel` (*Policy* property), 445
- `Process` (class in `cbc_sdk.platform.processes`), 471
- `Process.Summary` (class in `cbc_sdk.platform.processes`), 471
- `Process.Tree` (class in `cbc_sdk.platform.processes`), 472
- `process_limits()` (*CBCloudAPI* method), 125
- `process_md5` (*Process* property), 474
- `process_pids` (*Process* property), 474
- `process_sha256` (*EnrichedEvent* property), 173
- `process_sha256` (*Process* property), 474

- ProcessFacet (class in *cbc_sdk.platform.processes*), 475
- ProcessFacet.Ranges (class in *cbc_sdk.platform.processes*), 476
- ProcessFacet.Terms (class in *cbc_sdk.platform.processes*), 476
- profiles_ (Grant property), 404
- proxy_headers() (CBCSDKSessionAdapter method), 585
- proxy_manager_for() (CBCSDKSessionAdapter method), 585
- put() (Connection method), 588
- put_file() (CbLRSessionBase method), 604
- put_file() (LiveResponseSession method), 616
- put_object() (BaseAPI method), 583
- put_object() (CBCloudAPI method), 126
- ## Q
- quarantine() (Device method), 372
- quarantine() (DeviceSearchQuery method), 379
- Query (class in *cbc_sdk.base*), 569
- query_device_summaries() (Result method), 136
- query_device_summaries() (Run method), 147
- query_device_summaries() (RunHistory method), 148
- query_device_summaries() (Template method), 157
- query_device_summaries() (TemplateHistory method), 159
- query_device_summary_facets() (Result method), 137
- query_devices() (DeviceFacet.DeviceFacetValue method), 374
- query_facets() (Run method), 147
- query_facets() (RunHistory method), 148
- query_facets() (Template method), 157
- query_facets() (TemplateHistory method), 159
- query_result_facets() (Result method), 137
- query_results() (Run method), 147
- query_results() (RunHistory method), 149
- query_results() (Template method), 158
- query_results() (TemplateHistory method), 160
- query_runs() (Template method), 158
- query_runs() (TemplateHistory method), 160
- QueryBuilder (class in *cbc_sdk.base*), 574
- QueryBuilderSupportMixin (class in *cbc_sdk.base*), 575
- QuerySyntaxError, 594
- QueryValueEx() (in module *cbc_sdk.credential_providers.registry_credential_provider*), 171
- ranges_ (EventFacet property), 387
- ranges_ (ObservationFacet property), 428
- ranges_ (ProcessFacet property), 477
- RawErrorCode (class in *cbc_sdk.winerror*), 621
- read_iocs() (in module *cbc_sdk.helpers*), 597
- Recommendation (class in *cbc_sdk.endpoint_standard.recommendation*), 180
- Recommendation.RecommendationApplication (class in *cbc_sdk.endpoint_standard.recommendation*), 181
- Recommendation.RecommendationImpact (class in *cbc_sdk.endpoint_standard.recommendation*), 181
- Recommendation.RecommendationNewRule (class in *cbc_sdk.endpoint_standard.recommendation*), 182
- Recommendation.RecommendationWorkflow (class in *cbc_sdk.endpoint_standard.recommendation*), 183
- RecommendationQuery (class in *cbc_sdk.endpoint_standard.recommendation*), 186
- refresh() (Alert method), 255
- refresh() (Alert.Note method), 252
- refresh() (AssetGroup method), 360
- refresh() (AuditLog method), 365
- refresh() (AuthEvent method), 206
- refresh() (AuthEventFacet method), 209
- refresh() (AuthEventFacet.Ranges method), 208
- refresh() (AuthEventFacet.Terms method), 208
- refresh() (AWSComputeResource method), 521
- refresh() (BaseComputeResource method), 532
- refresh() (Binary method), 243
- refresh() (Binary.Summary method), 243
- refresh() (CBAnalyticsAlert method), 286
- refresh() (CBAnalyticsAlert.Note method), 282
- refresh() (ComputeResourceFacet method), 538
- refresh() (ComputeResourceFacet.ComputeResourceFacetValue method), 537
- refresh() (ContainerRuntimeAlert method), 294
- refresh() (ContainerRuntimeAlert.Note method), 291
- refresh() (CorePreventionRuleConfig method), 452
- refresh() (DataCollectionRuleConfig method), 455
- refresh() (Device method), 372
- refresh() (DeviceControlAlert method), 301
- refresh() (DeviceControlAlert.Note method), 298
- refresh() (DeviceFacet method), 375
- refresh() (DeviceFacet.DeviceFacetValue method), 375
- refresh() (DeviceSummary method), 128
- refresh() (DeviceSummary.Metrics method), 128
- refresh() (DeviceSummaryFacet method), 129
- ## R
- ranges_ (AuthEventFacet property), 209
- ranges_ (EnrichedEventFacet property), 175

- `refresh()` (*DeviceSummaryFacet.Values* method), 129
- `refresh()` (*Differential* method), 164
- `refresh()` (*Downloads* method), 245
- `refresh()` (*Downloads.FoundItem* method), 244
- `refresh()` (*EnrichedEvent* method), 173
- `refresh()` (*EnrichedEventFacet* method), 175
- `refresh()` (*EnrichedEventFacet.Ranges* method), 174
- `refresh()` (*EnrichedEventFacet.Terms* method), 174
- `refresh()` (*Event* method), 385
- `refresh()` (*EventFacet* method), 388
- `refresh()` (*EventFacet.Ranges* method), 386
- `refresh()` (*EventFacet.Terms* method), 387
- `refresh()` (*Feed* method), 218
- `refresh()` (*FeedModel* method), 220
- `refresh()` (*Grant* method), 404
- `refresh()` (*Grant.Profile* method), 401
- `refresh()` (*GroupedAlert* method), 304
- `refresh()` (*HostBasedFirewallAlert* method), 333
- `refresh()` (*HostBasedFirewallAlert.Note* method), 329
- `refresh()` (*HostBasedFirewallRuleConfig* method), 461
- `refresh()` (*HostBasedFirewallRuleConfig.FirewallRule* method), 457
- `refresh()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup* method), 458
- `refresh()` (*IntrusionDetectionSystemAlert* method), 342
- `refresh()` (*IntrusionDetectionSystemAlert.Note* method), 338
- `refresh()` (*IOC* method), 223
- `refresh()` (*IOC_V2* method), 226
- `refresh()` (*Job* method), 408
- `refresh()` (*MutableBaseModel* method), 566
- `refresh()` (*NetworkThreatMetadata* method), 424
- `refresh()` (*NewBaseModel* method), 567
- `refresh()` (*Observation* method), 426
- `refresh()` (*ObservationFacet* method), 429
- `refresh()` (*ObservationFacet.Ranges* method), 427
- `refresh()` (*ObservationFacet.Terms* method), 428
- `refresh()` (*PlatformModel* method), 246
- `refresh()` (*Policy* method), 445
- `refresh()` (*PolicyRule* method), 450
- `refresh()` (*PolicyRuleConfig* method), 464
- `refresh()` (*Process* method), 475
- `refresh()` (*Process.Summary* method), 472
- `refresh()` (*Process.Tree* method), 473
- `refresh()` (*ProcessFacet* method), 477
- `refresh()` (*ProcessFacet.Ranges* method), 476
- `refresh()` (*ProcessFacet.Terms* method), 477
- `refresh()` (*Recommendation* method), 185
- `refresh()` (*Recommendation.RecommendationApplication* method), 181
- `refresh()` (*Recommendation.RecommendationImpact* method), 182
- `refresh()` (*Recommendation.RecommendationNewRule* method), 183
- `refresh()` (*Recommendation.RecommendationWorkflow* method), 184
- `refresh()` (*Report* method), 231
- `refresh()` (*ReportSeverity* method), 235
- `refresh()` (*ReputationOverride* method), 481
- `refresh()` (*Result* method), 137
- `refresh()` (*Result.Device* method), 135
- `refresh()` (*Result.Fields* method), 135
- `refresh()` (*Result.Metrics* method), 136
- `refresh()` (*ResultFacet* method), 138
- `refresh()` (*ResultFacet.Values* method), 138
- `refresh()` (*Run* method), 147
- `refresh()` (*RunHistory* method), 149
- `refresh()` (*SensorKit* method), 517
- `refresh()` (*Template* method), 158
- `refresh()` (*TemplateHistory* method), 160
- `refresh()` (*UnrefreshableModel* method), 578
- `refresh()` (*USBDevice* method), 190
- `refresh()` (*USBDeviceApproval* method), 193
- `refresh()` (*USBDeviceBlock* method), 198
- `refresh()` (*User* method), 488
- `refresh()` (*VCenterComputeResource* method), 540
- `refresh()` (*Vulnerability* method), 500
- `refresh()` (*Vulnerability.OrgSummary* method), 499
- `refresh()` (*Watchlist* method), 239
- `refresh()` (*WatchlistAlert* method), 352
- `refresh()` (*WatchlistAlert.Note* method), 348
- `RegistryCredentialProvider` (class in *cbc_sdk.credential_providers.registry_credential_provider*), 171
- `reject()` (*Recommendation* method), 185
- `remove()` (*HostBasedFirewallRuleConfig.FirewallRule* method), 457
- `remove()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup* method), 458
- `remove()` (*Vulnerability.AssetView* method), 499
- `remove_from_groups()` (*Device* method), 372
- `remove_from_groups_by_id()` (*Device* method), 372
- `remove_iocs()` (*Report* method), 231
- `remove_iocs_by_id()` (*Report* method), 231
- `remove_members()` (*AssetGroup* method), 360
- `replace_reports()` (*Feed* method), 219
- `replace_reports_rawdata()` (*Feed* method), 219
- `replace_rule()` (*Policy* method), 445
- `replace_rule_config()` (*Policy* method), 445
- `Report` (class in *cbc_sdk.enterprise_edr.threat_intelligence*), 227
- `Report.ReportBuilder` (class in *cbc_sdk.enterprise_edr.threat_intelligence*), 228
- `ReportQuery` (class in *cbc_sdk.enterprise_edr.threat_intelligence*), 228

`cbc_sdk.enterprise_edr.threat_intelligence)`, 233
`reports` (*Feed property*), 219
`reports` (*Watchlist property*), 239
`ReportSeverity` (class in `cbc_sdk.enterprise_edr.threat_intelligence`), 234
`reputation_override()` (*Recommendation method*), 185
`ReputationOverride` (class in `cbc_sdk.platform.reputation`), 480
`ReputationOverrideQuery` (class in `cbc_sdk.platform.reputation`), 482
`request_session()` (*CbLRManagerBase method*), 597
`request_session()` (*LiveResponseSessionManager method*), 617
`request_url()` (*CBCSDKSessionAdapter method*), 586
`requires_boolean_value()` (*CredentialValue method*), 589
`requires_integer_value()` (*CredentialValue method*), 589
`reset()` (*AssetGroup method*), 360
`reset()` (*CorePreventionRuleConfig method*), 452
`reset()` (*DataCollectionRuleConfig method*), 455
`reset()` (*Feed method*), 219
`reset()` (*FeedModel method*), 221
`reset()` (*Grant method*), 404
`reset()` (*Grant.Profile method*), 401
`reset()` (*HostBasedFirewallRuleConfig method*), 461
`reset()` (*HostBasedFirewallRuleConfig.FirewallRule method*), 457
`reset()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup method*), 458
`reset()` (*IOC method*), 223
`reset()` (*IOC_V2 method*), 226
`reset()` (*MutableBaseModel method*), 566
`reset()` (*Policy method*), 445
`reset()` (*PolicyRule method*), 450
`reset()` (*PolicyRuleConfig method*), 464
`reset()` (*Recommendation method*), 185
`reset()` (*Report method*), 231
`reset()` (*ReportSeverity method*), 235
`reset()` (*USBDeviceApproval method*), 193
`reset()` (*User method*), 488
`reset()` (*Watchlist method*), 239
`reset_google_authenticator_registration()` (*User method*), 488
`Result` (class in `cbc_sdk.audit_remediation.base`), 134
`Result.Device` (class in `cbc_sdk.audit_remediation.base`), 134
`Result.Fields` (class in `cbc_sdk.audit_remediation.base`), 135
`Result.Metrics` (class in `cbc_sdk.audit_remediation.base`), 135
`ResultFacet` (class in `cbc_sdk.audit_remediation.base`), 137
`ResultFacet.Values` (class in `cbc_sdk.audit_remediation.base`), 137
`ResultFromScore()` (in module `cbc_sdk.winerror`), 622
`ResultQuery` (class in `cbc_sdk.audit_remediation.base`), 138
`results` (*EventFacetQuery property*), 391
`results` (*FacetQuery property*), 562
`results` (*FeedQuery property*), 222
`results` (*ReportQuery property*), 234
`results` (*SimpleQuery property*), 577
`results` (*SummaryQuery property*), 479
`results` (*WatchlistQuery property*), 241
`reverse()` (*Vulnerability.AssetView method*), 499
`rule_groups` (*HostBasedFirewallRuleConfig property*), 461
`rules_` (*HostBasedFirewallRuleConfig.FirewallRuleGroup property*), 458
`Run` (class in `cbc_sdk.audit_remediation.base`), 145
`run()` (*GetFileJob method*), 606
`run()` (*JobWorker method*), 607
`run()` (*LiveResponseJobScheduler method*), 609
`run()` (*LRUCacheDict.EmptyCacheThread method*), 554
`run_id()` (*FacetQuery method*), 132
`run_id()` (*ResultQuery method*), 142
`run_job()` (*JobWorker method*), 607
`RunHistory` (class in `cbc_sdk.audit_remediation.base`), 148
`RunHistoryQuery` (class in `cbc_sdk.audit_remediation.base`), 149
`RunQuery` (class in `cbc_sdk.audit_remediation.base`), 152

S

`save()` (*AssetGroup method*), 360
`save()` (*CorePreventionRuleConfig method*), 452
`save()` (*DataCollectionRuleConfig method*), 455
`save()` (*Feed method*), 219
`save()` (*FeedModel method*), 221
`save()` (*Grant method*), 404
`save()` (*Grant.Profile method*), 401
`save()` (*HostBasedFirewallRuleConfig method*), 461
`save()` (*HostBasedFirewallRuleConfig.FirewallRule method*), 457
`save()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup method*), 459
`save()` (*IOC method*), 223
`save()` (*IOC_V2 method*), 226
`save()` (*MutableBaseModel method*), 566
`save()` (*Policy method*), 445
`save()` (*PolicyRule method*), 450
`save()` (*PolicyRuleConfig method*), 464
`save()` (*Report method*), 231

`save()` (*ReportSeverity* method), 235
`save()` (*USBDeviceApproval* method), 193
`save()` (*User* method), 489
`save()` (*Watchlist* method), 239
`save_watchlist()` (*Report* method), 232
`schedule()` (*RunQuery* method), 154
`SCODE_CODE()` (in module *cbc_sdk.winerror*), 622
`SCODE_FACILITY()` (in module *cbc_sdk.winerror*), 622
`SCODE_SEVERITY()` (in module *cbc_sdk.winerror*), 622
`scroll()` (*DeviceSearchQuery* method), 380
`scroll()` (*ResultQuery* method), 142
`search_suggestions()` (*Alert* static method), 255
`search_suggestions()` (*AuthEvent* static method), 206
`search_suggestions()` (*CBAnalyticsAlert* static method), 286
`search_suggestions()` (*ContainerRuntimeAlert* static method), 294
`search_suggestions()` (*DeviceControlAlert* static method), 301
`search_suggestions()` (*HostBasedFirewallAlert* static method), 333
`search_suggestions()` (*IntrusionDetectionSystemAlert* static method), 342
`search_suggestions()` (*Observation* static method), 426
`search_suggestions()` (*WatchlistAlert* static method), 352
`select()` (*BaseAPI* method), 583
`select()` (*CBCloudAPI* method), 126
`select_class_instance()` (in module *cbc_sdk.connection*), 588
`send()` (*CBSDKSessionAdapter* method), 586
`SensorKit` (class in *cbc_sdk.workload.sensor_lifecycle*), 516
`SensorKitQuery` (class in *cbc_sdk.workload.sensor_lifecycle*), 517
`ServerError`, 595, 635
`session_status()` (*LiveResponseSessionManager* method), 618
`set_ad_group_ids()` (*DeviceSearchQuery* method), 380
`set_alert_ids()` (*AlertSearchQuery* method), 260
`set_alert_ids()` (*GroupedAlertSearchQuery* method), 308
`set_alert_ids()` (*LegacyAlertSearchQueryCriterionMixin* method), 410
`set_alert_notes_present()` (*AlertSearchQuery* method), 261
`set_alert_notes_present()` (*GroupedAlertSearchQuery* method), 308
`set_alerttable()` (*Feed.FeedBuilder* method), 216
`set_alerts_enabled()` (*Watchlist.WatchlistBuilder* method), 236
`set_appliance_uuid()` (*VCenterComputeResourceQuery* method), 547
`set_assignment_mode()` (*CorePreventionRuleConfig* method), 452
`set_auth_event_collection()` (*Policy* method), 446
`set_auth_method()` (*User.UserBuilder* method), 485
`set_auto_delete_bad_hash_delay()` (*Policy.PolicyBuilder* method), 438
`set_auto_deregister_interval()` (*Policy.PolicyBuilder* method), 438
`set_auto_scaling_group_name()` (*AWSComputeResourceQuery* method), 527
`set_auto_scaling_group_name()` (*DeviceSearchQuery* method), 380
`set_availability_zone()` (*AWSComputeResourceQuery* method), 527
`set_avira_protection_cloud()` (*Policy.PolicyBuilder* method), 438
`set_blocked_threat_categories()` (*AlertSearchQuery* method), 261
`set_blocked_threat_categories()` (*GroupedAlertSearchQuery* method), 308
`set_blocked_threat_categories()` (*LegacyAlertSearchQueryCriterionMixin* method), 410
`set_categories()` (*AlertSearchQuery* method), 261
`set_categories()` (*GroupedAlertSearchQuery* method), 309
`set_categories()` (*LegacyAlertSearchQueryCriterionMixin* method), 410
`set_category()` (*Feed.FeedBuilder* method), 217
`set_cloud_provider_account_id()` (*AWSComputeResourceQuery* method), 527
`set_cloud_provider_account_id()` (*DeviceSearchQuery* method), 380
`set_cloud_provider_resource_id()` (*AWSComputeResourceQuery* method), 528
`set_cloud_provider_tags()` (*AWSComputeResourceQuery* method), 528
`set_cluster_name()` (*VCenterComputeResourceQuery* method), 547
`set_cluster_names()` (*AlertSearchQuery* method), 261
`set_cluster_names()` (*GroupedAlertSearchQuery* method), 309
`set_cluster_names()` (*LegacyAlertSearchQueryCriterionMixin* method), 410
`set_conditions()` (*Grant.ProfileBuilder* method), 402
`set_create_time()` (*AlertSearchQuery* method), 261
`set_create_time()` (*GroupedAlertSearchQuery* method), 309
`set_create_time()` (*LegacyAlertSearchQueryCriterionMixin* method), 410
`set_data_collection()` (*Policy* method), 446
`set_datacenter_name()` (*VCenterComputeResource*

[Query method](#)), 547
[set_default_action\(\)](#) (*HostBasedFirewallRuleConfig method*), 461
[set_deployment_type\(\)](#) (*AffectedAssetQuery method*), 493
[set_deployment_type\(\)](#) (*DeviceSearchQuery method*), 381
[set_deployment_type\(\)](#) (*VulnerabilityAssetViewQuery method*), 503
[set_deployment_type\(\)](#) (*VulnerabilityQuery method*), 510
[set_description\(\)](#) (*Policy.PolicyBuilder method*), 438
[set_description\(\)](#) (*Report.ReportBuilder method*), 228
[set_description\(\)](#) (*Watchlist.WatchlistBuilder method*), 237
[set_device_guid\(\)](#) (*VCenterComputeResourceQuery method*), 548
[set_device_ids\(\)](#) (*AlertSearchQuery method*), 262
[set_device_ids\(\)](#) (*DeviceSearchQuery method*), 381
[set_device_ids\(\)](#) (*DifferentialQuery method*), 167
[set_device_ids\(\)](#) (*FacetQuery method*), 132
[set_device_ids\(\)](#) (*GroupedAlertSearchQuery method*), 309
[set_device_ids\(\)](#) (*LegacyAlertSearchQueryCriterionMixin method*), 411
[set_device_ids\(\)](#) (*ResultQuery method*), 142
[set_device_ids\(\)](#) (*USBDeviceApprovalQuery method*), 196
[set_device_locations\(\)](#) (*AlertSearchQuery method*), 262
[set_device_locations\(\)](#) (*GroupedAlertSearchQuery method*), 310
[set_device_locations\(\)](#) (*LegacyAlertSearchQueryCriterionMixin method*), 411
[set_device_names\(\)](#) (*AlertSearchQuery method*), 262
[set_device_names\(\)](#) (*FacetQuery method*), 132
[set_device_names\(\)](#) (*GroupedAlertSearchQuery method*), 310
[set_device_names\(\)](#) (*LegacyAlertSearchQueryCriterionMixin method*), 411
[set_device_names\(\)](#) (*ResultQuery method*), 142
[set_device_os\(\)](#) (*AlertSearchQuery method*), 263
[set_device_os\(\)](#) (*FacetQuery method*), 132
[set_device_os\(\)](#) (*GroupedAlertSearchQuery method*), 310
[set_device_os\(\)](#) (*LegacyAlertSearchQueryCriterionMixin method*), 412
[set_device_os\(\)](#) (*ResultQuery method*), 142
[set_device_os_versions\(\)](#) (*AlertSearchQuery method*), 263
[set_device_os_versions\(\)](#) (*GroupedAlertSearchQuery method*), 311
[set_device_os_versions\(\)](#) (*LegacyAlertSearchQueryCriterionMixin method*), 412
[set_device_type\(\)](#) (*AffectedAssetQuery method*), 493
[set_device_type\(\)](#) (*VulnerabilityAssetViewQuery method*), 503
[set_device_type\(\)](#) (*VulnerabilityQuery method*), 510
[set_device_username\(\)](#) (*AlertSearchQuery method*), 263
[set_device_username\(\)](#) (*GroupedAlertSearchQuery method*), 311
[set_device_username\(\)](#) (*LegacyAlertSearchQueryCriterionMixin method*), 412
[set_disabled\(\)](#) (*Grant.Profile method*), 401
[set_disabled\(\)](#) (*Grant.ProfileBuilder method*), 402
[set_egress_group_ids\(\)](#) (*AlertSearchQuery method*), 263
[set_egress_group_ids\(\)](#) (*GroupedAlertSearchQuery method*), 311
[set_egress_group_ids\(\)](#) (*LegacyAlertSearchQueryCriterionMixin method*), 412
[set_egress_group_names\(\)](#) (*AlertSearchQuery method*), 264
[set_egress_group_names\(\)](#) (*GroupedAlertSearchQuery method*), 311
[set_egress_group_names\(\)](#) (*LegacyAlertSearchQueryCriterionMixin method*), 413
[set_eligibility\(\)](#) (*VCenterComputeResourceQuery method*), 548
[set_eligibility_code\(\)](#) (*VCenterComputeResourceQuery method*), 548
[set_email\(\)](#) (*User.UserBuilder method*), 485
[set_enabled\(\)](#) (*HostBasedFirewallRuleConfig method*), 461
[set_endpoint_names\(\)](#) (*USBDeviceQuery method*), 202
[set_esx_host_name\(\)](#) (*VCenterComputeResourceQuery method*), 548
[set_esx_host_uuid\(\)](#) (*VCenterComputeResourceQuery method*), 548
[set_exclude_sensor_versions\(\)](#) (*DeviceSearchQuery method*), 381
[set_expiration\(\)](#) (*Grant.Profile method*), 401
[set_expiration\(\)](#) (*Grant.ProfileBuilder method*), 402
[set_external_device_friendly_names\(\)](#) (*AlertSearchQuery method*), 264
[set_external_device_friendly_names\(\)](#) (*GroupedAlertSearchQuery method*), 312
[set_external_device_friendly_names\(\)](#) (*LegacyAlertSearchQueryCriterionMixin method*), 413
[set_external_device_ids\(\)](#) (*AlertSearchQuery method*), 264
[set_external_device_ids\(\)](#) (*GroupedAlertSearchQuery method*), 312

`set_external_device_ids()` (*LegacyAlertSearchQueryCriterionMixin* method), 413
`set_fields()` (*AsyncProcessQuery* method), 468
`set_fields()` (*AuthEventQuery* method), 213
`set_fields()` (*EnrichedEventQuery* method), 177
`set_fields()` (*EventQuery* method), 396
`set_fields()` (*ObservationQuery* method), 432
`set_fields()` (*Query* method), 572
`set_first_name()` (*User.UserBuilder* method), 486
`set_group_by()` (*AlertSearchQuery* method), 264
`set_group_by()` (*GroupedAlertSearchQuery* method), 312
`set_group_results()` (*AlertSearchQuery* method), 265
`set_group_results()` (*GroupedAlertSearchQuery* method), 312
`set_group_results()` (*LegacyAlertSearchQueryCriterionMixin* method), 413
`set_hashes()` (*RecommendationQuery* method), 187
`set_highest_risk_score()` (*AffectedAssetQuery* method), 493
`set_highest_risk_score()` (*VulnerabilityAssetViewQuery* method), 503
`set_highest_risk_score()` (*VulnerabilityQuery* method), 511
`set_host_name()` (*VCenterComputeResourceQuery* method), 548
`set_id()` (*AWSComputeResourceQuery* method), 528
`set_installation_status()` (*AWSComputeResourceQuery* method), 528
`set_installation_status()` (*VCenterComputeResourceQuery* method), 549
`set_installation_type()` (*VCenterComputeResourceQuery* method), 549
`set_ip_address()` (*VCenterComputeResourceQuery* method), 549
`set_ip_reputations()` (*AlertSearchQuery* method), 265
`set_ip_reputations()` (*GroupedAlertSearchQuery* method), 312
`set_ip_reputations()` (*LegacyAlertSearchQueryCriterionMixin* method), 414
`set_kill_chain_statuses()` (*AlertSearchQuery* method), 265
`set_kill_chain_statuses()` (*GroupedAlertSearchQuery* method), 313
`set_kill_chain_statuses()` (*LegacyAlertSearchQueryCriterionMixin* method), 414
`set_last_contact_time()` (*DeviceSearchQuery* method), 381
`set_last_name()` (*User.UserBuilder* method), 486
`set_last_sync_ts()` (*AffectedAssetQuery* method), 494
`set_last_sync_ts()` (*VulnerabilityAssetViewQuery* method), 503
`set_last_sync_ts()` (*VulnerabilityQuery* method), 511
`set_legacy_alert_ids()` (*AlertSearchQuery* method), 265
`set_legacy_alert_ids()` (*GroupedAlertSearchQuery* method), 313
`set_legacy_alert_ids()` (*LegacyAlertSearchQueryCriterionMixin* method), 414
`set_link()` (*Report.ReportBuilder* method), 228
`set_managed_detection_response_permissions()` (*Policy.PolicyBuilder* method), 438
`set_max_rows()` (*DeviceSearchQuery* method), 381
`set_max_rows()` (*USBDeviceQuery* method), 202
`set_minimum_severity()` (*AlertSearchQuery* method), 266
`set_minimum_severity()` (*GroupedAlertSearchQuery* method), 313
`set_name()` (*AffectedAssetQuery* method), 494
`set_name()` (*AWSComputeResourceQuery* method), 528
`set_name()` (*Feed.FeedBuilder* method), 217
`set_name()` (*Policy.PolicyBuilder* method), 439
`set_name()` (*VCenterComputeResourceQuery* method), 549
`set_name()` (*VulnerabilityAssetViewQuery* method), 503
`set_name()` (*VulnerabilityQuery* method), 511
`set_name()` (*Watchlist.WatchlistBuilder* method), 237
`set_namespaces()` (*AlertSearchQuery* method), 266
`set_namespaces()` (*GroupedAlertSearchQuery* method), 313
`set_namespaces()` (*LegacyAlertSearchQueryCriterionMixin* method), 414
`set_not_blocked_threat_categories()` (*AlertSearchQuery* method), 266
`set_not_blocked_threat_categories()` (*GroupedAlertSearchQuery* method), 313
`set_not_blocked_threat_categories()` (*LegacyAlertSearchQueryCriterionMixin* method), 415
`set_on_access_scan()` (*Policy.PolicyBuilder* method), 439
`set_on_demand_scan()` (*Policy.PolicyBuilder* method), 439
`set_on_demand_scan_schedule()` (*Policy.PolicyBuilder* method), 439
`set_org()` (*Grant.GrantBuilder* method), 399
`set_orgs()` (*Grant.ProfileBuilder* method), 403
`set_os()` (*DeviceSearchQuery* method), 382
`set_os_arch()` (*AffectedAssetQuery* method), 494
`set_os_arch()` (*VulnerabilityAssetViewQuery* method), 504
`set_os_arch()` (*VulnerabilityQuery* method), 511
`set_os_architecture()` (*VCenterComputeResourceQuery* method), 549

set_os_description() (*VCenterComputeResourceQuery method*), 550
 set_os_name() (*AffectedAssetQuery method*), 494
 set_os_name() (*VulnerabilityAssetViewQuery method*), 504
 set_os_name() (*VulnerabilityQuery method*), 511
 set_os_product_id() (*AffectedAssetQuery method*), 495
 set_os_type() (*AffectedAssetQuery method*), 495
 set_os_type() (*VCenterComputeResourceQuery method*), 550
 set_os_type() (*VulnerabilityAssetViewQuery method*), 504
 set_os_type() (*VulnerabilityQuery method*), 512
 set_os_version() (*AffectedAssetQuery method*), 495
 set_os_version() (*VulnerabilityAssetViewQuery method*), 504
 set_os_version() (*VulnerabilityQuery method*), 512
 set_override_list() (*ReputationOverrideQuery method*), 483
 set_override_type() (*ReputationOverrideQuery method*), 483
 set_parameter() (*CorePreventionRuleConfig method*), 453
 set_parameter() (*DataCollectionRuleConfig method*), 455
 set_parameter() (*HostBasedFirewallRuleConfig method*), 462
 set_parameter() (*PolicyRuleConfig method*), 464
 set_phone() (*User.UserBuilder method*), 486
 set_platform() (*AWSComputeResourceQuery method*), 529
 set_platform_details() (*AWSComputeResourceQuery method*), 529
 set_policy_applied() (*AlertSearchQuery method*), 266
 set_policy_applied() (*GroupedAlertSearchQuery method*), 314
 set_policy_applied() (*LegacyAlertSearchQueryCriterionMixin method*), 415
 set_policy_ids() (*AlertSearchQuery method*), 266
 set_policy_ids() (*DeviceSearchQuery method*), 382
 set_policy_ids() (*FacetQuery method*), 133
 set_policy_ids() (*GroupedAlertSearchQuery method*), 314
 set_policy_ids() (*LegacyAlertSearchQueryCriterionMixin method*), 415
 set_policy_ids() (*ResultQuery method*), 143
 set_policy_names() (*AlertSearchQuery method*), 267
 set_policy_names() (*FacetQuery method*), 133
 set_policy_names() (*GroupedAlertSearchQuery method*), 314
 set_policy_names() (*LegacyAlertSearchQueryCriterionMixin method*), 415
 set_policy_names() (*ResultQuery method*), 143
 set_policy_types() (*RecommendationQuery method*), 187
 set_ports() (*AlertSearchQuery method*), 267
 set_ports() (*GroupedAlertSearchQuery method*), 314
 set_ports() (*LegacyAlertSearchQueryCriterionMixin method*), 416
 set_principal_name() (*Grant.GrantBuilder method*), 399
 set_priority() (*Policy.PolicyBuilder method*), 440
 set_process_names() (*AlertSearchQuery method*), 267
 set_process_names() (*GroupedAlertSearchQuery method*), 315
 set_process_names() (*LegacyAlertSearchQueryCriterionMixin method*), 416
 set_process_sha256() (*AlertSearchQuery method*), 268
 set_process_sha256() (*GroupedAlertSearchQuery method*), 315
 set_process_sha256() (*LegacyAlertSearchQueryCriterionMixin method*), 416
 set_product_ids() (*AlertSearchQuery method*), 268
 set_product_ids() (*GroupedAlertSearchQuery method*), 315
 set_product_ids() (*LegacyAlertSearchQueryCriterionMixin method*), 416
 set_product_names() (*AlertSearchQuery method*), 268
 set_product_names() (*GroupedAlertSearchQuery method*), 316
 set_product_names() (*LegacyAlertSearchQueryCriterionMixin method*), 417
 set_product_names() (*USBDeviceApprovalQuery method*), 196
 set_product_names() (*USBDeviceQuery method*), 202
 set_profile_expiration() (*User method*), 489
 set_protocols() (*AlertSearchQuery method*), 268
 set_protocols() (*GroupedAlertSearchQuery method*), 316
 set_protocols() (*LegacyAlertSearchQueryCriterionMixin method*), 417
 set_provider_url() (*Feed.FeedBuilder method*), 217
 set_reason_code() (*AlertSearchQuery method*), 269
 set_reason_code() (*GroupedAlertSearchQuery method*), 316
 set_reason_code() (*LegacyAlertSearchQueryCriterionMixin method*), 417
 set_region() (*AWSComputeResourceQuery method*), 529
 set_registration_id() (*VCenterComputeResourceQuery method*), 550
 set_registry_value() (*CbLRSessionBase method*),

- 604
- `set_registry_value()` (*LiveResponseSession* method), 616
- `set_remote_domains()` (*AlertSearchQuery* method), 269
- `set_remote_domains()` (*GroupedAlertSearchQuery* method), 316
- `set_remote_domains()` (*LegacyAlertSearchQueryCriterionMixin* method), 417
- `set_remote_ips()` (*AlertSearchQuery* method), 269
- `set_remote_ips()` (*GroupedAlertSearchQuery* method), 317
- `set_remote_ips()` (*LegacyAlertSearchQueryCriterionMixin* method), 418
- `set_remote_is_private()` (*AlertSearchQuery* method), 269
- `set_remote_is_private()` (*GroupedAlertSearchQuery* method), 317
- `set_replica_ids()` (*AlertSearchQuery* method), 270
- `set_replica_ids()` (*GroupedAlertSearchQuery* method), 317
- `set_replica_ids()` (*LegacyAlertSearchQueryCriterionMixin* method), 418
- `set_reputations()` (*AlertSearchQuery* method), 270
- `set_reputations()` (*GroupedAlertSearchQuery* method), 317
- `set_reputations()` (*LegacyAlertSearchQueryCriterionMixin* method), 418
- `set_role()` (*User.UserBuilder* method), 486
- `set_roles()` (*Grant.GrantBuilder* method), 400
- `set_roles()` (*Grant.ProfileBuilder* method), 403
- `set_rows()` (*AlertSearchQuery* method), 270
- `set_rows()` (*AssetGroupQuery* method), 363
- `set_rows()` (*AsyncProcessQuery* method), 468
- `set_rows()` (*AuthEventQuery* method), 213
- `set_rows()` (*EnrichedEventQuery* method), 178
- `set_rows()` (*EventFacetQuery* method), 391
- `set_rows()` (*EventQuery* method), 396
- `set_rows()` (*FacetQuery* method), 562
- `set_rows()` (*GroupedAlertSearchQuery* method), 318
- `set_rows()` (*ObservationQuery* method), 432
- `set_rows()` (*Query* method), 572
- `set_rule_ids()` (*AlertSearchQuery* method), 270
- `set_rule_ids()` (*GroupedAlertSearchQuery* method), 318
- `set_rule_ids()` (*LegacyAlertSearchQueryCriterionMixin* method), 418
- `set_rule_names()` (*AlertSearchQuery* method), 271
- `set_rule_names()` (*GroupedAlertSearchQuery* method), 318
- `set_rule_names()` (*LegacyAlertSearchQueryCriterionMixin* method), 419
- `set_run_ids()` (*ResultQuery* method), 143
- `set_run_states()` (*AlertSearchQuery* method), 271
- `set_run_states()` (*GroupedAlertSearchQuery* method), 318
- `set_run_states()` (*LegacyAlertSearchQueryCriterionMixin* method), 419
- `set_sensor_actions()` (*AlertSearchQuery* method), 271
- `set_sensor_actions()` (*GroupedAlertSearchQuery* method), 319
- `set_sensor_actions()` (*LegacyAlertSearchQueryCriterionMixin* method), 419
- `set_serial_numbers()` (*AlertSearchQuery* method), 271
- `set_serial_numbers()` (*GroupedAlertSearchQuery* method), 319
- `set_serial_numbers()` (*LegacyAlertSearchQueryCriterionMixin* method), 420
- `set_serial_numbers()` (*USBDeviceQuery* method), 202
- `set_severity()` (*AffectedAssetQuery* method), 495
- `set_severity()` (*Report.ReportBuilder* method), 229
- `set_severity()` (*VulnerabilityAssetViewQuery* method), 505
- `set_severity()` (*VulnerabilityOrgSummaryQuery* method), 507
- `set_severity()` (*VulnerabilityQuery* method), 512
- `set_signature_update()` (*Policy.PolicyBuilder* method), 440
- `set_signature_update_schedule()` (*Policy.PolicyBuilder* method), 440
- `set_source_label()` (*Feed.FeedBuilder* method), 217
- `set_start()` (*AsyncProcessQuery* method), 468
- `set_start()` (*AuthEventQuery* method), 213
- `set_start()` (*EnrichedEventQuery* method), 178
- `set_start()` (*EventQuery* method), 396
- `set_start()` (*ObservationQuery* method), 432
- `set_start()` (*Query* method), 572
- `set_status()` (*DeviceSearchQuery* method), 382
- `set_statuses()` (*FacetQuery* method), 133
- `set_statuses()` (*RecommendationQuery* method), 187
- `set_statuses()` (*ResultQuery* method), 143
- `set_statuses()` (*USBDeviceQuery* method), 203
- `set_subnet_id()` (*AWSComputeResourceQuery* method), 529
- `set_summary()` (*Feed.FeedBuilder* method), 217
- `set_sync_status()` (*AffectedAssetQuery* method), 496
- `set_sync_status()` (*VulnerabilityAssetViewQuery* method), 505
- `set_sync_status()` (*VulnerabilityQuery* method), 512
- `set_sync_type()` (*AffectedAssetQuery* method), 496
- `set_sync_type()` (*VulnerabilityAssetViewQuery* method), 505
- `set_sync_type()` (*VulnerabilityQuery* method), 513
- `set_system()` (*PolicyQuery* method), 449
- `set_tags()` (*AlertSearchQuery* method), 272

- set_tags() (*GroupedAlertSearchQuery* method), 319
 set_tags() (*LegacyAlertSearchQueryCriterionMixin* method), 420
 set_tags_enabled() (*Watchlist.WatchlistBuilder* method), 237
 set_target_priorities() (*AlertSearchQuery* method), 272
 set_target_priorities() (*DeviceSearchQuery* method), 382
 set_target_priorities() (*GroupedAlertSearchQuery* method), 319
 set_target_priorities() (*LegacyAlertSearchQueryCriterionMixin* method), 420
 set_template_ids() (*RunHistoryQuery* method), 151
 set_threat_cause_vectors() (*AlertSearchQuery* method), 272
 set_threat_cause_vectors() (*GroupedAlertSearchQuery* method), 320
 set_threat_cause_vectors() (*LegacyAlertSearchQueryCriterionMixin* method), 420
 set_threat_ids() (*AlertSearchQuery* method), 272
 set_threat_ids() (*GroupedAlertSearchQuery* method), 320
 set_threat_ids() (*LegacyAlertSearchQueryCriterionMixin* method), 420
 set_threat_notes_present() (*AlertSearchQuery* method), 273
 set_threat_notes_present() (*GroupedAlertSearchQuery* method), 320
 set_time_range() (*AlertSearchQuery* method), 273
 set_time_range() (*AsyncProcessQuery* method), 469
 set_time_range() (*AuthEventQuery* method), 213
 set_time_range() (*EnrichedEventQuery* method), 178
 set_time_range() (*EventFacetQuery* method), 391
 set_time_range() (*EventQuery* method), 396
 set_time_range() (*FacetQuery* method), 562
 set_time_range() (*GroupedAlertSearchQuery* method), 320
 set_time_range() (*ObservationQuery* method), 432
 set_time_range() (*Query* method), 572
 set_time_range() (*SummaryQuery* method), 479
 set_time_received() (*ResultQuery* method), 144
 set_timestamp() (*Report.ReportBuilder* method), 229
 set_title() (*Report.ReportBuilder* method), 229
 set_types() (*AlertSearchQuery* method), 274
 set_types() (*GroupedAlertSearchQuery* method), 321
 set_types() (*LegacyAlertSearchQueryCriterionMixin* method), 421
 set_update_servers_offsite() (*Policy.PolicyBuilder* method), 440
 set_update_servers_onsite() (*Policy.PolicyBuilder* method), 440
 set_update_servers_override() (*Policy.PolicyBuilder* method), 441
 set_uuid() (*VCenterComputeResourceQuery* method), 550
 set_vcenter() (*AffectedAssetQuery* method), 496
 set_vcenter() (*VulnerabilityAssetViewQuery* method), 505
 set_vcenter() (*VulnerabilityOrgSummaryQuery* method), 507
 set_vcenter() (*VulnerabilityQuery* method), 513
 set_vcenter_host_url() (*VCenterComputeResourceQuery* method), 550
 set_vcenter_name() (*VCenterComputeResourceQuery* method), 551
 set_vcenter_uuid() (*VCenterComputeResourceQuery* method), 551
 set_vendor_ids() (*AlertSearchQuery* method), 274
 set_vendor_ids() (*GroupedAlertSearchQuery* method), 322
 set_vendor_ids() (*LegacyAlertSearchQueryCriterionMixin* method), 421
 set_vendor_names() (*AlertSearchQuery* method), 274
 set_vendor_names() (*GroupedAlertSearchQuery* method), 322
 set_vendor_names() (*LegacyAlertSearchQueryCriterionMixin* method), 421
 set_vendor_names() (*USBDeviceApprovalQuery* method), 196
 set_vendor_names() (*USBDeviceQuery* method), 203
 set_virtual_private_cloud_id() (*AWSComputeResourceQuery* method), 529
 set_virtual_private_cloud_id() (*DeviceSearchQuery* method), 382
 set_visibility() (*AffectedAssetQuery* method), 496
 set_visibility() (*Report.ReportBuilder* method), 229
 set_visibility() (*VulnerabilityAssetViewQuery* method), 506
 set_visibility() (*VulnerabilityOrgSummaryQuery* method), 507
 set_visibility() (*VulnerabilityQuery* method), 513
 set_vm_id() (*AffectedAssetQuery* method), 496
 set_vm_id() (*VulnerabilityAssetViewQuery* method), 506
 set_vm_id() (*VulnerabilityQuery* method), 513
 set_vmwaretools_version() (*VCenterComputeResourceQuery* method), 551
 set_vuln_count() (*AffectedAssetQuery* method), 497
 set_vuln_count() (*VulnerabilityAssetViewQuery* method), 506
 set_vuln_count() (*VulnerabilityQuery* method), 514
 set_watchlist_ids() (*AlertSearchQuery* method), 275
 set_watchlist_ids() (*GroupedAlertSearchQuery* method), 322
 set_watchlist_ids() (*LegacyAlertSearchQueryCriterionMixin* method), 421

rionMixin method), 422
`set_watchlist_names()` (*AlertSearchQuery method*), 275
`set_watchlist_names()` (*GroupedAlertSearchQuery method*), 322
`set_watchlist_names()` (*LegacyAlertSearchQueryCriterionMixin method*), 422
`set_workflows()` (*AlertSearchQuery method*), 275
`set_workflows()` (*GroupedAlertSearchQuery method*), 323
`set_workflows()` (*LegacyAlertSearchQueryCriterionMixin method*), 422
`set_workload_ids()` (*AlertSearchQuery method*), 275
`set_workload_ids()` (*GroupedAlertSearchQuery method*), 323
`set_workload_ids()` (*LegacyAlertSearchQueryCriterionMixin method*), 422
`set_workload_kinds()` (*AlertSearchQuery method*), 276
`set_workload_kinds()` (*GroupedAlertSearchQuery method*), 323
`set_workload_kinds()` (*LegacyAlertSearchQueryCriterionMixin method*), 423
`set_workload_names()` (*AlertSearchQuery method*), 276
`set_workload_names()` (*GroupedAlertSearchQuery method*), 323
`set_workload_names()` (*LegacyAlertSearchQueryCriterionMixin method*), 423
`set_xdr_collection()` (*Policy method*), 446
`setDaemon()` (*JobWorker method*), 607
`setDaemon()` (*LiveResponseJobScheduler method*), 609
`setDaemon()` (*LRUCacheDict.EmptyCacheThread method*), 555
`setName()` (*JobWorker method*), 607
`setName()` (*LiveResponseJobScheduler method*), 609
`setName()` (*LRUCacheDict.EmptyCacheThread method*), 555
`severity_levels()` (*Vulnerability.OrgSummary method*), 499
`siblings` (*Process property*), 475
`SimpleQuery` (*class in cbc_sdk.base*), 576
`sort()` (*FeedQuery method*), 222
`sort()` (*ReportQuery method*), 234
`sort()` (*SimpleQuery method*), 577
`sort()` (*Vulnerability.AssetView method*), 499
`sort()` (*WatchlistQuery method*), 241
`sort_by()` (*AffectedAssetQuery method*), 497
`sort_by()` (*AlertSearchQuery method*), 276
`sort_by()` (*AssetGroupQuery method*), 363
`sort_by()` (*AsyncProcessQuery method*), 469
`sort_by()` (*AuthEventQuery method*), 214
`sort_by()` (*AWSComputeResourceQuery method*), 529
`sort_by()` (*BaseComputeResourceQuery method*), 536
`sort_by()` (*DeviceSearchQuery method*), 383
`sort_by()` (*EnrichedEventQuery method*), 178
`sort_by()` (*EventQuery method*), 396
`sort_by()` (*GroupedAlertSearchQuery method*), 324
`sort_by()` (*ObservationQuery method*), 433
`sort_by()` (*Query method*), 572
`sort_by()` (*RecommendationQuery method*), 188
`sort_by()` (*ReputationOverrideQuery method*), 483
`sort_by()` (*ResultQuery method*), 144
`sort_by()` (*RunHistoryQuery method*), 151
`sort_by()` (*TemplateHistoryQuery method*), 162
`sort_by()` (*USBDeviceQuery method*), 203
`sort_by()` (*VCenterComputeResourceQuery method*), 551
`sort_by()` (*VulnerabilityAssetViewQuery method*), 506
`sort_by()` (*VulnerabilityQuery method*), 514
`start()` (*JobWorker method*), 607
`start()` (*LiveResponseJobScheduler method*), 609
`start()` (*LRUCacheDict.EmptyCacheThread method*), 555
`start_memdump()` (*CbLRSessionBase method*), 604
`start_memdump()` (*LiveResponseSession method*), 616
`start_request()` (*NSXRemediationJob class method*), 515
`status` (*NSXRemediationJob property*), 516
`stop()` (*Run method*), 147
`stop()` (*RunHistory method*), 149
`stop()` (*Template method*), 158
`stop()` (*TemplateHistory method*), 160
`stop_keepalive_thread()` (*CbLRManagerBase method*), 598
`stop_keepalive_thread()` (*LiveResponseSessionManager method*), 618
`submit()` (*DifferentialQuery method*), 168
`submit()` (*RunQuery method*), 155
`submit()` (*VulnerabilityOrgSummaryQuery method*), 508
`submit_job()` (*CbLRManagerBase method*), 598
`submit_job()` (*LiveResponseJobScheduler method*), 609
`submit_job()` (*LiveResponseSessionManager method*), 618
`SUCCEEDED()` (*in module cbc_sdk.winerror*), 622
`summarize()` (*AWSComputeResourceQuery method*), 530
`summary` (*Binary property*), 244
`summary` (*Process property*), 475
`SummaryQuery` (*class in cbc_sdk.platform.processes*), 477
`swagger_meta_file` (*AssetGroup attribute*), 360
`swagger_meta_file` (*Device attribute*), 372
`SwaggerLoader` (*class in cbc_sdk.base*), 578
`systemPolicy` (*Policy property*), 446

T

- Template (class in `cbc_sdk.audit_remediation.base`), 155
- TemplateHistory (class `cbc_sdk.audit_remediation.base`), 158
- TemplateHistoryQuery (class `cbc_sdk.audit_remediation.base`), 160
- terms_ (AuthEventFacet property), 209
- terms_ (EnrichedEventFacet property), 175
- terms_ (EventFacet property), 388
- terms_ (ObservationFacet property), 429
- terms_ (ProcessFacet property), 477
- timeout() (AsyncProcessQuery method), 469
- timeout() (AuthEventQuery method), 214
- timeout() (EnrichedEventQuery method), 179
- timeout() (EventFacetQuery method), 392
- timeout() (FacetQuery method), 563
- timeout() (ObservationQuery method), 433
- timeout() (SummaryQuery method), 479
- TimeoutError, 595, 636
- to_dict() (Credentials method), 590
- to_json() (Alert method), 255
- to_json() (Alert.Note method), 252
- to_json() (AssetGroup method), 360
- to_json() (AuditLog method), 365
- to_json() (AuthEvent method), 207
- to_json() (AuthEventFacet method), 209
- to_json() (AuthEventFacet.Ranges method), 208
- to_json() (AuthEventFacet.Terms method), 208
- to_json() (AWSComputeResource method), 521
- to_json() (BaseComputeResource method), 532
- to_json() (Binary method), 244
- to_json() (Binary.Summary method), 243
- to_json() (CBAnalyticsAlert method), 286
- to_json() (CBAnalyticsAlert.Note method), 282
- to_json() (ComputeResourceFacet method), 538
- to_json() (ComputeResourceFacet.ComputeResourceFacetValue method), 537
- to_json() (ContainerRuntimeAlert method), 294
- to_json() (ContainerRuntimeAlert.Note method), 291
- to_json() (CorePreventionRuleConfig method), 453
- to_json() (DataCollectionRuleConfig method), 455
- to_json() (Device method), 372
- to_json() (DeviceControlAlert method), 302
- to_json() (DeviceControlAlert.Note method), 298
- to_json() (DeviceFacet method), 375
- to_json() (DeviceFacet.DeviceFacetValue method), 375
- to_json() (DeviceSummary method), 128
- to_json() (DeviceSummary.Metrics method), 128
- to_json() (DeviceSummaryFacet method), 129
- to_json() (DeviceSummaryFacet.Values method), 129
- to_json() (Differential method), 164
- to_json() (Downloads method), 245
- to_json() (Downloads.FoundItem method), 244
- to_json() (EnrichedEvent method), 173
- to_json() (EnrichedEventFacet method), 175
- to_json() (EnrichedEventFacet.Ranges method), 174
- to_json() (EnrichedEventFacet.Terms method), 174
- to_json() (Event method), 385
- to_json() (EventFacet method), 388
- to_json() (EventFacet.Ranges method), 387
- to_json() (EventFacet.Terms method), 387
- to_json() (Feed method), 219
- to_json() (FeedModel method), 221
- to_json() (Grant method), 405
- to_json() (Grant.Profile method), 401
- to_json() (GroupedAlert method), 304
- to_json() (HostBasedFirewallAlert method), 333
- to_json() (HostBasedFirewallAlert.Note method), 329
- to_json() (HostBasedFirewallRuleConfig method), 462
- to_json() (HostBasedFirewallRuleConfig.FirewallRule method), 457
- to_json() (HostBasedFirewallRuleConfig.FirewallRuleGroup method), 459
- to_json() (IntrusionDetectionSystemAlert method), 342
- to_json() (IntrusionDetectionSystemAlert.Note method), 338
- to_json() (IOC method), 223
- to_json() (IOC_V2 method), 227
- to_json() (Job method), 408
- to_json() (MutableBaseModel method), 566
- to_json() (NetworkThreatMetadata method), 424
- to_json() (NewBaseModel method), 567
- to_json() (Observation method), 427
- to_json() (ObservationFacet method), 429
- to_json() (ObservationFacet.Ranges method), 428
- to_json() (ObservationFacet.Terms method), 428
- to_json() (PlatformModel method), 246
- to_json() (Policy method), 446
- to_json() (PolicyRule method), 450
- to_json() (PolicyRuleConfig method), 464
- to_json() (Process method), 475
- to_json() (Process.Summary method), 472
- to_json() (Process.Tree method), 473
- to_json() (ProcessFacet method), 477
- to_json() (ProcessFacet.Ranges method), 476
- to_json() (ProcessFacet.Terms method), 477
- to_json() (Recommendation method), 185
- to_json() (Recommendation.RecommendationApplication method), 181
- to_json() (Recommendation.RecommendationImpact method), 182
- to_json() (Recommendation.RecommendationNewRule method), 183

[to_json\(\)](#) (*RecommendationWorkflow method*), 184
[to_json\(\)](#) (*Report method*), 232
[to_json\(\)](#) (*ReportSeverity method*), 235
[to_json\(\)](#) (*ReputationOverride method*), 481
[to_json\(\)](#) (*Result method*), 137
[to_json\(\)](#) (*Result.Device method*), 135
[to_json\(\)](#) (*Result.Fields method*), 135
[to_json\(\)](#) (*Result.Metrics method*), 136
[to_json\(\)](#) (*ResultFacet method*), 138
[to_json\(\)](#) (*ResultFacet.Values method*), 138
[to_json\(\)](#) (*Run method*), 147
[to_json\(\)](#) (*RunHistory method*), 149
[to_json\(\)](#) (*SensorKit method*), 517
[to_json\(\)](#) (*Template method*), 158
[to_json\(\)](#) (*TemplateHistory method*), 160
[to_json\(\)](#) (*UnrefreshableModel method*), 578
[to_json\(\)](#) (*USBDevice method*), 190
[to_json\(\)](#) (*USBDeviceApproval method*), 193
[to_json\(\)](#) (*USBDeviceBlock method*), 198
[to_json\(\)](#) (*User method*), 489
[to_json\(\)](#) (*VCenterComputeResource method*), 540
[to_json\(\)](#) (*Vulnerability method*), 500
[to_json\(\)](#) (*Vulnerability.OrgSummary method*), 500
[to_json\(\)](#) (*Watchlist method*), 240
[to_json\(\)](#) (*WatchlistAlert method*), 352
[to_json\(\)](#) (*WatchlistAlert.Note method*), 348
[touch\(\)](#) (*AssetGroup method*), 361
[touch\(\)](#) (*CorePreventionRuleConfig method*), 453
[touch\(\)](#) (*DataCollectionRuleConfig method*), 455
[touch\(\)](#) (*Feed method*), 219
[touch\(\)](#) (*FeedModel method*), 221
[touch\(\)](#) (*Grant method*), 405
[touch\(\)](#) (*Grant.Profile method*), 401
[touch\(\)](#) (*HostBasedFirewallRuleConfig method*), 462
[touch\(\)](#) (*HostBasedFirewallRuleConfig.FirewallRule method*), 457
[touch\(\)](#) (*HostBasedFirewallRuleConfig.FirewallRuleGroup method*), 459
[touch\(\)](#) (*IOC method*), 224
[touch\(\)](#) (*IOC_V2 method*), 227
[touch\(\)](#) (*MutableBaseModel method*), 567
[touch\(\)](#) (*Policy method*), 446
[touch\(\)](#) (*PolicyRule method*), 450
[touch\(\)](#) (*PolicyRuleConfig method*), 464
[touch\(\)](#) (*Report method*), 232
[touch\(\)](#) (*ReportSeverity method*), 235
[touch\(\)](#) (*USBDeviceApproval method*), 193
[touch\(\)](#) (*User method*), 489
[touch\(\)](#) (*Watchlist method*), 240
[tree\(\)](#) (*Process property*), 475
[try_json\(\)](#) (*in module cbc_sdk.connection*), 588

U

[UnauthorizedError](#), 595
[unignore\(\)](#) (*IOC_V2 method*), 227
[unignore\(\)](#) (*Report method*), 232
[uninstall_sensor\(\)](#) (*Device method*), 372
[uninstall_sensor\(\)](#) (*DeviceSearchQuery method*), 383
[UnrefreshableModel](#) (*class in cbc_sdk.base*), 578
[update\(\)](#) (*Alert method*), 255
[update\(\)](#) (*AlertSearchQuery method*), 277
[update\(\)](#) (*CBAAnalyticsAlert method*), 287
[update\(\)](#) (*ContainerRuntimeAlert method*), 294
[update\(\)](#) (*DeviceControlAlert method*), 302
[update\(\)](#) (*Feed method*), 219
[update\(\)](#) (*GroupedAlertSearchQuery method*), 324
[update\(\)](#) (*HostBasedFirewallAlert method*), 333
[update\(\)](#) (*IntrusionDetectionSystemAlert method*), 343
[update\(\)](#) (*Report method*), 232
[update\(\)](#) (*Watchlist method*), 240
[update\(\)](#) (*WatchlistAlert method*), 352
[update_criteria\(\)](#) (*AlertSearchQuery method*), 277
[update_criteria\(\)](#) (*AssetGroupQuery method*), 363
[update_criteria\(\)](#) (*AsyncProcessQuery method*), 470
[update_criteria\(\)](#) (*AuthEventQuery method*), 214
[update_criteria\(\)](#) (*AWSComputeResourceQuery method*), 530
[update_criteria\(\)](#) (*BaseComputeResourceQuery method*), 536
[update_criteria\(\)](#) (*CriteriaBuilderSupportMixin method*), 558
[update_criteria\(\)](#) (*DeviceSearchQuery method*), 383
[update_criteria\(\)](#) (*DifferentialQuery method*), 168
[update_criteria\(\)](#) (*EnrichedEventQuery method*), 179
[update_criteria\(\)](#) (*EventFacetQuery method*), 392
[update_criteria\(\)](#) (*EventQuery method*), 397
[update_criteria\(\)](#) (*FacetQuery method*), 133, 563
[update_criteria\(\)](#) (*GroupedAlertSearchQuery method*), 324
[update_criteria\(\)](#) (*ObservationQuery method*), 433
[update_criteria\(\)](#) (*Query method*), 573
[update_criteria\(\)](#) (*RecommendationQuery method*), 188
[update_criteria\(\)](#) (*ResultQuery method*), 144
[update_criteria\(\)](#) (*RunHistoryQuery method*), 152
[update_criteria\(\)](#) (*SensorKitQuery method*), 519
[update_criteria\(\)](#) (*TemplateHistoryQuery method*), 162
[update_criteria\(\)](#) (*USBDeviceApprovalQuery method*), 196
[update_criteria\(\)](#) (*USBDeviceQuery method*), 203
[update_criteria\(\)](#) (*VCenterComputeResourceQuery method*), 552

- [update_exclusions\(\)](#) (*AlertSearchQuery* method), 278
[update_exclusions\(\)](#) (*AsyncProcessQuery* method), 470
[update_exclusions\(\)](#) (*AuthEventQuery* method), 215
[update_exclusions\(\)](#) (*EnrichedEventQuery* method), 179
[update_exclusions\(\)](#) (*EventFacetQuery* method), 393
[update_exclusions\(\)](#) (*EventQuery* method), 397
[update_exclusions\(\)](#) (*ExclusionBuilderSupportMixin* method), 559
[update_exclusions\(\)](#) (*FacetQuery* method), 564
[update_exclusions\(\)](#) (*GroupedAlertSearchQuery* method), 325
[update_exclusions\(\)](#) (*ObservationQuery* method), 434
[update_exclusions\(\)](#) (*Query* method), 573
[update_policy\(\)](#) (*Device* method), 373
[update_policy\(\)](#) (*DeviceSearchQuery* method), 384
[update_sensor_version\(\)](#) (*Device* method), 373
[update_sensor_version\(\)](#) (*DeviceSearchQuery* method), 384
[update_threat\(\)](#) (*Alert* method), 256
[update_threat\(\)](#) (*CBAanalyticsAlert* method), 287
[update_threat\(\)](#) (*ContainerRuntimeAlert* method), 295
[update_threat\(\)](#) (*DeviceControlAlert* method), 302
[update_threat\(\)](#) (*HostBasedFirewallAlert* method), 334
[update_threat\(\)](#) (*IntrusionDetectionSystemAlert* method), 343
[update_threat\(\)](#) (*WatchlistAlert* method), 353
[url](#) (*BaseAPI* property), 584
[url](#) (*CBCloudAPI* property), 126
[urn](#) (*User* property), 489
[USBDevice](#) (class in *cbc_sdk.endpoint_standard.usb_device_control*), 189
[USBDeviceApproval](#) (class in *cbc_sdk.endpoint_standard.usb_device_control*), 191
[USBDeviceApprovalQuery](#) (class in *cbc_sdk.endpoint_standard.usb_device_control*), 194
[USBDeviceBlock](#) (class in *cbc_sdk.endpoint_standard.usb_device_control*), 197
[USBDeviceBlockQuery](#) (class in *cbc_sdk.endpoint_standard.usb_device_control*), 198
[USBDeviceQuery](#) (class in *cbc_sdk.endpoint_standard.usb_device_control*), 199
[User](#) (class in *cbc_sdk.platform.users*), 484
[User.UserBuilder](#) (class in *cbc_sdk.platform.users*), 485
[user_ids\(\)](#) (*UserQuery* method), 490
[UserQuery](#) (class in *cbc_sdk.platform.users*), 489
- ## V
- [valid_rule_configs\(\)](#) (*Policy* method), 446
[validate\(\)](#) (*AssetGroup* method), 361
[validate\(\)](#) (*CorePreventionRuleConfig* method), 453
[validate\(\)](#) (*DataCollectionRuleConfig* method), 455
[validate\(\)](#) (*Feed* method), 220
[validate\(\)](#) (*FeedModel* method), 221
[validate\(\)](#) (*Grant* method), 405
[validate\(\)](#) (*Grant.Profile* method), 401
[validate\(\)](#) (*HostBasedFirewallRuleConfig* method), 462
[validate\(\)](#) (*HostBasedFirewallRuleConfig.FirewallRule* method), 457
[validate\(\)](#) (*HostBasedFirewallRuleConfig.FirewallRuleGroup* method), 459
[validate\(\)](#) (*IOC* method), 224
[validate\(\)](#) (*IOC_V2* method), 227
[validate\(\)](#) (*MutableBaseModel* method), 567
[validate\(\)](#) (*Policy* method), 447
[validate\(\)](#) (*PolicyRule* method), 450
[validate\(\)](#) (*PolicyRuleConfig* method), 464
[validate\(\)](#) (*Report* method), 233
[validate\(\)](#) (*ReportSeverity* method), 235
[validate\(\)](#) (*USBDeviceApproval* method), 193
[validate\(\)](#) (*User* method), 489
[validate\(\)](#) (*Watchlist* method), 240
[validate_process_query\(\)](#) (*CBCloudAPI* method), 126
[values](#) (*ComputeResourceFacet* property), 538
[values_](#) (*DeviceFacet* property), 375
[values_](#) (*DeviceSummaryFacet* property), 129
[values_](#) (*ResultFacet* property), 138
[VCenterComputeResource](#) (class in *cbc_sdk.workload.vm_workloads_search*), 538
[VCenterComputeResourceQuery](#) (class in *cbc_sdk.workload.vm_workloads_search*), 540
[Vulnerability](#) (class in *cbc_sdk.platform.vulnerability_assessment*), 497
[Vulnerability.AssetView](#) (class in *cbc_sdk.platform.vulnerability_assessment*), 498
[Vulnerability.OrgSummary](#) (class in *cbc_sdk.platform.vulnerability_assessment*), 499
[vulnerability_refresh\(\)](#) (*Device* method), 373
[VulnerabilityAssetViewQuery](#) (class in *cbc_sdk.platform.vulnerability_assessment*),

- 501
- VulnerabilityOrgSummaryQuery (class in *cbc_sdk.platform.vulnerability_assessment*), 507
- VulnerabilityQuery (class in *cbc_sdk.platform.vulnerability_assessment*), 508
- ## W
- wait() (*LiveResponseMemdump* method), 609
- walk() (*CbLRSessionBase* method), 605
- walk() (*LiveResponseSession* method), 617
- Watchlist (class in *cbc_sdk.enterprise_edr.threat_intelligence*), 235
- Watchlist.WatchlistBuilder (class in *cbc_sdk.enterprise_edr.threat_intelligence*), 236
- WatchlistAlert (class in *cbc_sdk.platform.alerts*), 344
- WatchlistAlert.Note (class in *cbc_sdk.platform.alerts*), 347
- WatchlistQuery (class in *cbc_sdk.enterprise_edr.threat_intelligence*), 240
- where() (*AffectedAssetQuery* method), 497
- where() (*AlertSearchQuery* method), 278
- where() (*AssetGroupQuery* method), 364
- where() (*AsyncProcessQuery* method), 470
- where() (*AuthEventQuery* method), 215
- where() (*AWSComputeResourceQuery* method), 531
- where() (*BaseComputeResourceQuery* method), 536
- where() (*DeviceSearchQuery* method), 384
- where() (*EnrichedEventQuery* method), 180
- where() (*EventFacetQuery* method), 393
- where() (*EventQuery* method), 398
- where() (*FacetQuery* method), 134, 564
- where() (*FeedQuery* method), 222
- where() (*GroupedAlertSearchQuery* method), 325
- where() (*ObservationQuery* method), 434
- where() (*Query* method), 573
- where() (*QueryBuilder* method), 575
- where() (*QueryBuilderSupportMixin* method), 576
- where() (*ReportQuery* method), 234
- where() (*ReputationOverrideQuery* method), 484
- where() (*ResultQuery* method), 145
- where() (*RunHistoryQuery* method), 152
- where() (*RunQuery* method), 155
- where() (*SimpleQuery* method), 577
- where() (*SummaryQuery* method), 479
- where() (*TemplateHistoryQuery* method), 163
- where() (*USBDeviceApprovalQuery* method), 197
- where() (*USBDeviceQuery* method), 204
- where() (*VCenterComputeResourceQuery* method), 552
- where() (*VulnerabilityAssetViewQuery* method), 507
- where() (*VulnerabilityQuery* method), 514
- where() (*WatchlistQuery* method), 241
- Win32Error (class in *cbc_sdk.winerror*), 622
- with_traceback() (*ApiError* method), 591
- with_traceback() (*ClientError* method), 591
- with_traceback() (*ConnectionError* method), 591
- with_traceback() (*CredentialError* method), 592
- with_traceback() (*FunctionalityDecommissioned* method), 592
- with_traceback() (*InvalidHashError* method), 592
- with_traceback() (*InvalidObjectError* method), 593
- with_traceback() (*LiveResponseError* method), 608
- with_traceback() (*ModelNotFound* method), 593
- with_traceback() (*MoreThanOneResultError* method), 593
- with_traceback() (*NonQueryableModel* method), 594
- with_traceback() (*NSXJobError* method), 593
- with_traceback() (*ObjectNotFoundError* method), 594
- with_traceback() (*OperationCancelled* method), 594
- with_traceback() (*QuerySyntaxError* method), 595
- with_traceback() (*ServerError* method), 595
- with_traceback() (*TimeoutError* method), 595
- with_traceback() (*UnauthorizedError* method), 596
- WorkerStatus (class in *cbc_sdk.live_response_api*), 618
- workflow_ (*Alert* property), 256
- workflow_ (*CBAnalyticsAlert* property), 287
- workflow_ (*ContainerRuntimeAlert* property), 295
- workflow_ (*DeviceControlAlert* property), 303
- workflow_ (*HostBasedFirewallAlert* property), 334
- workflow_ (*IntrusionDetectionSystemAlert* property), 343
- workflow_ (*Recommendation* property), 185
- workflow_ (*WatchlistAlert* property), 353
- WorkItem (class in *cbc_sdk.live_response_api*), 618