

---

# **Carbon Black Cloud Python API Documentation**

***Release 1.5.1***

**Carbon Black Developer Network**

**Apr 18, 2024**



# USER GUIDE

<b>1</b>	<b>Major Features</b>	<b>3</b>
<b>2</b>	<b>Audience for the SDK</b>	<b>5</b>
<b>3</b>	<b>API Credentials</b>	<b>7</b>
<b>4</b>	<b>Getting Started</b>	<b>9</b>
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 . . . . .	120
4.7	Audit and Remediation Package . . . . .	131
4.8	Credential Providers Package . . . . .	172
4.9	Endpoint Standard Package . . . . .	176
4.10	Enterprise EDR Package . . . . .	208
4.11	Platform Package . . . . .	249
4.12	Workload Package . . . . .	507
4.13	CBC SDK Package . . . . .	546
4.14	Logging & Diagnostics . . . . .	617
4.15	Testing . . . . .	617
4.16	Changelog . . . . .	619
4.17	Exceptions . . . . .	629
<b>5</b>	<b>Indices and tables</b>	<b>631</b>
	<b>Python Module Index</b>	<b>633</b>
	<b>Index</b>	<b>635</b>



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 system<sup>1</sup>.

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

---

<sup>1</sup> <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=ABCDEFGHijklmno123456789/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. The file must be encoded as UTF-8, or as UTF-16 using either big-endian or little-endian format.

**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.

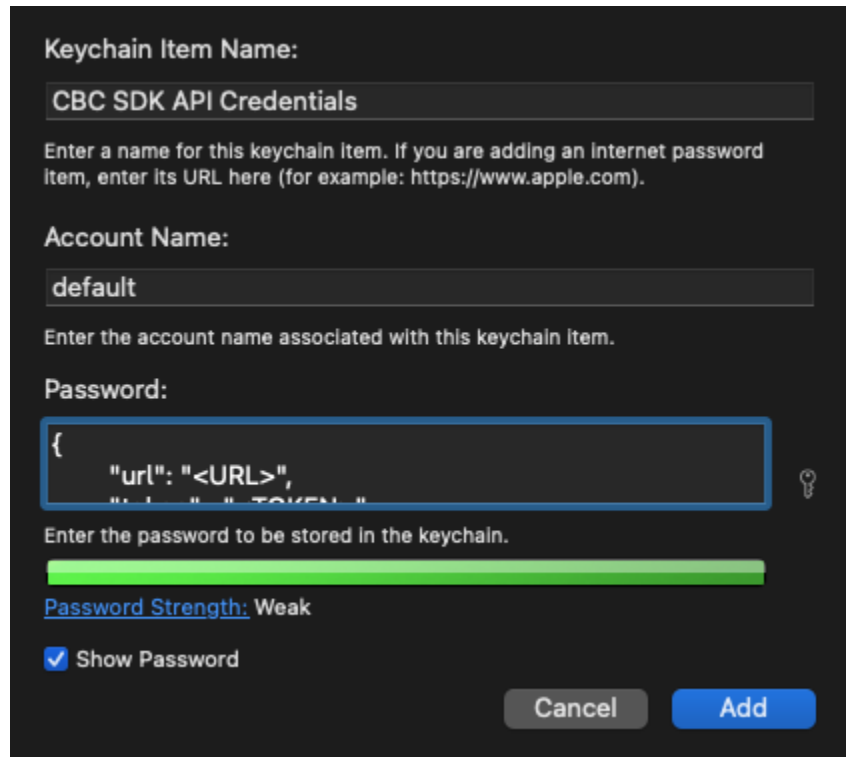
```
{
  "url": "<URL>",
  "token" : "<TOKEN>",
  "org_key": "<ORG_KEY>",
  "ssl_verify": true,
  "ssl_verify_hostname": true,
  "ssl_cert_file": "<FILE_PATH>",
  "ssl_force_tls_1_2": true,
  "proxy": "<NAME_OF_THE_PROXY_HOST>",
  "ignore_system_proxy": true,
  "integration": "<INTEGRATION_NAME>",
  "default_timeout": 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 contains 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'. At the bottom 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 <a href="#">Developer Network Authentication Guide</a> . 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=ABCDEFGHGIJKLMNOPQRSTUVWXYZ/ABCDEFGHIJ
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=ABCDEFGHGIJKLMNOPQRSTUVWXYZ/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.confdeploy.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.

## Devices

*Devices*, also known as *endpoints*, are at the heart of Carbon Black Cloud's functionality. Each device has a Carbon Black Cloud sensor installed on it, which communicates with Carbon Black analytics and the Carbon Black Cloud back end.

Using the Carbon Black Cloud SDK, you can search for devices with a wide range of criteria, filtering on many different fields. You can also perform actions on individual devices, such as setting quarantine status, setting bypass status, or upgrading to a new sensor version.

## Searching for Devices

Using a query of the Device object, you can list the devices configured for your organization:

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import Device
>>> query = api.select(Device).where("os:WINDOWS")
>>> query.add_criteria('target_priority', ['LOW']).add_criteria('virtualization_provider', ['VirtualBox'])
>>> for d in query:
...     print(f"{d.name} - {d.last_internal_ip_address}")
DESKTOP-A19 - 10.0.2.44
DESKTOP-Q210 - 10.10.25.169
DESKTOP-Q211 - 10.10.25.170
DESKTOP-Q211B - 10.10.25.180
EVALUATION-1 - 10.0.2.51
EVALUATION-2 - 10.0.2.52
STAGING-1A - 192.168.1.99
ZZIGNORE-1 - 10.0.3.74
```

The criteria supported in the `where()` and `add_criteria()` query methods are too numerous to enumerate here; please see [the Developer Network documentation](#) for more details.

The results of a search query can also be exported:

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import Device
>>> query = api.select(Device).where("os:WINDOWS")
>>> query.add_criteria('target_priority', ['LOW']).add_criteria('virtualization_provider', ['VirtualBox'])
>>> job = query.export()
>>> csv_report = job.get_output_as_string()
>>> # can also get the output as a file or as enumerated lines of text
```

## Faceting

Facet search queries return statistical information indicating the relative weighting of the requested values as per the specified criteria. Device queries support faceting:

```
>>> from cbc_sdk import CBCloudAPI
>>> api = CBCloudAPI(profile='sample')
>>> from cbc_sdk.platform import Device
>>> query = api.select(Device).where("os:WINDOWS")
>>> query.add_criteria('target_priority', ['LOW']).add_criteria('virtualization_provider', ['VirtualBox'])
>>> facets = query.facets(['policy_id'])
>>> for value in facets[0].values_:
...     print(f"Policy ID {value.id}: {value.total} device(s)")
Policy ID 8801: 4 device(s)
Policy ID 81664: 3 device(s)
Policy ID 82804: 1 device(s)
```

Note that you can facet on multiple fields by passing more than one field name to the `facets()` call. It returns one `DeviceFacet` object per field name, each of which may contain multiple `DeviceFacetValue` objects.

## Search Scrolling

A Device Search request can return no more than 10,000 items at a time. Some customers may have more endpoints than that; to return *all* devices, you can use the `scroll()` method on the query to continue searching after all devices that have been previously returned. This snippet illustrates the technique:

```
# assume "api" is your CBCloudAPI reference
query = api.select(Device)
# add search terms and/or criteria to the query (not shown here)
while query.num_remaining is None or query.num_remaining > 0:
    devicelist = query.scroll() # fetch the batch - 10,000 is default
    for d in devicelist:
        do_something_with_device(d) # whatever you need for each device
```

## Device Actions

Most device actions in the Carbon Black Cloud can be performed on a single device through the `Device` object, on multiple devices specified by ID, or on the results of a device query.

### Bypass Enable/Disable

Setting a device to *bypass* disables all enforcement on the device; its sensor stops sending data to the Carbon Black Cloud.

Setting bypass on a single device:

```
>>> # assume "api" is your CBCloudAPI reference
>>> d = api.select(Device, 12345)
>>> d.bypass(True)
```

Setting bypass on multiple devices:

```
>>> # assume "api" is your CBCloudAPI reference
api.device_bypass([1001, 1002, 1003], True)
```

Setting bypass on the results of a device search:

```
>>> # assume "api" is your CBCloudAPI reference
query = api.select(Device)
# add search terms and/or criteria to the query (not shown here)
query.bypass(True)
```

### Quarantine

A device that has been *quarantined* has its outbound traffic limited, and all inbound traffic to it stopped, except for communication with the Carbon Black Cloud back end. This would be used on any device determined to be interacting maliciously.

Setting quarantine on a single device:

```
>>> # assume "api" is your CBCloudAPI reference
>>> d = api.select(Device, 12345)
>>> d.quarantine(True)
```

Setting quarantine on multiple devices:

```
>>> # assume "api" is your CBCloudAPI reference
api.device_quarantine([1001, 1002, 1003], True)
```

Setting quarantine on the results of a device search:

```
>>> # assume "api" is your CBCloudAPI reference
query = api.select(Device)
# add search terms and/or criteria to the query (not shown here)
query.quarantine(True)
```

### Background Scan

Enabling *background scan* causes a one-time inventory scan on the device to identify any malware files already present there. Disabling background scan causes any background scan currently running on the device to be temporarily suspended; it will restart when background scan is enabled again, or when the endpoint restarts.

Enabling background scan on a single device:

```
>>> # assume "api" is your CBCloudAPI reference
>>> d = api.select(Device, 12345)
>>> d.background_scan(True)
```

Enabling background scan on multiple devices:

```
>>> # assume "api" is your CBCloudAPI reference
api.device_background_scan([1001, 1002, 1003], True)
```

Enabling background scan on the results of a device search:

```
>>> # assume "api" is your CBCloudAPI reference
query = api.select(Device)
# add search terms and/or criteria to the query (not shown here)
query.background_scan(True)
```

## 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}")
...
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('jcdqsju4utpaayj5dh5r21lzffeolg0u').
↳ 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
```

(continues on next page)

(continued from previous page)

```

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': [],
 'older_run_create_time': '2022-08-10T12:57:03.872Z',
 'older_run_id': 'yhb3wcea9y1l4asiltky5tupkgauzas',
 '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('yhb3wcea9y1l4asiltky5tupkgauzas').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

```

(continues on next page)

(continued from previous page)

```

→': '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')
>>>
>>> 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')

```

(continues on next page)

(continued from previous page)

```
>>>
>>> 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\\\ndbutil_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_\ndwhich {run.in_progress_count} are in progress and {run.not_started_count} have not_\ndstarted')
Run id: vsc2be500dcuhc1q5bhvq7kdwoqh367i has 97 active devices in the org of which 0 are_\ndin 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_\ndwhich {run.in_progress_count} are in progress and {run.not_started_count} have not_\ndstarted')
Run id: vsc2be500dcuhc1q5bhvq7kdwoqh367i has 97 active devices in the org of which 45_\ndare 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 = huoobhistdtxxpzhmg52yns7wmsuvjyx, Run Status = ACTIVE, Run create time = 2022-  
↳01-19T21:00:00.000Z, Results Returned = 2333, Template Id =  
↳p7qtvxms0oaju46whcrfmyppa9fiqpn9  
Run id = bdygnd8jvpjddqjmatdsuqzopaxebquqb, 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()	<b>CREATE</b> , org.liveresponse.session	<b>READ</b> CREATE is needed to start the LR session and READ is needed to check the status of the command
Close session lr_session.close()	<b>READ</b> , org.liveresponse.session	<b>DELETE</b> 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(...)	<b>READ</b> org.liveresponse.file	
Get File lr_session.get_file(...)	<b>READ</b> org.liveresponse.file	
Upload File lr_session.put_file(...)	<b>CREATE</b> , org.liveresponse.file	<b>READ</b> CREATE is needed to upload the file and READ is needed to check the status of the command
Delete file lr_session.delete_file(...)	<b>READ</b> , org.liveresponse.file	<b>DELETE</b> DELETE is needed to delete the file and READ is needed to check the status of the command
List Directory lr_session.list_directory(...)	<b>READ</b> org.liveresponse.file	
Create Directory lr_session.create_directory(...)	<b>CREATE</b> , org.liveresponse.file	<b>READ</b> CREATE is needed to create the directory and READ is needed to check the status of the command
Walk Directory lr_session.walk(...)	<b>READ</b> org.liveresponse.file	
Kill Process lr_session.kill_process(...)	<b>READ</b> , org.liveresponse.process	<b>DELETE</b> DELETE is needed to kill the process and READ is needed to check the status of the command
Create Process lr_session.create_process(...)	<b>EXECUTE</b> org.liveresponse.process OR <b>EXECUTE</b> org.liveresponse.process	If wait_for_completion = False, wait_for_output = False only EXECUTE is needed. Otherwise also file permissions are needed.
62	<b>READ, DELETE</b> org.liveresponse.file	<b>Chapter 4. Getting Started</b>
	<b>READ</b> org.liveresponse.process	

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 CBCCloudAPI
>>> from cbc_sdk.enterprise_edr import IOC_V2
>>> cbcsdk = CBCCloudAPI(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_ip4`) cannot support CIDR notation; full IP addresses must be used.
- equality IOCs for IPv6 fields (e.g. `netconn_remote_ip6`) 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-0febda35fcdf2dbd1
        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-0febda35fcdf2dbd1
        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: 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-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-0febda35fcaf2dbd1","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.
- *Devices* - Search for, get information about, and act on endpoints.
- *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 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
<code>blocked_threat_category</code>	CB Analytics
<code>category</code>	All
<code>count</code>	Watchlist
<code>document_guid</code>	Watchlist
<code>group_details</code>	All
<code>kill_chain_status</code>	CB Analytics
<code>not_blocked_threat_category</code>	CB Analytics
<code>target_value</code>	Container Runtime
<code>threat_activity_dlp</code>	CB Analytics
<code>threat_activity_phish</code>	CB Analytics
<code>threat_cause_threat_category</code>	All
<code>threat_cause_vector</code>	All
<code>threat_indicators</code>	Watchlist
<code>workload_id</code>	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(CBAalyticsAlert)
>>> 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": [
      {
        "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()))
```

(continues on next page)

(continued from previous page)

```
>>> 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
<code>cluster_name</code>	<code>k8s_cluster</code>
<code>create_time</code>	<code>backend_timestamp</code>
<code>first_event_time</code>	<code>first_event_timestamp</code>
<code>last_event_time</code>	<code>last_event_timestamp</code>
<code>last_update_time</code>	<code>backend_update_timestamp</code>
<code>namespace</code>	<code>k8s_namespace</code>
<code>notes_present</code>	<code>alert_notes_present</code>

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
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
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.



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 <a href="#">Observations</a>
<code>cbapi.psc.defense Policy</code>	<code>cbc_sdk.platform Policy</code>	<a href="#">IntegrationServices Policy v3 API Migration</a>
<code>cbc_sdk.endpoint_standard EnrichedEvent</code>	<code>cbc_sdk.platform Observation</code>	Enriched Events will remain available until July 2024. <a href="#">Enriched Events API Migration</a>
<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>	<a href="#">Notification Migration</a>

## 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.cbc_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>cbapi.cbc_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 recommendedation 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

The model and query classes for supporting alerts and alert workflows.

*Alerts* indicate suspicious behavior and known threats in the monitored environment. They should be regularly reviewed to determine whether action must be taken or policies should be modified. The Carbon Black Cloud Python SDK may be used to retrieve alerts, as well as manage the workflow by modifying alert status or closing alerts.

The Carbon Black Cloud Python SDK currently implements the Alerts v7 API, as documented on [the Developer Network](#). It works with any Carbon Black Cloud product, although certain alert types are only generated by specific products.

Typical usage example:

```
# assume "cb" is an instance of CBCloudAPI
query = cb.select(Alert).add_criteria("device_os", ["WINDOWS"]).set_minimum_severity(3)
query.set_time_range(range="-1d").set_rows(1000).add_exclusions("type", ["WATCHLIST"])
for alert in query:
    print(f"Alert ID {alert.id} with severity {alert.severity} at {alert.detection_
    ↳timestamp}")
```

**class Alert**(*cb, model\_unique\_id, initial\_data=None*)

Bases: *PlatformModel*

Represents a basic alert within the Carbon Black Cloud.

Alert objects are typically located through a search (using `AlertSearchQuery`) before they can be operated on.

The complete list of alert fields is too large to be reproduced here; please see the list of available fields for each alert type on [the Developer Network](#).

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 placed on 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*) – True if the note is a threat note, False if the note is an alert note.``
- **initial\_data** (*dict*) – Initial data used to populate the note.

**delete()**

Deletes a note from an alert.

#### Required Permissions:

org.alerts.notes (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

**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.

---

**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.
- 

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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 alert workflow action.

**Return type**

*Job*



### 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()
```

**create\_note**(note, threat\_note=False)

Creates a new note for this alert.

**Required Permissions:**

org.alerts.notes (CREATE)

**Parameters**

- **note** (str) – Note content to add.
- **threat\_note** (bool) – True to add this alert to the treat, False to add this note to the alert.

**Returns**

The newly-added note.

**Return type**

*Note*

**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.

**Required Permissions:**

org.alerts.dismiss (EXECUTE)

**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 ``Note``s added and workflow state changes.

**Required Permissions:**

org.alerts (READ)

**Parameters**

**threat** (*bool*) – If True, the threat history is returned; if False, the alert history is returned.

**Returns**

The ``dict``s of each determination, note or workflow change.

**Return type**

list

**get\_observations**(*timeout=0*)

Requests observations that are associated with the Alert.

Uses `Observation.bulk_get_details`.

**Required Permissions:**

org.search.events (READ, CREATE)

**Returns**

Observations associated with the Alert.

**Return type**list[*Observation*]**get\_process**(*async\_mode=False*)

Gets the process corresponding with the alert.

**Required Permissions:**

org.search.events (CREATE, READ)

**Parameters**

**async\_mode** – True to request process in an asynchronous manner.

**Returns**

The process corresponding to the alert.

**Return type***Process*

---

**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\_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 this alert.

**Required Permissions:**

org.alerts.notes (READ)

**Parameters**

**threat\_note** (*bool*) – True to retrieve threat notes, False to retrieve alert notes.

**Returns**

The list of notes for the alert.

**Return type**list[*Note*]**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.

**Required Permissions:**

org.alerts (READ)

**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.

---

**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.
- 

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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 alert workflow action.

**Return type***Job***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()
```

**update\_threat**(*remediation=None, comment=None*)

Updates all future alerts assigned to the threat\_id to the OPEN state.

**Required Permissions:**

org.alerts.dismiss (EXECUTE)

**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*

Query object that is used to locate Alert objects.

The AlertSearchQuery 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 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.

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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.

**Required Permissions:**

org.alerts (READ)

**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**

*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*)

Converts the AlertSearchQuery to a GroupAlertSearchQuery grouped by the argument.

**Parameters**

**field** (*string*) – The field to group by, defaults to “threat\_id.”

**Returns**

New query instance.

**Return type***GroupedAlertSearchQuery*

---

**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**

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(addr)**

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**

**addr** (*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.

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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_PROGESS", "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** *CBAalyticsAlert*(*cb, model\_unique\_id, initial\_data=None*)

Bases: *Alert*

A specialization of the base *Alert* class that represents a CB Analytics alert.

The complete list of alert fields is too large to be reproduced here; please see the list of available fields for each alert type on [the Developer Network](#).

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 placed on 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*) – True if the note is a threat note, False if the note is an alert note.
- **initial\_data** (*dict*) – Initial data used to populate the note.

**delete()**

Deletes a note from an alert.

**Required Permissions:**

org.alerts.notes (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

**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.

---

**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.

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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 alert workflow action.

**Return type**

*Job*

**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()
```

**create\_note**(*note*, *threat\_note=False*)

Creates a new note for this alert.

**Required Permissions:**

org.alerts.notes (CREATE)

**Parameters**

- **note** (*str*) – Note content to add.
- **threat\_note** (*bool*) – True to add this alert to the treat, False to add this note to the alert.

**Returns**

The newly-added note.

**Return type**

*Note*

**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.

##### **Required Permissions:**

org.alerts.dismiss (EXECUTE)

##### **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 ``Note``s added and workflow state changes.

**Required Permissions:**

org.alerts (READ)

**Parameters**

**threat** (*bool*) – If True, the threat history is returned; if False, the alert history is returned.

**Returns**

The ``dict``s of each determination, note or workflow change.

**Return type**

list

**get\_observations**(*timeout=0*)

Requests observations that are associated with the Alert.

Uses `Observation.bulk_get_details`.

**Required Permissions:**

org.search.events (READ, CREATE)

**Returns**

Observations associated with the Alert.

**Return type**

list[*Observation*]

**get\_process**(*async\_mode=False*)

Gets the process corresponding with the alert.

**Required Permissions:**

org.search.events (CREATE, READ)

**Parameters**

**async\_mode** – True to request process in an asynchronous manner.

**Returns**

The process corresponding to the alert.

**Return type**

*Process*

---

**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\_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 this alert.

**Required Permissions:**

org.alerts.notes (READ)

**Parameters**

**threat\_note** (*bool*) – True to retrieve threat notes, False to retrieve alert notes.

**Returns**

The list of notes for the alert.

**Return type**

list[*Note*]

**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.

**Required Permissions:**

org.alerts (READ)

**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.

**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.

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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 alert workflow action.

**Return type**

*Job*

**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()
```

**update\_threat**(*remediation=None, comment=None*)

Updates all future alerts assigned to the `threat_id` to the OPEN state.

**Required Permissions:**

`org.alerts.dismiss` (EXECUTE)

**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](#)

A specialization of the base `Alert` class that represents a Container Runtime alert.

The complete list of alert fields is too large to be reproduced here; please see the list of available fields for each alert type on [the Developer Network](#).

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 placed on 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*) – True if the note is a threat note, False if the note is an alert note.
- **initial\_data** (*dict*) – Initial data used to populate the note.

**delete()**

Deletes a note from an alert.

**Required Permissions:**

org.alerts.notes (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

**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.

---

**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.
- 

**Required Permissions:**

`org.alerts.close (EXECUTE)`, `jobs.status (READ)`

**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 alert workflow action.

**Return type**

*Job*

**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()
```

**create\_note**(*note, threat\_note=False*)

Creates a new note for this alert.

**Required Permissions:**

`org.alerts.notes (CREATE)`

**Parameters**

- **note** (*str*) – Note content to add.
- **threat\_note** (*bool*) – True to add this alert to the treat, False to add this note to the alert.

**Returns**

The newly-added note.

**Return type**

*Note*

**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.

**Required Permissions:**

org.alerts.dismiss (EXECUTE)

**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 ``Note``s added and workflow state changes.

**Required Permissions:**

org.alerts (READ)

**Parameters**

**threat** (*bool*) – If True, the threat history is returned; if False, the alert history is returned.

**Returns**

The ``dict``s of each determination, note or workflow change.

**Return type**

list

**get\_observations**(*timeout=0*)

Requests observations that are associated with the Alert.

Uses `Observation.bulk_get_details`.

**Required Permissions:**

org.search.events (READ, CREATE)

**Returns**

Observations associated with the Alert.

**Return type**

list[*Observation*]

**get\_process**(*async\_mode=False*)

Gets the process corresponding with the alert.

**Required Permissions:**

org.search.events (CREATE, READ)

**Parameters**

**async\_mode** – True to request process in an asynchronous manner.

**Returns**

The process corresponding to the alert.

**Return type**

*Process*

---

**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\_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 this alert.

**Required Permissions:**

org.alerts.notes (READ)

**Parameters**

**threat\_note** (*bool*) – True to retrieve threat notes, False to retrieve alert notes.

**Returns**

The list of notes for the alert.

**Return type**

list[[Note](#)]

**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.

**Required Permissions:**

org.alerts (READ)

**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.

---

**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.
- 

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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 alert workflow action.

**Return type***Job***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()
```

**update\_threat**(*remediation*=None, *comment*=None)

Updates all future alerts assigned to the threat\_id to the OPEN state.

**Required Permissions:**

org.alerts.dismiss (EXECUTE)

**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](#)

A specialization of the base [Alert](#) class that represents a Device Control alert.

The complete list of alert fields is too large to be reproduced here; please see the list of available fields for each alert type on the [Developer Network](#).

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 placed on 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*) – True if the note is a threat note, False if the note is an alert note.
- **initial\_data** (*dict*) – Initial data used to populate the note.

**delete()**

Deletes a note from an alert.

**Required Permissions:**

org.alerts.notes (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

**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.

---

**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.

---

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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 alert workflow action.

**Return type**

*Job*

**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()
```

**create\_note**(*note*, *threat\_note=False*)

Creates a new note for this alert.

**Required Permissions:**

org.alerts.notes (CREATE)

**Parameters**

- **note** (*str*) – Note content to add.
- **threat\_note** (*bool*) – True to add this alert to the treat, False to add this note to the alert.

**Returns**

The newly-added note.

**Return type**

*Note*

**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.

**Required Permissions:**

org.alerts.dismiss (EXECUTE)

**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 ``Note``s added and workflow state changes.

**Required Permissions:**

org.alerts (READ)

**Parameters**

**threat** (*bool*) – If True, the threat history is returned; if False, the alert history is returned.

**Returns**

The ``dict``s of each determination, note or workflow change.

**Return type**

list

**get\_observations**(*timeout=0*)

Requests observations that are associated with the Alert.

Uses `Observation.bulk_get_details`.

**Required Permissions:**

org.search.events (READ, CREATE)

**Returns**

Observations associated with the Alert.

**Return type**

list[*Observation*]

**get\_process**(*async\_mode=False*)

Gets the process corresponding with the alert.

**Required Permissions:**

org.search.events (CREATE, READ)

**Parameters**

**async\_mode** – True to request process in an asynchronous manner.

**Returns**

The process corresponding to the alert.

**Return type**

*Process*

---

**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\_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 this alert.

**Required Permissions:**

org.alerts.notes (READ)

**Parameters**

**threat\_note** (*bool*) – True to retrieve threat notes, False to retrieve alert notes.

**Returns**

The list of notes for the alert.

**Return type**

list[[Note](#)]

**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.

**Required Permissions:**

org.alerts (READ)

**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.

---

**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.
- 

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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 alert workflow action.

**Return type**

*Job*

**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()
```

**update\_threat**(*remediation=None, comment=None*)

Updates all future alerts assigned to the threat\_id to the OPEN state.

**Required Permissions:**

org.alerts.dismiss (EXECUTE)

**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 alerts that have been grouped together based on a common characteristic.

This allows viewing of similar alerts across multiple endpoints.

#### 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*

Query object that is used to locate Alert objects.

This query is constructed by using the `select()` method on `CBCloudAPI` to create an `AlertSearchQuery`, then using that query's `set_group_by()` method to specify grouping.

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.

#### Required Permissions:

org.alerts (READ)

**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 calculation. When False (default), a filter on the term is ignored while calculating facets.

**Returns**

A list of facet information specified as ``dict``s.

**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**

New query instance.

**Return type**

*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***CBAalyticsAlertSearchQuery***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**

CBAanalyticsAlertSearchQuery

**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**

CBAnalyticsAlertSearchQuery

**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_PROGESS", "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*

A specialization of the base `Alert` class that represents a host-based firewall alert.

The complete list of alert fields is too large to be reproduced here; please see the list of available fields for each alert type on [the Developer Network](#).

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 placed on 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*) – True if the note is a threat note, False if the note is an alert note.
- **initial\_data** (*dict*) – Initial data used to populate the note.

#### delete()

Deletes a note from an alert.

#### Required Permissions:

org.alerts.notes (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

#### 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.

---

**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.
- 

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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 alert workflow action.

**Return type**

*Job*

**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()
```

**create\_note**(*note, threat\_note=False*)

Creates a new note for this alert.

**Required Permissions:**

org.alerts.notes (CREATE)

**Parameters**

- **note** (*str*) – Note content to add.
- **threat\_note** (*bool*) – True to add this alert to the treat, False to add this note to the alert.

**Returns**

The newly-added note.



**Return type***Note***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.

**Required Permissions:**

org.alerts.dismiss (EXECUTE)

**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 ``Note``s added and workflow state changes.

**Required Permissions:**

org.alerts (READ)

**Parameters**

**threat** (*bool*) – If True, the threat history is returned; if False, the alert history is returned.

**Returns**

The ``dict``s of each determination, note or workflow change.

**Return type**

list

**get\_observations**(*timeout=0*)

Requests observations that are associated with the Alert.

Uses `Observation.bulk_get_details`.

**Required Permissions:**

org.search.events (READ, CREATE)

**Returns**

Observations associated with the Alert.

**Return type**

list[*Observation*]

**get\_process**(*async\_mode=False*)

Gets the process corresponding with the alert.

**Required Permissions:**

org.search.events (CREATE, READ)

**Parameters**

**async\_mode** – True to request process in an asynchronous manner.

**Returns**

The process corresponding to the alert.

**Return type**

*Process*

---

**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\_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 this alert.

**Required Permissions:**

org.alerts.notes (READ)

**Parameters**

**threat\_note** (*bool*) – True to retrieve threat notes, False to retrieve alert notes.

**Returns**

The list of notes for the alert.

**Return type**

list[[Note](#)]

**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.

**Required Permissions:**

org.alerts (READ)

**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*.

---

**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.

---

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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 alert workflow action.

**Return type**

*Job*

**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()
```

**update\_threat**(*remediation=None*, *comment=None*)

Updates all future alerts assigned to the *threat\_id* to the OPEN state.

**Required Permissions:**

org.alerts.dismiss (EXECUTE)

**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](#)

A specialization of the base [Alert](#) class that represents an intrusion detection system alert.

The complete list of alert fields is too large to be reproduced here; please see the list of available fields for each alert type on [the Developer Network](#).

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 placed on 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*) – True if the note is a threat note, False if the note is an alert note.``

- **initial\_data** (*dict*) – Initial data used to populate the note.

**delete()**

Deletes a note from an alert.

**Required Permissions:**

org.alerts.notes (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

**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.

---

**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.
- 

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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 alert workflow action.

**Return type**

*Job*

**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()
```

**create\_note**(*note*, *threat\_note=False*)

Creates a new note for this alert.

**Required Permissions:**

org.alerts.notes (CREATE)

**Parameters**

- **note** (*str*) – Note content to add.
- **threat\_note** (*bool*) – True to add this alert to the treat, False to add this note to the alert.

**Returns**

The newly-added note.

**Return type**

*Note*

**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.

**Required Permissions:**

org.alerts.dismiss (EXECUTE)

**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 ``Note``s added and workflow state changes.

**Required Permissions:**

org.alerts (READ)

**Parameters**

**threat** (*bool*) – If True, the threat history is returned; if False, the alert history is returned.



**Returns**

The ``dict``s of each determination, note or workflow change.

**Return type**

list

**get\_network\_threat\_metadata()**

Retrun 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 `Observation.bulk_get_details`.

**Required Permissions:**

org.search.events (READ, CREATE)

**Returns**

Observations associated with the Alert.

**Return type**

list[*Observation*]

**get\_process(async\_mode=False)**

Gets the process corresponding with the alert.

**Required Permissions:**

org.search.events (CREATE, READ)

**Parameters**

**async\_mode** – True to request process in an asynchronous manner.

**Returns**

The process corresponding to the alert.

**Return type**

*Process*

---

**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\_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 this alert.

**Required Permissions:**

org.alerts.notes (READ)

**Parameters**

**threat\_note** (*bool*) – True to retrieve threat notes, False to retrieve alert notes.

**Returns**

The list of notes for the alert.

**Return type**

list[*Note*]

**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.

**Required Permissions:**

org.alerts (READ)

**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.

---

**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.
- 

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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 alert workflow action.

**Return type***Job***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()
```

**update\_threat**(*remediation*=None, *comment*=None)

Updates all future alerts assigned to the threat\_id to the OPEN state.

**Required Permissions:**

org.alerts.dismiss (EXECUTE)

**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](#)

A specialization of the base `Alert` class that represents a watchlist alert.

The complete list of alert fields is too large to be reproduced here; please see the list of available fields for each alert type on [the Developer Network](#).

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 placed on 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*) – True if the note is a threat note, False if the note is an alert note.
- **initial\_data** (*dict*) – Initial data used to populate the note.

**delete()**

Deletes a note from an alert.

**Required Permissions:**

org.alerts.notes (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

**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.

---

**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.
- 

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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 alert workflow action.

**Return type**

*Job*

**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()
```

**create\_note**(*note*, *threat\_note=False*)

Creates a new note for this alert.

**Required Permissions:**

org.alerts.notes (CREATE)

**Parameters**

- **note** (*str*) – Note content to add.
- **threat\_note** (*bool*) – True to add this alert to the treat, False to add this note to the alert.

**Returns**

The newly-added note.

**Return type**

*Note*

**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.

**Required Permissions:**

org.alerts.dismiss (EXECUTE)

**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 ``Note``s added and workflow state changes.

**Required Permissions:**

org.alerts (READ)

**Parameters**

**threat** (*bool*) – If True, the threat history is returned; if False, the alert history is returned.

**Returns**

The ``dict``s of each determination, note or workflow change.

**Return type**

list

**get\_observations**(*timeout=0*)

Requests observations that are associated with the Alert.

Uses `Observation.bulk_get_details`.

**Required Permissions:**

org.search.events (READ, CREATE)

**Returns**

Observations associated with the Alert.

**Return type**

list[*Observation*]

**get\_process**(*async\_mode=False*)

Gets the process corresponding with the alert.

**Required Permissions:**

org.search.events (CREATE, READ)

**Parameters**

**async\_mode** – True to request process in an asynchronous manner.

**Returns**

The process corresponding to the alert.

**Return type**

*Process*

---

**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\_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 this alert.

#### Required Permissions:

org.alerts.notes (READ)

#### Parameters

**threat\_note** (*bool*) – True to retrieve threat notes, False to retrieve alert notes.

#### Returns

The list of notes for the alert.

#### Return type

list[[Note](#)]

#### `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.

#### Required Permissions:

org.alerts (READ)

#### 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.

---

**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.
- 

**Required Permissions:**

org.alerts.close (EXECUTE), jobs.status (READ)

**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 alert workflow action.

**Return type***Job***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()
```

**update\_threat**(*remediation*=None, *comment*=None)

Updates all future alerts assigned to the threat\_id to the OPEN state.

**Required Permissions:**

org.alerts.dismiss (EXECUTE)

**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”, “IN-ACTIVE”, “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=
"WNEXFKQ7-00050603-00000066c-000000000-1d6c9acb43e29bb"))
>>> 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 facets**

Returns 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**(*timeout=0*)

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)

#### Parameters

**timeout** (*int*) – The timeout for this wait in milliseconds. If this is 0, the default value will be used.

#### 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**

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\_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**

CBAalyticsAlertSearchQuery

**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(addr)**

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**

CBAnalyticsAlertSearchQuery

**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

**Required Permissions:**

org.search.events (READ, CREATE)

**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 “MIS-

SION\_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”, “TERMI-  
NATE\_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”, “RAN-  
SOM”, “NETWORK”, “PROCESS\_ISOLATION”, “CODE\_INJECTION”, “MEM-  
ORY\_SCRAPE”, “RUN\_INMEMORY\_CODE”, “ESCALATE”, “RUN”]

“required”: [True, False]

**property bypass\_rule\_configs**

Returns a dictionary of bypass rule configuration IDs and objects for this Policy.

**Returns**

A dictionary with bypass rule configuration IDs as keys and `BypassRuleConfig` objects as values.

**Return type**

dict

**property bypass\_rule\_configs\_list**

Returns a list of bypass rule configuration objects for this Policy.

**Returns**

A list of `BypassRuleConfig` objects.

**Return type**

list

**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
```

(continues on next page)

(continued from previous page)

```
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 BypassRuleConfig**(*cb, parent, model\_unique\_id=None, initial\_data=None, force\_init=False, full\_doc=False*)

Bases: *PolicyRuleConfig*

Represents a bypass 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 BypassRuleConfig, change the values of its property fields, then call its `save()` method. This requires the `org.policies(UPDATE)` permission.

To delete an existing BypassRuleConfig, 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
- **exclusions** – The exclusions associated with this rule config

Initialize the BypassRuleConfig 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)**

Not Supported

**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**

Not Supported

**refresh()**

Reload this object from the server.

**replace\_exclusions(exclusions)**

Replaces all the exclusions for a bypass rule configuration

**Parameters**

**exclusions** (*dict*) – The entire exclusion set to be replaced

**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)**

Not Supported

**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 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
- **exclusions** – The exclusions 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.



**replace\_exclusions**(*exclusions*)

Replaces all the exclusions for a bypass rule configuration

**Parameters**

**exclusions** (*dict*) – The entire exclusion set to be replaced

**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 `DataCollectionRuleConfig`, 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
- **exclusions** – The exclusions associated with this rule config

Initialize the `DataCollectionRuleConfig` 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
- **exclusions** – The exclusions associated with this rule config

Initialize the HostBasedFirewallRuleConfig 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
- **exclusions** – The exclusions 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](#).

Processes have many fields, too many to list here; for a complete list of available fields, visit [the Search Fields page](#) on the Carbon Black Developer Network, and filter on the PROCESS route.

**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, "WNEXFKQ7-00050603-00000066c-000000000-  
↪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 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.

**not\_**(*q*, \*\*kwargs)

Adds a negative 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.

**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"'))
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])
{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'',
 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,
             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 *cb\_lr\_base*

**cb\_lr\_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.

**class BackoffHandler**(*cb*, *timeout*=0, *initial*=0.1, *multiplier*=2.0, *threshold*=2.0)

Bases: object

Logic for handling exponential backoff of multiple communications requests.

The logic also handles timeouts of operations that go on too long.

Example:

```
backoff = BackoffHandler(timeout=600000) # 10 minutes = 600 seconds
with backoff as b:
    while operation_continues():
        b.pause()
        do_operation()
```

Initialize the BackoffHandler.

**Parameters**



- **cb** (*BaseAPI*) – The API object for the operation.
- **timeout** (*int*) – The timeout for the operation, in milliseconds. If this is 0, the default timeout as configured in the credentials will be used. The default is 0.
- **initial** (*float*) – The initial value for the exponential backoff pause, in seconds. The default is 0.1.
- **multiplier** (*float*) – The value by which the exponential backoff pause will be multiplied each time a pause happens. The default is 2.0.
- **threshold** (*float*) – The maximum value for the exponential backoff pause, in seconds. The default is 2.0.

**class BackoffOperation**(*timeout, initial, multiplier, threshold*)

Bases: `object`

Handler for a single operation requiring exponential backoff between communication attempts.

This is returned by `BackoffHandler` as part of the `with` operation, and is stored in the variable referred to in its `as` clause.

Initialize the `BackoffOperation`.

#### Parameters

- **timeout** (*int*) – The timeout for the operation, in milliseconds.
- **initial** (*float*) – The initial value for the exponential backoff pause, in seconds.
- **multiplier** (*float*) – The value by which the exponential backoff pause will be multiplied each time a pause happens.
- **threshold** (*float*) – The maximum value for the exponential backoff pause, in seconds.

#### `pause()`

Pauses operation for a determined amount of time.

The method also checks for a timeout and raises `TimeoutError` if it happens, and computes the amount of time to pause the next time this method is called.

#### Raises

**`TimeoutError`** – If the timeout value is reached.

#### `reset(full=False)`

Resets the state of the operation so that the pause time is reset.

Does not affect the timeout value. This should be used, for instance, after a successful operation to minimize the pause before the next operation is started.

#### Parameters

**full** (*bool*) – If this is `True`, the next pause time will be reset to 0. If this is `False`, the next pause time will be reset to the initial pause time.

#### property `timeout`

Returns the current timeout associated with this handler, in milliseconds.

#### `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`, 131
- `cbc_sdk.audit_remediation.differential`, 167
- `cbc_sdk.base`, 549
- `cbc_sdk.cache.lru`, 546
- `cbc_sdk.connection`, 572
- `cbc_sdk.credential_providers.aws_sm_credential_provider`, 173
- `cbc_sdk.credential_providers.default`, 172
- `cbc_sdk.credential_providers.envron_credential_provider`, 173
- `cbc_sdk.credential_providers.file_credential_provider`, 174
- `cbc_sdk.credential_providers.keychain_credential_provider`, 174
- `cbc_sdk.credential_providers.registry_credential_provider`, 175
- `cbc_sdk.credentials`, 582
- `cbc_sdk.endpoint_standard.base`, 176
- `cbc_sdk.endpoint_standard.recommendation`, 184
- `cbc_sdk.endpoint_standard.usb_device_control`, 193
- `cbc_sdk.enterprise_edr.auth_events`, 208
- `cbc_sdk.enterprise_edr.threat_intelligence`, 220
- `cbc_sdk.enterprise_edr.ubs`, 246
- `cbc_sdk.errors`, 584
- `cbc_sdk.helpers`, 589
- `cbc_sdk.live_response_api`, 590
- `cbc_sdk.platform.alerts`, 250
- `cbc_sdk.platform.asset_groups`, 343
- `cbc_sdk.platform.audit`, 354
- `cbc_sdk.platform.base`, 249
- `cbc_sdk.platform.devices`, 355
- `cbc_sdk.platform.events`, 374
- `cbc_sdk.platform.grants`, 387
- `cbc_sdk.platform.jobs`, 395
- `cbc_sdk.platform.legacy_alerts`, 399
- `cbc_sdk.platform.network_threat_metadata`, 412
- `cbc_sdk.platform.observations`, 414
- `cbc_sdk.platform.policies`, 424
- `cbc_sdk.platform.policy_ruleconfigs`, 441
- `cbc_sdk.platform.previewer`, 457
- `cbc_sdk.platform.processes`, 458
- `cbc_sdk.platform.reputation`, 472
- `cbc_sdk.platform.users`, 477
- `cbc_sdk.platform.vulnerability_assessment`, 484
- `cbc_sdk.utils`, 612
- `cbc_sdk.winerror`, 614
- `cbc_sdk.workload.nsx_remediation`, 507
- `cbc_sdk.workload.sensor_lifecycle`, 509
- `cbc_sdk.workload.vm_workloads_search`, 513



## A

- accept() (*Recommendation method*), 188
- add\_criteria() (*AffectedAssetQuery method*), 484
- add\_criteria() (*AlertSearchQuery method*), 257
- add\_criteria() (*AssetGroupQuery method*), 351
- add\_criteria() (*AsyncProcessQuery method*), 458
- add\_criteria() (*AuthEventQuery method*), 214
- add\_criteria() (*AWSComputeResourceQuery method*), 514
- add\_criteria() (*BaseComputeResourceQuery method*), 526
- add\_criteria() (*CriteriaBuilderSupportMixin method*), 550
- add\_criteria() (*DeviceSearchQuery method*), 365
- add\_criteria() (*DifferentialQuery method*), 169
- add\_criteria() (*EnrichedEventQuery method*), 179
- add\_criteria() (*EventFacetQuery method*), 377
- add\_criteria() (*EventQuery method*), 382
- add\_criteria() (*FacetQuery method*), 134, 552
- add\_criteria() (*GroupedAlertSearchQuery method*), 301
- add\_criteria() (*ObservationQuery method*), 419
- add\_criteria() (*Query method*), 562
- add\_criteria() (*RecommendationQuery method*), 190
- add\_criteria() (*ResultQuery method*), 143
- add\_criteria() (*RunHistoryQuery method*), 154
- add\_criteria() (*SensorKitQuery method*), 510
- add\_criteria() (*TemplateHistoryQuery method*), 165
- add\_criteria() (*USBDeviceApprovalQuery method*), 198
- add\_criteria() (*USBDeviceQuery method*), 203
- add\_criteria() (*VCenterComputeResourceQuery method*), 533
- add\_criteria() (*VulnerabilityAssetViewQuery method*), 494
- add\_criteria() (*VulnerabilityQuery method*), 501
- add\_descriptions() (*PolicyQuery method*), 437
- add\_directory\_action\_rule() (*Policy.PolicyBuilder method*), 426
- add\_exclusions() (*AlertSearchQuery method*), 258
- add\_exclusions() (*AsyncProcessQuery method*), 459
- add\_exclusions() (*AuthEventQuery method*), 214
- add\_exclusions() (*EnrichedEventQuery method*), 180
- add\_exclusions() (*EventFacetQuery method*), 378
- add\_exclusions() (*EventQuery method*), 383
- add\_exclusions() (*ExclusionBuilderSupportMixin method*), 551
- add\_exclusions() (*FacetQuery method*), 553
- add\_exclusions() (*GroupedAlertSearchQuery method*), 302
- add\_exclusions() (*ObservationQuery method*), 419
- add\_exclusions() (*Query method*), 563
- add\_facet\_field() (*EventFacetQuery method*), 378
- add\_facet\_field() (*FacetQuery method*), 553
- add\_grant\_profile() (*User.UserBuilder method*), 478
- add\_headers() (*CBCSDKSessionAdapter method*), 577
- add\_ioc() (*Report.ReportBuilder method*), 232
- add\_members() (*AssetGroup method*), 344
- add\_names() (*PolicyQuery method*), 437
- add\_note() (*ApiError method*), 584
- add\_note() (*ClientError method*), 584
- add\_note() (*ConnectionError method*), 584
- add\_note() (*CredentialError method*), 585
- add\_note() (*FunctionalityDecommissioned method*), 585
- add\_note() (*InvalidHashError method*), 585
- add\_note() (*InvalidObjectError method*), 585
- add\_note() (*LiveResponseError method*), 601
- add\_note() (*ModelNotFound method*), 586
- add\_note() (*MoreThanOneResultError method*), 586
- add\_note() (*NonQueryableModel method*), 587
- add\_note() (*NSXJobError method*), 586
- add\_note() (*ObjectNotFoundError method*), 587
- add\_note() (*OperationCancelled method*), 587
- add\_note() (*QuerySyntaxError method*), 587
- add\_note() (*ServerError method*), 588
- add\_note() (*TimeoutError method*), 588
- add\_note() (*UnauthorizedError method*), 589
- add\_org() (*Grant.ProfileBuilder method*), 391
- add\_policy\_ids() (*PolicyQuery method*), 438
- add\_principal() (*GrantQuery method*), 394
- add\_priorities() (*PolicyQuery method*), 438
- add\_profiles() (*User method*), 479
- add\_range() (*EventFacetQuery method*), 378

`add_range()` (*FacetQuery* method), 553  
`add_report_ids()` (*Watchlist* method), 241  
`add_report_ids()` (*Watchlist.WatchlistBuilder* method), 240  
`add_reports()` (*Feed.FeedBuilder* method), 220  
`add_reports()` (*Watchlist* method), 241  
`add_reports()` (*Watchlist.WatchlistBuilder* method), 240  
`add_role()` (*Grant.GrantBuilder* method), 388  
`add_role()` (*Grant.ProfileBuilder* method), 391  
`add_rule()` (*Policy* method), 431  
`add_rule()` (*Policy.PolicyBuilder* method), 426  
`add_rule_config()` (*Policy.PolicyBuilder* method), 426  
`add_rule_config_copy()` (*Policy.PolicyBuilder* method), 427  
`add_rule_copy()` (*Policy.PolicyBuilder* method), 427  
`add_sensor_kit_type()` (*SensorKitQuery* method), 511  
`add_sensor_setting()` (*Policy.PolicyBuilder* method), 427  
`add_tag()` (*Report.ReportBuilder* method), 232  
`add_threat_tags()` (*Alert* method), 251  
`add_threat_tags()` (*CBAAnalyticsAlert* method), 280  
`add_threat_tags()` (*ContainerRuntimeAlert* method), 287  
`add_threat_tags()` (*DeviceControlAlert* method), 294  
`add_threat_tags()` (*HostBasedFirewallAlert* method), 323  
`add_threat_tags()` (*IntrusionDetectionSystemAlert* method), 330  
`add_threat_tags()` (*WatchlistAlert* method), 337  
`add_time_criteria()` (*AlertSearchQuery* method), 258  
`add_time_criteria()` (*GroupedAlertSearchQuery* method), 302  
`add_to_groups()` (*Device* method), 357  
`add_to_groups_by_id()` (*Device* method), 357  
`AffectedAssetQuery` (class in *cbc\_sdk.platform.vulnerability\_assessment*), 484  
`aggregation()` (*EnrichedEventQuery* method), 180  
`Alert` (class in *cbc\_sdk.platform.alerts*), 250  
`Alert.Note` (class in *cbc\_sdk.platform.alerts*), 251  
`alert_search_suggestions()` (*CBCloudAPI* method), 122  
`AlertSearchQuery` (class in *cbc\_sdk.platform.alerts*), 257  
`all()` (*AffectedAssetQuery* method), 484  
`all()` (*AlertSearchQuery* method), 259  
`all()` (*AssetGroupQuery* method), 351  
`all()` (*AsyncProcessQuery* method), 459  
`all()` (*AuthEventQuery* method), 215  
`all()` (*AWSComputeResourceQuery* method), 515  
`all()` (*BaseComputeResourceQuery* method), 526  
`all()` (*DeviceSearchQuery* method), 365  
`all()` (*DifferentialQuery* method), 169  
`all()` (*EnrichedEventQuery* method), 180  
`all()` (*EventQuery* method), 383  
`all()` (*FacetQuery* method), 134  
`all()` (*FeedQuery* method), 225  
`all()` (*GrantQuery* method), 394  
`all()` (*GroupedAlertSearchQuery* method), 303  
`all()` (*IterableQueryMixin* method), 558  
`all()` (*JobQuery* method), 398  
`all()` (*ObservationQuery* method), 420  
`all()` (*PaginatedQuery* method), 561  
`all()` (*PolicyQuery* method), 438  
`all()` (*Query* method), 563  
`all()` (*RecommendationQuery* method), 190  
`all()` (*ReportQuery* method), 237  
`all()` (*ReputationOverrideQuery* method), 474  
`all()` (*ResultQuery* method), 143  
`all()` (*RunHistoryQuery* method), 154  
`all()` (*SensorKitQuery* method), 511  
`all()` (*SimpleQuery* method), 569  
`all()` (*TemplateHistoryQuery* method), 165  
`all()` (*USBDeviceApprovalQuery* method), 198  
`all()` (*USBDeviceBlockQuery* method), 203  
`all()` (*USBDeviceQuery* method), 204  
`all()` (*UserQuery* method), 482  
`all()` (*VCenterComputeResourceQuery* method), 533  
`all()` (*VulnerabilityAssetViewQuery* method), 494  
`all()` (*VulnerabilityQuery* method), 501  
`all()` (*WatchlistQuery* method), 244  
`allowed_orgs` (*Grant.Profile* property), 389  
`and_()` (*AffectedAssetQuery* method), 484  
`and_()` (*AlertSearchQuery* method), 259  
`and_()` (*AssetGroupQuery* method), 351  
`and_()` (*AsyncProcessQuery* method), 459  
`and_()` (*AuthEventQuery* method), 215  
`and_()` (*AWSComputeResourceQuery* method), 515  
`and_()` (*BaseComputeResourceQuery* method), 526  
`and_()` (*DeviceSearchQuery* method), 365  
`and_()` (*EnrichedEventQuery* method), 180  
`and_()` (*EventFacetQuery* method), 379  
`and_()` (*EventQuery* method), 383  
`and_()` (*FacetQuery* method), 134, 554  
`and_()` (*FeedQuery* method), 226  
`and_()` (*GroupedAlertSearchQuery* method), 303  
`and_()` (*ObservationQuery* method), 420  
`and_()` (*Query* method), 563  
`and_()` (*QueryBuilder* method), 567  
`and_()` (*QueryBuilderSupportMixin* method), 568  
`and_()` (*ReportQuery* method), 237  
`and_()` (*ReputationOverrideQuery* method), 475  
`and_()` (*ResultQuery* method), 143  
`and_()` (*RunHistoryQuery* method), 154



- `and_()` (*SimpleQuery* method), 569
  - `and_()` (*SummaryQuery* method), 470
  - `and_()` (*TemplateHistoryQuery* method), 165
  - `and_()` (*USBDeviceApprovalQuery* method), 198
  - `and_()` (*USBDeviceQuery* method), 204
  - `and_()` (*VCenterComputeResourceQuery* method), 533
  - `and_()` (*VulnerabilityAssetViewQuery* method), 494
  - `and_()` (*VulnerabilityQuery* method), 501
  - `and_()` (*WatchlistQuery* method), 245
  - `api_json_request()` (*BaseAPI* method), 573
  - `api_json_request()` (*CBCloudAPI* method), 122
  - `api_request_iterate()` (*BaseAPI* method), 573
  - `api_request_iterate()` (*CBCloudAPI* method), 122
  - `api_request_stream()` (*BaseAPI* method), 574
  - `api_request_stream()` (*CBCloudAPI* method), 123
  - `ApiError`, 584, 629
  - `append()` (*Vulnerability.AssetView* method), 491
  - `append_iocs()` (*Report* method), 233
  - `append_reports()` (*Feed* method), 221
  - `append_reports_rawdata()` (*Feed* method), 222
  - `append_rule()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup* method), 450
  - `append_rule_group()` (*HostBasedFirewallRuleConfig* method), 451
  - `application_` (*Recommendation.RecommendationNewRule* property), 187
  - `approve()` (*USBDevice* method), 193
  - `approve_process_sha256()` (*EnrichedEvent* method), 176
  - `approve_process_sha256()` (*Process* method), 465
  - `ArrayFieldDescriptor` (class in *cbc\_sdk.base*), 549
  - `asset_count` (*DevicePolicyChangePreview* property), 457
  - `asset_query` (*DevicePolicyChangePreview* property), 457
  - `AssetGroup` (class in *cbc\_sdk.platform.asset\_groups*), 343
  - `AssetGroupQuery` (class in *cbc\_sdk.platform.asset\_groups*), 350
  - `assets` (*DevicePolicyChangePreview* property), 457
  - `async_await_result()` (*NSXRemediationJob* method), 508
  - `async_export()` (*DifferentialQuery* method), 169
  - `async_export()` (*ResultQuery* method), 143
  - `ASYNC_RATE_LIMIT` (in *cbc\_sdk.audit\_remediation.differential*), 167
  - `AsyncProcessQuery` (class in *cbc\_sdk.platform.processes*), 458
  - `AsyncQueryMixin` (class in *cbc\_sdk.base*), 549
  - `audit_remediation()` (*CBCloudAPI* method), 123
  - `audit_remediation_history()` (*CBCloudAPI* method), 123
  - `AuditLog` (class in *cbc\_sdk.platform.audit*), 354
  - `AuthEvent` (class in *cbc\_sdk.enterprise\_edr.auth\_events*), 208
  - `AuthEventFacet` (class in *cbc\_sdk.enterprise\_edr.auth\_events*), 211
  - `AuthEventFacet.Ranges` (class in *cbc\_sdk.enterprise\_edr.auth\_events*), 211
  - `AuthEventFacet.Terms` (class in *cbc\_sdk.enterprise\_edr.auth\_events*), 212
  - `AuthEventGroup` (class in *cbc\_sdk.enterprise\_edr.auth\_events*), 213
  - `AuthEventQuery` (class in *cbc\_sdk.enterprise\_edr.auth\_events*), 214
  - `await_completion()` (*Job* method), 396
  - `await_result()` (*NSXRemediationJob* method), 508
  - `AWSComputeResource` (class in *cbc\_sdk.workload.vm\_workloads\_search*), 513
  - `AWSComputeResourceQuery` (class in *cbc\_sdk.workload.vm\_workloads\_search*), 514
  - `AWSCredentialProvider` (class in *cbc\_sdk.credential\_providers.aws\_sm\_credential\_provider*), 173
- ## B
- `background_scan()` (*Device* method), 357
  - `background_scan()` (*DeviceSearchQuery* method), 366
  - `BackoffHandler` (class in *cbc\_sdk.utils*), 612
  - `BackoffHandler.BackoffOperation` (class in *cbc\_sdk.utils*), 613
  - `ban_process_sha256()` (*EnrichedEvent* method), 176
  - `ban_process_sha256()` (*Process* method), 465
  - `BaseAPI` (class in *cbc\_sdk.connection*), 572
  - `BaseComputeResource` (class in *cbc\_sdk.workload.vm\_workloads\_search*), 524
  - `BaseComputeResourceQuery` (class in *cbc\_sdk.workload.vm\_workloads\_search*), 526
  - `BaseQuery` (class in *cbc\_sdk.base*), 549
  - `batch_size()` (*AsyncProcessQuery* method), 459
  - `batch_size()` (*AuthEventQuery* method), 215
  - `batch_size()` (*EnrichedEventQuery* method), 180
  - `batch_size()` (*EventQuery* method), 384
  - `batch_size()` (*ObservationQuery* method), 420
  - `batch_size()` (*PaginatedQuery* method), 561
  - `batch_size()` (*Query* method), 563
  - `Binary` (class in *cbc\_sdk.enterprise\_edr.ubs*), 246
  - `Binary.Summary` (class in *cbc\_sdk.enterprise\_edr.ubs*), 246
  - `BinaryFieldDescriptor` (class in *cbc\_sdk.base*), 550
  - `build()` (*Feed.FeedBuilder* method), 220
  - `build()` (*Grant.GrantBuilder* method), 388

`build()` (*Grant.ProfileBuilder* method), 391  
`build()` (*Policy.PolicyBuilder* method), 427  
`build()` (*Report.ReportBuilder* method), 232  
`build()` (*User.UserBuilder* method), 478  
`build()` (*Watchlist.WatchlistBuilder* method), 240  
`build_cli_parser()` (in module *cbc\_sdk.helpers*), 589  
`build_response()` (*CBCSDKSessionAdapter* method), 577  
`bulk_add_profiles()` (*User* class method), 479  
`bulk_create()` (*USBDeviceApproval* class method), 195  
`bulk_create()` (*USBDeviceBlock* class method), 201  
`bulk_create()` (*User* class method), 479  
`bulk_create_csv()` (*USBDeviceApproval* class method), 196  
`bulk_delete()` (*ReputationOverride* class method), 473  
`bulk_delete()` (*User* class method), 479  
`bulk_disable_all_access()` (*User* class method), 479  
`bulk_disable_profiles()` (*User* class method), 480  
`bulk_get_details()` (*AuthEvent* static method), 209  
`bulk_get_details()` (*Observation* static method), 414  
`bulk_install()` (*AWSComputeResource* class method), 513  
`bulk_install()` (*BaseComputeResource* class method), 524  
`bulk_install()` (*VCenterComputeResource* class method), 531  
`bulk_install_by_id()` (*AWSComputeResource* class method), 513  
`bulk_install_by_id()` (*BaseComputeResource* class method), 524  
`bulk_install_by_id()` (*VCenterComputeResource* class method), 532  
`bulk_threat_dismiss()` (*CBCloudAPI* method), 123  
`bulk_threat_update()` (*CBCloudAPI* method), 124  
`bypass()` (*Device* method), 357  
`bypass()` (*DeviceSearchQuery* method), 366  
`bypass_rule_configs` (*Policy* property), 431  
`bypass_rule_configs_list` (*Policy* property), 431  
`BypassRuleConfig` (class in *cbc\_sdk.platform.policy\_ruleconfigs*), 441

## C

`cancel_command()` (*CbLRSessionBase* method), 591  
`cancel_command()` (*LiveResponseSession* method), 603  
`CBAntalyticsAlert` (class in *cbc\_sdk.platform.alerts*), 279  
`CBAntalyticsAlert.Note` (class in *cbc\_sdk.platform.alerts*), 279  
`cbc_sdk.audit_remediation.base` module, 131  
`cbc_sdk.audit_remediation.differential` module, 167

`cbc_sdk.base` module, 549  
`cbc_sdk.cache.lru` module, 546  
`cbc_sdk.connection` module, 572  
`cbc_sdk.credential_providers.aws_sm_credential_provider` module, 173  
`cbc_sdk.credential_providers.default` module, 172  
`cbc_sdk.credential_providers.envron_credential_provider` module, 173  
`cbc_sdk.credential_providers.file_credential_provider` module, 174  
`cbc_sdk.credential_providers.keychain_credential_provider` module, 174  
`cbc_sdk.credential_providers.registry_credential_provider` module, 175  
`cbc_sdk.credentials` module, 582  
`cbc_sdk.endpoint_standard.base` module, 176  
`cbc_sdk.endpoint_standard.recommendation` module, 184  
`cbc_sdk.endpoint_standard.usb_device_control` module, 193  
`cbc_sdk.enterprise_edr.auth_events` module, 208  
`cbc_sdk.enterprise_edr.threat_intelligence` module, 220  
`cbc_sdk.enterprise_edr.ubs` module, 246  
`cbc_sdk.errors` module, 584  
`cbc_sdk.helpers` module, 589  
`cbc_sdk.live_response_api` module, 590  
`cbc_sdk.platform.alerts` module, 250  
`cbc_sdk.platform.asset_groups` module, 343  
`cbc_sdk.platform.audit` module, 354  
`cbc_sdk.platform.base` module, 249  
`cbc_sdk.platform.devices` module, 355  
`cbc_sdk.platform.events` module, 374  
`cbc_sdk.platform.grants` module, 387  
`cbc_sdk.platform.jobs` module, 395

- `cbc_sdk.platform.legacy_alerts`  
module, 399
- `cbc_sdk.platform.network_threat_metadata`  
module, 412
- `cbc_sdk.platform.observations`  
module, 414
- `cbc_sdk.platform.policies`  
module, 424
- `cbc_sdk.platform.policy_ruleconfigs`  
module, 441
- `cbc_sdk.platform.previewer`  
module, 457
- `cbc_sdk.platform.processes`  
module, 458
- `cbc_sdk.platform.reputation`  
module, 472
- `cbc_sdk.platform.users`  
module, 477
- `cbc_sdk.platform.vulnerability_assessment`  
module, 484
- `cbc_sdk.utils`  
module, 612
- `cbc_sdk.winerror`  
module, 614
- `cbc_sdk.workload.nsx_remediation`  
module, 507
- `cbc_sdk.workload.sensor_lifecycle`  
module, 509
- `cbc_sdk.workload.vm_workloads_search`  
module, 513
- `CBCloudAPI` (class in `cbc_sdk.rest_api`), 121
- `CBSDKSessionAdapter` (class in `cbc_sdk.connection`), 577
- `cbldr_session_cls` (*LiveResponseSessionManager* attribute), 610
- `CbLRManagerBase` (class in `cbc_sdk.live_response_api`), 590
- `CbLRSessionBase` (class in `cbc_sdk.live_response_api`), 591
- `CbMetaModel` (class in `cbc_sdk.base`), 550
- `cert_verify()` (*CBSDKSessionAdapter* method), 577
- `change_role()` (*User* method), 480
- `check_python_tls_compatibility()` (in module `cbc_sdk.connection`), 581
- `check_state_key()` (*SwaggerLoader* method), 571
- `children` (*Process* property), 466
- `classifier_` (*Watchlist* property), 241
- `clear()` (*Vulnerability.AssetView* method), 491
- `ClientError`, 584
- `close()` (*Alert* method), 252
- `close()` (*AlertSearchQuery* method), 259
- `close()` (*CBAnalyticsAlert* method), 280
- `close()` (*CBSDKSessionAdapter* method), 578
- `close()` (*CbLRSessionBase* method), 591
- `close()` (*ContainerRuntimeAlert* method), 287
- `close()` (*DeviceControlAlert* method), 294
- `close()` (*GroupedAlertSearchQuery* method), 303
- `close()` (*HostBasedFirewallAlert* method), 324
- `close()` (*IntrusionDetectionSystemAlert* method), 330
- `close()` (*LiveResponseSession* method), 603
- `close()` (*WatchlistAlert* method), 337
- `close_session()` (*CbLRManagerBase* method), 590
- `close_session()` (*LiveResponseSessionManager* method), 610
- `command_status()` (*CbLRSessionBase* method), 591
- `command_status()` (*LiveResponseSession* method), 603
- `CommDlgError` (class in `cbc_sdk.winerror`), 614
- `CompletionNotification` (class in `cbc_sdk.live_response_api`), 598
- `ComputeResourceFacet` (class in `cbc_sdk.workload.vm_workloads_search`), 530
- `ComputeResourceFacet.ComputeResourceFacetValue` (class in `cbc_sdk.workload.vm_workloads_search`), 530
- `config_params()` (*SensorKitQuery* method), 511
- `Connection` (class in `cbc_sdk.connection`), 579
- `ConnectionError`, 584
- `construct_include()` (in module `cbc_sdk.base`), 571
- `ContainerRuntimeAlert` (class in `cbc_sdk.platform.alerts`), 286
- `ContainerRuntimeAlert.Note` (class in `cbc_sdk.platform.alerts`), 286
- `convert_feed_query()` (*CBCloudAPI* method), 124
- `convert_from_cb()` (in module `cbc_sdk.utils`), 613
- `convert_to_cb()` (in module `cbc_sdk.utils`), 614
- `copy()` (*Vulnerability.AssetView* method), 491
- `copy_rules()` (*HostBasedFirewallRuleConfig* method), 452
- `core_prevention_rule_configs` (*Policy* property), 431
- `core_prevention_rule_configs_list` (*Policy* property), 432
- `CorePreventionRuleConfig` (class in `cbc_sdk.platform.policy_ruleconfigs`), 443
- `count()` (*Vulnerability.AssetView* method), 491
- `count_only()` (*DifferentialQuery* method), 170
- `CreatableModelMixin` (class in `cbc_sdk.base`), 550
- `create()` (*BaseAPI* method), 574
- `create()` (*CBCloudAPI* method), 124
- `create()` (*Feed* class method), 222
- `create()` (*Grant* class method), 392
- `create()` (*Policy* class method), 432
- `create()` (*Report* class method), 233
- `create()` (*ReputationOverride* class method), 473
- `create()` (*USBDeviceBlock* class method), 202
- `create()` (*User* class method), 480
- `create()` (*Watchlist* class method), 241

create\_directory() (*CbLRSessionBase* method), 591  
 create\_directory() (*LiveResponseSession* method), 603  
 create\_equality() (*IOC\_V2* class method), 228  
 create\_from\_feed() (*Watchlist* class method), 242  
 create\_from\_usb\_device() (*USBDeviceApproval* class method), 196  
 create\_group() (*AssetGroup* class method), 344  
 create\_note() (*Alert* method), 253  
 create\_note() (*CBAnalyticsAlert* method), 281  
 create\_note() (*ContainerRuntimeAlert* method), 288  
 create\_note() (*DeviceControlAlert* method), 295  
 create\_note() (*HostBasedFirewallAlert* method), 324  
 create\_note() (*IntrusionDetectionSystemAlert* method), 331  
 create\_note() (*WatchlistAlert* method), 338  
 create\_process() (*CbLRSessionBase* method), 592  
 create\_process() (*LiveResponseSession* method), 603  
 create\_profile() (*Grant* method), 392  
 create\_profile() (*Grant.GrantBuilder* method), 388  
 create\_query() (*IOC\_V2* class method), 229  
 create\_regex() (*IOC\_V2* class method), 229  
 create\_registry\_key() (*CbLRSessionBase* method), 592  
 create\_registry\_key() (*LiveResponseSession* method), 604  
 CredentialError, 584, 629  
 CredentialProvider (class in *cbc\_sdk.credentials*), 582  
 Credentials (class in *cbc\_sdk.credentials*), 582  
 CredentialValue (class in *cbc\_sdk.credentials*), 582  
 CriteriaBuilderSupportMixin (class in *cbc\_sdk.base*), 550  
 current\_policy (*DevicePolicyChangePreview* property), 457  
 current\_policy\_id (*DevicePolicyChangePreview* property), 457  
 current\_policy\_position (*DevicePolicyChangePreview* property), 457  
 custom\_severities (*CBCloudAPI* property), 125  
 custom\_severity (*Report* property), 234

## D

daemon (*JobWorker* property), 599  
 data\_collection\_rule\_configs (*Policy* property), 432  
 data\_collection\_rule\_configs\_list (*Policy* property), 432  
 DataCollectionRuleConfig (class in *cbc\_sdk.platform.policy\_ruleconfigs*), 445  
 decode\_hresult() (in module *cbc\_sdk.winerror*), 616  
 default\_action (*HostBasedFirewallRuleConfig* property), 452  
 default\_credential\_provider() (in module *cbc\_sdk.credential\_providers.default*), 173  
 DefaultProvider (class in *cbc\_sdk.credential\_providers.default*), 172  
 delete() (*Alert.Note* method), 251  
 delete() (*AssetGroup* method), 344  
 delete() (*BypassRuleConfig* method), 441  
 delete() (*CBAnalyticsAlert.Note* method), 280  
 delete() (*Connection* method), 580  
 delete() (*ContainerRuntimeAlert.Note* method), 287  
 delete() (*CorePreventionRuleConfig* method), 443  
 delete() (*DataCollectionRuleConfig* method), 446  
 delete() (*DeviceControlAlert.Note* method), 293  
 delete() (*Feed* method), 222  
 delete() (*FeedModel* method), 224  
 delete() (*Grant* method), 393  
 delete() (*Grant.Profile* method), 389  
 delete() (*HostBasedFirewallAlert.Note* method), 323  
 delete() (*HostBasedFirewallRuleConfig* method), 452  
 delete() (*HostBasedFirewallRuleConfig.FirewallRule* method), 449  
 delete() (*HostBasedFirewallRuleConfig.FirewallRuleGroup* method), 450  
 delete() (*IntrusionDetectionSystemAlert.Note* method), 330  
 delete() (*IOC* method), 227  
 delete() (*IOC\_V2* method), 229  
 delete() (*LiveResponseMemdump* method), 602  
 delete() (*MutableBaseModel* method), 559  
 delete() (*Policy* method), 432  
 delete() (*PolicyRule* method), 440  
 delete() (*PolicyRuleConfig* method), 455  
 delete() (*Report* method), 234  
 delete() (*ReportSeverity* method), 238  
 delete() (*ReputationOverride* method), 474  
 delete() (*Run* method), 150  
 delete() (*RunHistory* method), 152  
 delete() (*Template* method), 161  
 delete() (*TemplateHistory* method), 163  
 delete() (*USBDeviceApproval* method), 196  
 delete() (*USBDeviceBlock* method), 202  
 delete() (*User* method), 480  
 delete() (*Watchlist* method), 242  
 delete() (*WatchlistAlert.Note* method), 336  
 delete\_file() (*CbLRSessionBase* method), 592  
 delete\_file() (*LiveResponseSession* method), 604  
 delete\_object() (*BaseAPI* method), 574  
 delete\_object() (*CBCloudAPI* method), 125  
 delete\_registry\_key() (*CbLRSessionBase* method), 593  
 delete\_registry\_key() (*LiveResponseSession* method), 604  
 delete\_registry\_value() (*CbLRSessionBase* method), 593



[delete\\_registry\\_value\(\)](#) (*LiveResponseSession method*), 605  
[delete\\_rule\(\)](#) (*Policy method*), 432  
[delete\\_rule\\_config\(\)](#) (*Policy method*), 432  
[delete\\_sensor\(\)](#) (*Device method*), 358  
[delete\\_sensor\(\)](#) (*DeviceSearchQuery method*), 366  
[delete\\_threat\\_tag\(\)](#) (*Alert method*), 253  
[delete\\_threat\\_tag\(\)](#) (*CBAAnalyticsAlert method*), 281  
[delete\\_threat\\_tag\(\)](#) (*ContainerRuntimeAlert method*), 288  
[delete\\_threat\\_tag\(\)](#) (*DeviceControlAlert method*), 295  
[delete\\_threat\\_tag\(\)](#) (*HostBasedFirewallAlert method*), 325  
[delete\\_threat\\_tag\(\)](#) (*IntrusionDetectionSystemAlert method*), 331  
[delete\\_threat\\_tag\(\)](#) (*WatchlistAlert method*), 338  
[deobfuscate\\_cmdline\(\)](#) (*Alert method*), 253  
[deobfuscate\\_cmdline\(\)](#) (*CBAAnalyticsAlert method*), 282  
[deobfuscate\\_cmdline\(\)](#) (*ContainerRuntimeAlert method*), 289  
[deobfuscate\\_cmdline\(\)](#) (*DeviceControlAlert method*), 295  
[deobfuscate\\_cmdline\(\)](#) (*HostBasedFirewallAlert method*), 325  
[deobfuscate\\_cmdline\(\)](#) (*IntrusionDetectionSystemAlert method*), 331  
[deobfuscate\\_cmdline\(\)](#) (*Observation method*), 414  
[deobfuscate\\_cmdline\(\)](#) (*Process method*), 466  
[deobfuscate\\_cmdline\(\)](#) (*WatchlistAlert method*), 338  
[Device](#) (*class in cbc\_sdk.platform.devices*), 355  
[device\\_](#) (*Result property*), 140  
[device\\_background\\_scan\(\)](#) (*CBCloudAPI method*), 125  
[device\\_bypass\(\)](#) (*CBCloudAPI method*), 125  
[device\\_delete\\_sensor\(\)](#) (*CBCloudAPI method*), 125  
[device\\_ids\(\)](#) (*RunQuery method*), 157  
[device\\_quarantine\(\)](#) (*CBCloudAPI method*), 126  
[device\\_types\(\)](#) (*RunQuery method*), 157  
[device\\_uninstall\\_sensor\(\)](#) (*CBCloudAPI method*), 126  
[device\\_update\\_policy\(\)](#) (*CBCloudAPI method*), 126  
[device\\_update\\_sensor\\_version\(\)](#) (*CBCloudAPI method*), 126  
[DeviceControlAlert](#) (*class in cbc\_sdk.platform.alerts*), 293  
[DeviceControlAlert.Note](#) (*class in cbc\_sdk.platform.alerts*), 293  
[DeviceFacet](#) (*class in cbc\_sdk.platform.devices*), 363  
[DeviceFacet.DeviceFacetValue](#) (*class in cbc\_sdk.platform.devices*), 363  
[deviceId](#) (*Device property*), 358  
[DevicePolicyChangePreview](#) (*class in cbc\_sdk.platform.preview*), 457  
[DeviceSearchQuery](#) (*class in cbc\_sdk.platform.devices*), 365  
[DeviceSummary](#) (*class in cbc\_sdk.audit\_remediation.base*), 131  
[DeviceSummary.Metrics](#) (*class in cbc\_sdk.audit\_remediation.base*), 131  
[DeviceSummaryFacet](#) (*class in cbc\_sdk.audit\_remediation.base*), 132  
[DeviceSummaryFacet.Values](#) (*class in cbc\_sdk.audit\_remediation.base*), 133  
[Differential](#) (*class in cbc\_sdk.audit\_remediation.differential*), 167  
[DifferentialQuery](#) (*class in cbc\_sdk.audit\_remediation.differential*), 168  
[DirectoryStorageError](#) (*class in cbc\_sdk.winerror*), 614  
[disable\\_alerts\(\)](#) (*Watchlist method*), 242  
[disable\\_all\\_access\(\)](#) (*User method*), 480  
[disable\\_insecure\\_warnings\(\)](#) (*in module cbc\_sdk.helpers*), 589  
[disable\\_profiles\(\)](#) (*User method*), 480  
[disable\\_tags\(\)](#) (*Watchlist method*), 242  
[dismiss\\_threat\(\)](#) (*Alert method*), 253  
[dismiss\\_threat\(\)](#) (*CBAAnalyticsAlert method*), 282  
[dismiss\\_threat\(\)](#) (*ContainerRuntimeAlert method*), 289  
[dismiss\\_threat\(\)](#) (*DeviceControlAlert method*), 296  
[dismiss\\_threat\(\)](#) (*HostBasedFirewallAlert method*), 325  
[dismiss\\_threat\(\)](#) (*IntrusionDetectionSystemAlert method*), 332  
[dismiss\\_threat\(\)](#) (*WatchlistAlert method*), 339  
[download\(\)](#) (*AWSComputeResourceQuery method*), 515  
[download\(\)](#) (*BaseComputeResourceQuery method*), 527  
[download\(\)](#) (*DeviceSearchQuery method*), 366  
[download\(\)](#) (*VCenterComputeResourceQuery method*), 534  
[download\\_url\(\)](#) (*Binary method*), 247  
[Downloads](#) (*class in cbc\_sdk.enterprise\_edr.ubs*), 248  
[Downloads.FoundItem](#) (*class in cbc\_sdk.enterprise\_edr.ubs*), 248

## E

[email\\_addresses\(\)](#) (*UserQuery method*), 483  
[enable\\_alerts\(\)](#) (*Watchlist method*), 242  
[enable\\_tags\(\)](#) (*Watchlist method*), 242  
[enabled](#) (*HostBasedFirewallRuleConfig property*), 452  
[EnrichedEvent](#) (*class in cbc\_sdk.endpoint\_standard.base*), 176  
[EnrichedEventFacet](#) (*class in cbc\_sdk.endpoint\_standard.base*), 177

EnrichedEventFacet.Ranges (class in [cbc\\_sdk.endpoint\\_standard.base](#)), 177  
 EnrichedEventFacet.Terms (class in [cbc\\_sdk.endpoint\\_standard.base](#)), 178  
 EnrichedEventQuery (class in [cbc\\_sdk.endpoint\\_standard.base](#)), 179  
 EnvironCredentialProvider (class in [cbc\\_sdk.credential\\_providers.environ\\_credential\\_provider](#)), 173  
 EpochDateTimeFieldDescriptor (class in [cbc\\_sdk.base](#)), 551  
 eprint() (in module [cbc\\_sdk.helpers](#)), 589  
 ErrorBaseClass (class in [cbc\\_sdk.winerror](#)), 614  
 ErrorMetaClass (class in [cbc\\_sdk.winerror](#)), 615  
 Event (class in [cbc\\_sdk.endpoint\\_standard.base](#)), 184  
 Event (class in [cbc\\_sdk.platform.events](#)), 374  
 EventFacet (class in [cbc\\_sdk.platform.events](#)), 375  
 EventFacet.Ranges (class in [cbc\\_sdk.platform.events](#)), 375  
 EventFacet.Terms (class in [cbc\\_sdk.platform.events](#)), 376  
 EventFacetQuery (class in [cbc\\_sdk.platform.events](#)), 377  
 EventQuery (class in [cbc\\_sdk.platform.events](#)), 382  
 events() (Process method), 466  
 exclude\_appliance\_uuid() (VCenterComputeResourceQuery method), 534  
 exclude\_auto\_scaling\_group\_name() (AWSComputeResourceQuery method), 516  
 exclude\_availability\_zone() (AWSComputeResourceQuery method), 516  
 exclude\_cloud\_provider\_account\_id() (AWSComputeResourceQuery method), 516  
 exclude\_cloud\_provider\_resource\_id() (AWSComputeResourceQuery method), 516  
 exclude\_cloud\_provider\_tags() (AWSComputeResourceQuery method), 517  
 exclude\_cluster\_name() (VCenterComputeResourceQuery method), 534  
 exclude\_datacenter\_name() (VCenterComputeResourceQuery method), 535  
 exclude\_device\_guid() (VCenterComputeResourceQuery method), 535  
 exclude\_eligibility() (VCenterComputeResourceQuery method), 535  
 exclude\_eligibility\_code() (VCenterComputeResourceQuery method), 535  
 exclude\_esx\_host\_name() (VCenterComputeResourceQuery method), 535  
 exclude\_esx\_host\_uuid() (VCenterComputeResourceQuery method), 536  
 exclude\_host\_name() (VCenterComputeResourceQuery method), 536  
 exclude\_id() (AWSComputeResourceQuery method), 517  
 exclude\_installation\_status() (AWSComputeResourceQuery method), 517  
 exclude\_installation\_status() (VCenterComputeResourceQuery method), 536  
 exclude\_installation\_type() (VCenterComputeResourceQuery method), 536  
 exclude\_ip\_address() (VCenterComputeResourceQuery method), 536  
 exclude\_name() (AWSComputeResourceQuery method), 517  
 exclude\_name() (VCenterComputeResourceQuery method), 536  
 exclude\_os\_architecture() (VCenterComputeResourceQuery method), 537  
 exclude\_os\_description() (VCenterComputeResourceQuery method), 537  
 exclude\_os\_type() (VCenterComputeResourceQuery method), 537  
 exclude\_platform() (AWSComputeResourceQuery method), 517  
 exclude\_platform\_details() (AWSComputeResourceQuery method), 517  
 exclude\_region() (AWSComputeResourceQuery method), 518  
 exclude\_registration\_id() (VCenterComputeResourceQuery method), 537  
 exclude\_subnet\_id() (AWSComputeResourceQuery method), 518  
 exclude\_uuid() (VCenterComputeResourceQuery method), 537  
 exclude\_vcenter\_host\_url() (VCenterComputeResourceQuery method), 538  
 exclude\_vcenter\_name() (VCenterComputeResourceQuery method), 538  
 exclude\_vcenter\_uuid() (VCenterComputeResourceQuery method), 538  
 exclude\_virtual\_private\_cloud\_id() (AWSComputeResourceQuery method), 518  
 exclude\_vmwaretools\_version() (VCenterComputeResourceQuery method), 538  
 ExclusionBuilderSupportMixin (class in [cbc\\_sdk.base](#)), 551  
 execute\_async() (AffectedAssetQuery method), 484  
 execute\_async() (AssetGroupQuery method), 351  
 execute\_async() (AsyncProcessQuery method), 460  
 execute\_async() (AsyncQueryMixin method), 549  
 execute\_async() (AuthEventQuery method), 215  
 execute\_async() (AWSComputeResourceQuery method), 518  
 execute\_async() (BaseComputeResourceQuery method), 527  
 execute\_async() (DeviceSearchQuery method), 367  
 execute\_async() (EnrichedEventQuery method), 181

- execute\_async() (*EventFacetQuery* method), 379  
 execute\_async() (*EventQuery* method), 384  
 execute\_async() (*FacetQuery* method), 134, 554  
 execute\_async() (*GrantQuery* method), 395  
 execute\_async() (*JobQuery* method), 398  
 execute\_async() (*ObservationQuery* method), 420  
 execute\_async() (*PolicyQuery* method), 438  
 execute\_async() (*Query* method), 564  
 execute\_async() (*RecommendationQuery* method), 190  
 execute\_async() (*ReputationOverrideQuery* method), 475  
 execute\_async() (*ResultQuery* method), 144  
 execute\_async() (*RunHistoryQuery* method), 154  
 execute\_async() (*RunQuery* method), 157  
 execute\_async() (*SensorKitQuery* method), 511  
 execute\_async() (*SummaryQuery* method), 470  
 execute\_async() (*TemplateHistoryQuery* method), 165  
 execute\_async() (*USBDeviceApprovalQuery* method), 198  
 execute\_async() (*USBDeviceBlockQuery* method), 203  
 execute\_async() (*USBDeviceQuery* method), 204  
 execute\_async() (*UserQuery* method), 483  
 execute\_async() (*VCenterComputeResourceQuery* method), 538  
 execute\_async() (*VulnerabilityAssetViewQuery* method), 494  
 execute\_async() (*VulnerabilityQuery* method), 502  
 expires() (*SensorKitQuery* method), 511  
 export() (*AffectedAssetQuery* method), 485  
 export() (*DeviceSearchQuery* method), 367  
 export() (*USBDeviceApprovalQuery* method), 199  
 export() (*USBDeviceQuery* method), 204  
 export() (*VulnerabilityAssetViewQuery* method), 494  
 export() (*VulnerabilityQuery* method), 502  
 export\_csv\_as\_file() (*ResultQuery* method), 144  
 export\_csv\_as\_lines() (*ResultQuery* method), 144  
 export\_csv\_as\_stream() (*ResultQuery* method), 144  
 export\_csv\_as\_string() (*ResultQuery* method), 144  
 export\_rules() (*HostBasedFirewallRuleConfig* method), 452  
 export\_zipped\_csv() (*ResultQuery* method), 145  
 extend() (*Vulnerability.AssetView* method), 491
- ## F
- facet() (*AWSComputeResourceQuery* method), 518  
 facet() (*BaseComputeResourceQuery* method), 527  
 facet() (*VCenterComputeResourceQuery* method), 539  
 facet\_field() (*FacetQuery* method), 135  
 FacetQuery (class in *cbc\_sdk.audit\_remediation.base*), 134  
 FacetQuery (class in *cbc\_sdk.base*), 552  
 facets (*AuthEventFacet.Ranges* property), 211  
 facets (*AuthEventFacet.Terms* property), 212  
 facets (*EnrichedEventFacet.Ranges* property), 178  
 facets (*EnrichedEventFacet.Terms* property), 178  
 facets (*EventFacet.Ranges* property), 376  
 facets (*EventFacet.Terms* property), 376  
 facets (*ObservationFacet.Ranges* property), 417  
 facets (*ObservationFacet.Terms* property), 417  
 facets (*ProcessFacet.Ranges* property), 468  
 facets (*ProcessFacet.Terms* property), 469  
 facets() (*AlertSearchQuery* method), 260  
 facets() (*DeviceSearchQuery* method), 367  
 facets() (*GroupedAlertSearchQuery* method), 303  
 facets() (*Process* method), 466  
 facets() (*USBDeviceQuery* method), 205  
 Facility (class in *cbc\_sdk.winerror*), 615  
 FAILED() (in module *cbc\_sdk.winerror*), 615  
 Feed (class in *cbc\_sdk.enterprise\_edr.threat\_intelligence*), 220  
 feed (Watchlist property), 243  
 Feed.FeedBuilder (class in *cbc\_sdk.enterprise\_edr.threat\_intelligence*), 220  
 FeedModel (class in *cbc\_sdk.enterprise\_edr.threat\_intelligence*), 224  
 FeedQuery (class in *cbc\_sdk.enterprise\_edr.threat\_intelligence*), 225  
 fetch\_process\_queries() (*CBCloudAPI* method), 127  
 FieldDescriptor (class in *cbc\_sdk.base*), 557  
 fields (*AuthEventFacet.Ranges* property), 211  
 fields (*AuthEventFacet.Terms* property), 212  
 fields (*EnrichedEventFacet.Ranges* property), 178  
 fields (*EnrichedEventFacet.Terms* property), 178  
 fields (*EventFacet.Ranges* property), 376  
 fields (*EventFacet.Terms* property), 376  
 fields (*ObservationFacet.Ranges* property), 417  
 fields (*ObservationFacet.Terms* property), 417  
 fields (*ProcessFacet.Ranges* property), 469  
 fields (*ProcessFacet.Terms* property), 469  
 fields\_ (*Result* property), 140  
 FileCredentialProvider (class in *cbc\_sdk.credential\_providers.file\_credential\_provider*), 174  
 first() (*AffectedAssetQuery* method), 485  
 first() (*AlertSearchQuery* method), 260  
 first() (*AssetGroupQuery* method), 352  
 first() (*AsyncProcessQuery* method), 460  
 first() (*AuthEventQuery* method), 215  
 first() (*AWSComputeResourceQuery* method), 519  
 first() (*BaseComputeResourceQuery* method), 528  
 first() (*DeviceSearchQuery* method), 368  
 first() (*DifferentialQuery* method), 170  
 first() (*EnrichedEventQuery* method), 181  
 first() (*EventQuery* method), 384

`first()` (*FacetQuery* method), 135  
`first()` (*FeedQuery* method), 226  
`first()` (*GrantQuery* method), 395  
`first()` (*GroupedAlertSearchQuery* method), 304  
`first()` (*IterableQueryMixin* method), 558  
`first()` (*JobQuery* method), 398  
`first()` (*ObservationQuery* method), 420  
`first()` (*PaginatedQuery* method), 561  
`first()` (*PolicyQuery* method), 438  
`first()` (*Query* method), 564  
`first()` (*RecommendationQuery* method), 190  
`first()` (*ReportQuery* method), 237  
`first()` (*ReputationOverrideQuery* method), 475  
`first()` (*ResultQuery* method), 145  
`first()` (*RunHistoryQuery* method), 154  
`first()` (*SensorKitQuery* method), 512  
`first()` (*SimpleQuery* method), 570  
`first()` (*TemplateHistoryQuery* method), 165  
`first()` (*USBDeviceApprovalQuery* method), 199  
`first()` (*USBDeviceBlockQuery* method), 203  
`first()` (*USBDeviceQuery* method), 205  
`first()` (*UserQuery* method), 483  
`first()` (*VCenterComputeResourceQuery* method), 539  
`first()` (*VulnerabilityAssetViewQuery* method), 495  
`first()` (*VulnerabilityQuery* method), 502  
`first()` (*WatchlistQuery* method), 245  
`ForeignKeyFieldDescriptor` (class in `cbc_sdk.base`), 557  
`found` (*Downloads* property), 248  
`from_type()` (*SensorKit* class method), 509  
`FunctionalityDecommissioned`, 585

## G

`get()` (*Alert* method), 254  
`get()` (*Alert.Note* method), 251  
`get()` (*AssetGroup* method), 345  
`get()` (*AuditLog* method), 354  
`get()` (*AuthEvent* method), 209  
`get()` (*AuthEventFacet* method), 212  
`get()` (*AuthEventFacet.Ranges* method), 212  
`get()` (*AuthEventFacet.Terms* method), 212  
`get()` (*AWSComputeResource* method), 514  
`get()` (*BaseComputeResource* method), 525  
`get()` (*Binary* method), 247  
`get()` (*Binary.Summary* method), 247  
`get()` (*BypassRuleConfig* method), 442  
`get()` (*CBAnalyticsAlert* method), 282  
`get()` (*CBAnalyticsAlert.Note* method), 280  
`get()` (*ComputeResourceFacet* method), 531  
`get()` (*ComputeResourceFacet.ComputeResourceFacet.ComputeResourceFacet.Value* method), 530  
`get()` (*Connection* method), 580  
`get()` (*ContainerRuntimeAlert* method), 289

`get()` (*ContainerRuntimeAlert.Note* method), 287  
`get()` (*CorePreventionRuleConfig* method), 443  
`get()` (*DataCollectionRuleConfig* method), 446  
`get()` (*Device* method), 358  
`get()` (*DeviceControlAlert* method), 296  
`get()` (*DeviceControlAlert.Note* method), 294  
`get()` (*DeviceFacet* method), 364  
`get()` (*DeviceFacet.DeviceFacetValue* method), 363  
`get()` (*DeviceSummary* method), 132  
`get()` (*DeviceSummary.Metrics* method), 131  
`get()` (*DeviceSummaryFacet* method), 133  
`get()` (*DeviceSummaryFacet.Values* method), 133  
`get()` (*Differential* method), 168  
`get()` (*Downloads* method), 249  
`get()` (*Downloads.FoundItem* method), 248  
`get()` (*EnrichedEvent* method), 176  
`get()` (*EnrichedEventFacet* method), 179  
`get()` (*EnrichedEventFacet.Ranges* method), 178  
`get()` (*EnrichedEventFacet.Terms* method), 178  
`get()` (*Event* method), 374  
`get()` (*EventFacet* method), 377  
`get()` (*EventFacet.Ranges* method), 376  
`get()` (*EventFacet.Terms* method), 376  
`get()` (*Feed* method), 222  
`get()` (*FeedModel* method), 224  
`get()` (*Grant* method), 393  
`get()` (*Grant.Profile* method), 389  
`get()` (*GroupedAlert* method), 300  
`get()` (*HostBasedFirewallAlert* method), 325  
`get()` (*HostBasedFirewallAlert.Note* method), 323  
`get()` (*HostBasedFirewallRuleConfig* method), 453  
`get()` (*HostBasedFirewallRuleConfig.FirewallRule* method), 449  
`get()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup* method), 450  
`get()` (*IntrusionDetectionSystemAlert* method), 332  
`get()` (*IntrusionDetectionSystemAlert.Note* method), 330  
`get()` (*IOC* method), 227  
`get()` (*IOC\_V2* method), 229  
`get()` (*Job* method), 396  
`get()` (*LiveResponseMemdump* method), 602  
`get()` (*MutableBaseModel* method), 559  
`get()` (*NetworkThreatMetadata* method), 413  
`get()` (*NewBaseModel* method), 560  
`get()` (*Observation* method), 415  
`get()` (*ObservationFacet* method), 418  
`get()` (*ObservationFacet.Ranges* method), 417  
`get()` (*ObservationFacet.Terms* method), 417  
`get()` (*PlatformModel* method), 249  
`get()` (*Policy* method), 433  
`get()` (*PolicyRule* method), 440  
`get()` (*PolicyRuleConfig* method), 455  
`get()` (*Process* method), 466  
`get()` (*Process.Summary* method), 464



- `get()` (*Process.Tree* method), 465
- `get()` (*ProcessFacet* method), 470
- `get()` (*ProcessFacet.Ranges* method), 469
- `get()` (*ProcessFacet.Terms* method), 469
- `get()` (*Recommendation* method), 188
- `get()` (*Recommendation.RecommendationApplication* method), 185
- `get()` (*Recommendation.RecommendationImpact* method), 186
- `get()` (*Recommendation.RecommendationNewRule* method), 187
- `get()` (*Recommendation.RecommendationWorkflow* method), 188
- `get()` (*Report* method), 234
- `get()` (*ReportSeverity* method), 238
- `get()` (*ReputationOverride* method), 474
- `get()` (*Result* method), 140
- `get()` (*Result.Device* method), 139
- `get()` (*Result.Fields* method), 139
- `get()` (*Result.Metrics* method), 140
- `get()` (*ResultFacet* method), 142
- `get()` (*ResultFacet.Values* method), 142
- `get()` (*Run* method), 150
- `get()` (*RunHistory* method), 152
- `get()` (*SensorKit* method), 510
- `get()` (*Template* method), 161
- `get()` (*TemplateHistory* method), 163
- `get()` (*UnrefreshableModel* method), 571
- `get()` (*USBDevice* method), 194
- `get()` (*USBDeviceApproval* method), 197
- `get()` (*USBDeviceBlock* method), 202
- `get()` (*User* method), 480
- `get()` (*VCenterComputeResource* method), 532
- `get()` (*Vulnerability* method), 493
- `get()` (*Vulnerability.OrgSummary* method), 492
- `get()` (*Watchlist* method), 243
- `get()` (*WatchlistAlert* method), 339
- `get()` (*WatchlistAlert.Note* method), 337
- `get_affected_assets()` (*Vulnerability* method), 493
- `get_alert_search_query()` (*GroupedAlert* method), 300
- `get_alert_search_query()` (*GroupedAlertSearch-Query* method), 304
- `get_alerts()` (*GroupedAlert* method), 301
- `get_all_groups()` (*AssetGroup* class method), 345
- `get_asset_group_ids()` (*Device* method), 358
- `get_asset_groups()` (*Device* method), 358
- `get_asset_groups_for_devices()` (*Device* class method), 359
- `get_assignment_mode()` (*CorePreventionRuleConfig* method), 444
- `get_auditlogs()` (*AuditLog* static method), 354
- `get_auditlogs()` (*CBCloudAPI* method), 127
- `get_auth_events_descriptions()` (*AuthEvent* static method), 209
- `get_cb_cloud_object()` (in module *cbc\_sdk.helpers*), 589
- `get_config_template()` (*SensorKit* class method), 510
- `get_connection()` (*CBCSDKSessionAdapter* method), 578
- `get_credentials()` (*AWSCredentialProvider* method), 173
- `get_credentials()` (*CredentialProvider* method), 582
- `get_credentials()` (*EnvironCredentialProvider* method), 173
- `get_credentials()` (*FileCredentialProvider* method), 174
- `get_credentials()` (*KeychainCredentialProvider* method), 175
- `get_credentials()` (*RegistryCredentialProvider* method), 175
- `get_default_provider()` (*DefaultProvider* method), 172
- `get_details()` (*AuthEvent* method), 210
- `get_details()` (*EnrichedEvent* method), 177
- `get_details()` (*Observation* method), 415
- `get_details()` (*Process* method), 466
- `get_endpoints()` (*USBDevice* method), 194
- `get_events()` (*CBAnalyticsAlert* method), 282
- `get_file()` (*CbLRSessionBase* method), 593
- `get_file()` (*LiveResponseSession* method), 605
- `get_group_results()` (*ObservationQuery* method), 420
- `get_history()` (*Alert* method), 254
- `get_history()` (*CBAnalyticsAlert* method), 283
- `get_history()` (*ContainerRuntimeAlert* method), 290
- `get_history()` (*DeviceControlAlert* method), 296
- `get_history()` (*HostBasedFirewallAlert* method), 326
- `get_history()` (*IntrusionDetectionSystemAlert* method), 332
- `get_history()` (*WatchlistAlert* method), 339
- `get_network_threat_metadata()` (*IntrusionDetectionSystemAlert* method), 333
- `get_network_threat_metadata()` (*Observation* method), 415
- `get_notifications()` (*CBCloudAPI* method), 127
- `get_object()` (*BaseAPI* method), 574
- `get_object()` (*CBCloudAPI* method), 127
- `get_object_by_name_or_id()` (in module *cbc\_sdk.helpers*), 589
- `get_observations()` (*Alert* method), 254
- `get_observations()` (*CBAnalyticsAlert* method), 283
- `get_observations()` (*ContainerRuntimeAlert* method), 290
- `get_observations()` (*DeviceControlAlert* method), 297

[get\\_observations\(\)](#) (*HostBasedFirewallAlert method*), 326  
[get\\_observations\(\)](#) (*IntrusionDetectionSystemAlert method*), 333  
[get\\_observations\(\)](#) (*WatchlistAlert method*), 340  
[get\\_output\\_as\\_file\(\)](#) (*Job method*), 396  
[get\\_output\\_as\\_lines\(\)](#) (*Job method*), 397  
[get\\_output\\_as\\_stream\(\)](#) (*Job method*), 397  
[get\\_output\\_as\\_string\(\)](#) (*Job method*), 397  
[get\\_parameter\(\)](#) (*BypassRuleConfig method*), 442  
[get\\_parameter\(\)](#) (*CorePreventionRuleConfig method*), 444  
[get\\_parameter\(\)](#) (*DataCollectionRuleConfig method*), 446  
[get\\_parameter\(\)](#) (*HostBasedFirewallRuleConfig method*), 453  
[get\\_parameter\(\)](#) (*PolicyRuleConfig method*), 456  
[get\\_permitted\\_role\\_urns\(\)](#) (*Grant class method*), 393  
[get\\_policy\\_ruleconfig\\_parameter\\_schema\(\)](#) (*CBCloudAPI method*), 128  
[get\\_process\(\)](#) (*Alert method*), 255  
[get\\_process\(\)](#) (*CBAntalyticsAlert method*), 283  
[get\\_process\(\)](#) (*ContainerRuntimeAlert method*), 290  
[get\\_process\(\)](#) (*DeviceControlAlert method*), 297  
[get\\_process\(\)](#) (*HostBasedFirewallAlert method*), 326  
[get\\_process\(\)](#) (*IntrusionDetectionSystemAlert method*), 333  
[get\\_process\(\)](#) (*WatchlistAlert method*), 340  
[get\\_progress\(\)](#) (*Job method*), 397  
[get\\_raw\\_data\(\)](#) (*BaseAPI method*), 575  
[get\\_raw\\_data\(\)](#) (*CBCloudAPI method*), 128  
[get\\_raw\\_file\(\)](#) (*CbLRSessionBase method*), 593  
[get\\_raw\\_file\(\)](#) (*LiveResponseSession method*), 605  
[get\\_registry\\_value\(\)](#) (*CbLRSessionBase method*), 593  
[get\\_registry\\_value\(\)](#) (*LiveResponseSession method*), 605  
[get\\_ruleconfig\\_parameter\\_schema\(\)](#) (*Policy method*), 433  
[get\\_statistics\(\)](#) (*AssetGroup method*), 345  
[get\\_threat\\_tags\(\)](#) (*Alert method*), 255  
[get\\_threat\\_tags\(\)](#) (*CBAntalyticsAlert method*), 284  
[get\\_threat\\_tags\(\)](#) (*ContainerRuntimeAlert method*), 290  
[get\\_threat\\_tags\(\)](#) (*DeviceControlAlert method*), 297  
[get\\_threat\\_tags\(\)](#) (*HostBasedFirewallAlert method*), 327  
[get\\_threat\\_tags\(\)](#) (*IntrusionDetectionSystemAlert method*), 333  
[get\\_threat\\_tags\(\)](#) (*WatchlistAlert method*), 340  
[get\\_token\(\)](#) (*Credentials method*), 583  
[get\\_token\\_type\(\)](#) (*Credentials method*), 583  
[get\\_value\(\)](#) (*Credentials method*), 583  
[get\\_vendors\\_and\\_products\\_seen\(\)](#) (*USBDevice class method*), 194  
[get\\_vulnerability\\_summary\(\)](#) (*Device method*), 359  
[get\\_vulnerabilities\(\)](#) (*Device method*), 359  
[get\\_watchlist\\_objects\(\)](#) (*WatchlistAlert method*), 340  
[GetFileJob](#) (*class in cbc\_sdk.live\_response\_api*), 598  
[getName\(\)](#) (*JobWorker method*), 599  
[getName\(\)](#) (*LiveResponseJobScheduler method*), 601  
[getName\(\)](#) (*LRUCacheDict.EmptyCacheThread method*), 547  
[GetScore\(\)](#) (*in module cbc\_sdk.winerror*), 615  
[Grant](#) (*class in cbc\_sdk.platform.grants*), 387  
[grant\(\)](#) (*User method*), 481  
[Grant.GrantBuilder](#) (*class in cbc\_sdk.platform.grants*), 388  
[Grant.Profile](#) (*class in cbc\_sdk.platform.grants*), 389  
[Grant.ProfileBuilder](#) (*class in cbc\_sdk.platform.grants*), 390  
[GrantQuery](#) (*class in cbc\_sdk.platform.grants*), 394  
[group\\_results\(\)](#) (*AuthEventQuery method*), 215  
[GroupedAlert](#) (*class in cbc\_sdk.platform.alerts*), 299  
[GroupedAlertSearchQuery](#) (*class in cbc\_sdk.platform.alerts*), 301

## H

[host\\_based\\_firewall\\_rule\\_config](#) (*Policy property*), 433  
[HostBasedFirewallAlert](#) (*class in cbc\_sdk.platform.alerts*), 322  
[HostBasedFirewallAlert.Note](#) (*class in cbc\_sdk.platform.alerts*), 322  
[HostBasedFirewallRuleConfig](#) (*class in cbc\_sdk.platform.policy\_ruleconfigs*), 448  
[HostBasedFirewallRuleConfig.FirewallRule](#) (*class in cbc\_sdk.platform.policy\_ruleconfigs*), 448  
[HostBasedFirewallRuleConfig.FirewallRuleGroup](#) (*class in cbc\_sdk.platform.policy\_ruleconfigs*), 450  
[HRESULT\\_CODE\(\)](#) (*in module cbc\_sdk.winerror*), 615  
[HRESULT\\_FACILITY\(\)](#) (*in module cbc\_sdk.winerror*), 615  
[HRESULT\\_FROM\\_NT\(\)](#) (*in module cbc\_sdk.winerror*), 615  
[HRESULT\\_FROM\\_WIN32\(\)](#) (*in module cbc\_sdk.winerror*), 615  
[HRESULT\\_SEVERITY\(\)](#) (*in module cbc\_sdk.winerror*), 616  
[http\\_request\(\)](#) (*Connection method*), 580

## I

[ident](#) (*JobWorker property*), 599  
[ident](#) (*LiveResponseJobScheduler property*), 601

- `ident` (*LRUCacheDict.EmptyCacheThread* property), 547
  - `ignore()` (*IOC\_V2* method), 230
  - `ignore()` (*Report* method), 234
  - `ignored` (*IOC\_V2* property), 230
  - `ignored` (*Report* property), 234
  - `impact_` (*Recommendation* property), 188
  - `index()` (*Vulnerability.AssetView* method), 491
  - `init_poolmanager()` (*CBCSDKSessionAdapter* method), 578
  - `insert()` (*Vulnerability.AssetView* method), 491
  - `install_sensor()` (*AWSComputeResource* method), 514
  - `install_sensor()` (*BaseComputeResource* method), 525
  - `install_sensor()` (*VCenterComputeResource* method), 532
  - `IntrusionDetectionSystemAlert` (class in *cbc\_sdk.platform.alerts*), 329
  - `IntrusionDetectionSystemAlert.Note` (class in *cbc\_sdk.platform.alerts*), 329
  - `InvalidHashError`, 585
  - `InvalidObjectError`, 585, 630
  - `IOC` (class in *cbc\_sdk.enterprise\_edr.threat\_intelligence*), 226
  - `IOC_V2` (class in *cbc\_sdk.enterprise\_edr.threat\_intelligence*), 228
  - `iocs_` (*Report* property), 235
  - `ipv6_equality_format()` (*IOC\_V2* class method), 230
  - `is_alive()` (*JobWorker* method), 599
  - `is_alive()` (*LiveResponseJobScheduler* method), 601
  - `is_alive()` (*LRUCacheDict.EmptyCacheThread* method), 547
  - `is_deleted` (*PolicyRule* property), 440
  - `is_dirty()` (*AssetGroup* method), 345
  - `is_dirty()` (*BypassRuleConfig* method), 442
  - `is_dirty()` (*CorePreventionRuleConfig* method), 444
  - `is_dirty()` (*DataCollectionRuleConfig* method), 447
  - `is_dirty()` (*Feed* method), 222
  - `is_dirty()` (*FeedModel* method), 224
  - `is_dirty()` (*Grant* method), 393
  - `is_dirty()` (*Grant.Profile* method), 389
  - `is_dirty()` (*HostBasedFirewallRuleConfig* method), 453
  - `is_dirty()` (*HostBasedFirewallRuleConfig.FirewallRule* method), 449
  - `is_dirty()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup* method), 451
  - `is_dirty()` (*IOC* method), 227
  - `is_dirty()` (*IOC\_V2* method), 230
  - `is_dirty()` (*MutableBaseModel* method), 559
  - `is_dirty()` (*Policy* method), 433
  - `is_dirty()` (*PolicyRule* method), 440
  - `is_dirty()` (*PolicyRuleConfig* method), 456
  - `is_dirty()` (*Report* method), 235
  - `is_dirty()` (*ReportSeverity* method), 239
  - `is_dirty()` (*USBDeviceApproval* method), 197
  - `is_dirty()` (*User* method), 481
  - `is_dirty()` (*Watchlist* method), 243
  - `isDaemon()` (*JobWorker* method), 599
  - `isDaemon()` (*LiveResponseJobScheduler* method), 601
  - `isDaemon()` (*LRUCacheDict.EmptyCacheThread* method), 547
  - `IsoDateTimeFieldDescriptor` (class in *cbc\_sdk.base*), 558
  - `IterableQueryMixin` (class in *cbc\_sdk.base*), 558
- ## J
- `Job` (class in *cbc\_sdk.platform.jobs*), 395
  - `JobQuery` (class in *cbc\_sdk.platform.jobs*), 398
  - `jobrunner()` (in module *cbc\_sdk.live\_response\_api*), 611
  - `JobWorker` (class in *cbc\_sdk.live\_response\_api*), 599
  - `join()` (*JobWorker* method), 599
  - `join()` (*LiveResponseJobScheduler* method), 601
  - `join()` (*LRUCacheDict.EmptyCacheThread* method), 547
- ## K
- `KeychainCredentialProvider` (class in *cbc\_sdk.credential\_providers.keychain\_credential\_provider*), 174
  - `kill_process()` (*CbLRSessionBase* method), 594
  - `kill_process()` (*LiveResponseSession* method), 606
- ## L
- `latestRevision` (*Policy* property), 433
  - `LegacyAlertSearchQueryCriterionMixin` (class in *cbc\_sdk.platform.legacy\_alerts*), 399
  - `limit()` (*EventFacetQuery* method), 379
  - `limit()` (*FacetQuery* method), 554
  - `list_directory()` (*CbLRSessionBase* method), 594
  - `list_directory()` (*LiveResponseSession* method), 606
  - `list_member_ids()` (*AssetGroup* method), 346
  - `list_members()` (*AssetGroup* method), 346
  - `list_processes()` (*CbLRSessionBase* method), 595
  - `list_processes()` (*LiveResponseSession* method), 607
  - `list_registry_keys_and_values()` (*CbLRSessionBase* method), 595
  - `list_registry_keys_and_values()` (*LiveResponseSession* method), 607
  - `list_registry_values()` (*CbLRSessionBase* method), 596
  - `list_registry_values()` (*LiveResponseSession* method), 608
  - `live_response` (*CBCloudAPI* property), 128
  - `LiveResponseError`, 600

[LiveResponseJobScheduler](#) (class in [cbc\\_sdk.live\\_response\\_api](#)), 601  
[LiveResponseMemdump](#) (class in [cbc\\_sdk.live\\_response\\_api](#)), 602  
[LiveResponseSession](#) (class in [cbc\\_sdk.live\\_response\\_api](#)), 603  
[LiveResponseSessionManager](#) (class in [cbc\\_sdk.live\\_response\\_api](#)), 610  
[log](#) (in module [cbc\\_sdk.base](#)), 572  
[log](#) (in module [cbc\\_sdk.endpoint\\_standard.base](#)), 184  
[log](#) (in module [cbc\\_sdk.endpoint\\_standard.recommendation](#)), 192  
[log](#) (in module [cbc\\_sdk.endpoint\\_standard.usb\\_device\\_control](#)), 208  
[log](#) (in module [cbc\\_sdk.enterprise\\_edr.threat\\_intelligence](#)), 246  
[log](#) (in module [cbc\\_sdk.platform.base](#)), 250  
[log](#) (in module [cbc\\_sdk.platform.devices](#)), 374  
[log](#) (in module [cbc\\_sdk.platform.grants](#)), 395  
[log](#) (in module [cbc\\_sdk.platform.users](#)), 483  
[log](#) (in module [cbc\\_sdk.platform.vulnerability\\_assessment](#)), 507  
[log](#) (in module [cbc\\_sdk.workload.vm\\_workloads\\_search](#)), 545  
[lookup\\_error\(\)](#) ([CommDlgError](#) class method), 614  
[lookup\\_error\(\)](#) ([DirectoryStorageError](#) class method), 614  
[lookup\\_error\(\)](#) ([ErrorBaseClass](#) class method), 614  
[lookup\\_error\(\)](#) ([Facility](#) class method), 615  
[lookup\\_error\(\)](#) ([RawErrorCode](#) class method), 616  
[lookup\\_error\(\)](#) ([Win32Error](#) class method), 616  
[lr\\_session\(\)](#) ([Device](#) method), 360  
[lru\\_cache\\_function\(\)](#) (in module [cbc\\_sdk.cache.lru](#)), 549  
[LRUCachedFunction](#) (class in [cbc\\_sdk.cache.lru](#)), 548  
[LRUCacheDict](#) (class in [cbc\\_sdk.cache.lru](#)), 546  
[LRUCacheDict.EmptyCacheThread](#) (class in [cbc\\_sdk.cache.lru](#)), 546

## M

[matches\\_template\(\)](#) ([Grant.Profile](#) method), 390  
[MAX\\_RESULTS\\_LIMIT](#) (in module [cbc\\_sdk.audit\\_remediation.base](#)), 138  
[memdump\(\)](#) ([CbLRSessionBase](#) method), 596  
[memdump\(\)](#) ([LiveResponseSession](#) method), 608  
[metrics\\_](#) ([DeviceSummary](#) property), 132  
[metrics\\_](#) ([Result](#) property), 140  
[ModelNotFound](#), 586  
[module](#)  
     [cbc\\_sdk.audit\\_remediation.base](#), 131  
     [cbc\\_sdk.audit\\_remediation.differential](#), 167  
     [cbc\\_sdk.base](#), 549  
     [cbc\\_sdk.cache.lru](#), 546  
     [cbc\\_sdk.connection](#), 572  
     [cbc\\_sdk.credential\\_providers.aws\\_sm\\_credential\\_provider](#), 173  
     [cbc\\_sdk.credential\\_providers.default](#), 172  
     [cbc\\_sdk.credential\\_providers.envron\\_credential\\_provider](#), 173  
     [cbc\\_sdk.credential\\_providers.file\\_credential\\_provider](#), 174  
     [cbc\\_sdk.credential\\_providers.keychain\\_credential\\_provider](#), 174  
     [cbc\\_sdk.credential\\_providers.registry\\_credential\\_provider](#), 175  
     [cbc\\_sdk.credentials](#), 582  
     [cbc\\_sdk.endpoint\\_standard.base](#), 176  
     [cbc\\_sdk.endpoint\\_standard.recommendation](#), 184  
     [cbc\\_sdk.endpoint\\_standard.usb\\_device\\_control](#), 193  
     [cbc\\_sdk.enterprise\\_edr.auth\\_events](#), 208  
     [cbc\\_sdk.enterprise\\_edr.threat\\_intelligence](#), 220  
     [cbc\\_sdk.enterprise\\_edr.ubs](#), 246  
     [cbc\\_sdk.errors](#), 584  
     [cbc\\_sdk.helpers](#), 589  
     [cbc\\_sdk.live\\_response\\_api](#), 590  
     [cbc\\_sdk.platform.alerts](#), 250  
     [cbc\\_sdk.platform.asset\\_groups](#), 343  
     [cbc\\_sdk.platform.audit](#), 354  
     [cbc\\_sdk.platform.base](#), 249  
     [cbc\\_sdk.platform.devices](#), 355  
     [cbc\\_sdk.platform.events](#), 374  
     [cbc\\_sdk.platform.grants](#), 387  
     [cbc\\_sdk.platform.jobs](#), 395  
     [cbc\\_sdk.platform.legacy\\_alerts](#), 399  
     [cbc\\_sdk.platform.network\\_threat\\_metadata](#), 412  
     [cbc\\_sdk.platform.observations](#), 414  
     [cbc\\_sdk.platform.policies](#), 424  
     [cbc\\_sdk.platform.policy\\_ruleconfigs](#), 441  
     [cbc\\_sdk.platform.previewer](#), 457  
     [cbc\\_sdk.platform.processes](#), 458  
     [cbc\\_sdk.platform.reputation](#), 472  
     [cbc\\_sdk.platform.users](#), 477  
     [cbc\\_sdk.platform.vulnerability\\_assessment](#), 484  
     [cbc\\_sdk.utils](#), 612  
     [cbc\\_sdk.winerror](#), 614  
     [cbc\\_sdk.workload.nsx\\_remediation](#), 507  
     [cbc\\_sdk.workload.sensor\\_lifecycle](#), 509  
     [cbc\\_sdk.workload.vm\\_workloads\\_search](#), 513  
[MoreThanOneResultError](#), 586, 630  
[most\\_recent\\_alert\\_](#) ([GroupedAlert](#) property), 301  
[mro\(\)](#) ([CbMetaModel](#) method), 550  
[mro\(\)](#) ([ErrorMetaClass](#) method), 615



MutableBaseModel (class in *cbc\_sdk.base*), 558

## N

name (JobWorker property), 600

name (LiveResponseJobScheduler property), 601

name (LRUCacheDict.EmptyCacheThread property), 547

name() (RunQuery method), 157

native\_id (JobWorker property), 600

native\_id (LiveResponseJobScheduler property), 602

native\_id (LRUCacheDict.EmptyCacheThread property), 547

NetworkThreatMetadata (class in *cbc\_sdk.platform.network\_threat\_metadata*), 412

new\_policy (DevicePolicyChangePreview property), 458

new\_policy\_id (DevicePolicyChangePreview property), 458

new\_policy\_position (DevicePolicyChangePreview property), 458

new\_rule\_ (Recommendation property), 189

NewBaseModel (class in *cbc\_sdk.base*), 560

newer\_run\_id() (DifferentialQuery method), 170

NonQueryableModel, 586

normalize\_org() (in module *cbc\_sdk.platform.grants*), 395

normalize\_profile\_list() (in module *cbc\_sdk.platform.users*), 483

not\_() (AffectedAssetQuery method), 485

not\_() (AlertSearchQuery method), 260

not\_() (AssetGroupQuery method), 352

not\_() (AsyncProcessQuery method), 460

not\_() (AuthEventQuery method), 216

not\_() (AWSComputeResourceQuery method), 519

not\_() (BaseComputeResourceQuery method), 528

not\_() (DeviceSearchQuery method), 368

not\_() (EnrichedEventQuery method), 181

not\_() (EventFacetQuery method), 380

not\_() (EventQuery method), 384

not\_() (FacetQuery method), 135, 555

not\_() (GroupedAlertSearchQuery method), 304

not\_() (ObservationQuery method), 421

not\_() (Query method), 564

not\_() (QueryBuilder method), 567

not\_() (QueryBuilderSupportMixin method), 568

not\_() (ReputationOverrideQuery method), 475

not\_() (ResultQuery method), 145

not\_() (RunHistoryQuery method), 155

not\_() (SummaryQuery method), 471

not\_() (TemplateHistoryQuery method), 165

not\_() (USBDeviceApprovalQuery method), 199

not\_() (USBDeviceQuery method), 205

not\_() (VCenterComputeResourceQuery method), 539

not\_() (VulnerabilityAssetViewQuery method), 495

not\_() (VulnerabilityQuery method), 502

notes\_() (Alert method), 255

notes\_() (CBAnalyticsAlert method), 284

notes\_() (ContainerRuntimeAlert method), 291

notes\_() (DeviceControlAlert method), 297

notes\_() (HostBasedFirewallAlert method), 327

notes\_() (IntrusionDetectionSystemAlert method), 334

notes\_() (WatchlistAlert method), 341

notification\_listener() (CBCloudAPI method), 128

notify\_on\_finish() (RunQuery method), 157

nsx\_available (Device property), 360

nsx\_remediation() (Device method), 360

NSXJobError, 586

NSXRemediationJob (class in *cbc\_sdk.workload.nsx\_remediation*), 507

## O

object\_rule\_configs (Policy property), 433

object\_rule\_configs\_list (Policy property), 434

object\_rules (Policy property), 434

ObjectFieldDescriptor (class in *cbc\_sdk.base*), 561

ObjectNotFoundError, 587, 630

Observation (class in *cbc\_sdk.platform.observations*), 414

ObservationFacet (class in *cbc\_sdk.platform.observations*), 416

ObservationFacet.Ranges (class in *cbc\_sdk.platform.observations*), 416

ObservationFacet.Terms (class in *cbc\_sdk.platform.observations*), 417

ObservationGroup (class in *cbc\_sdk.platform.observations*), 418

ObservationQuery (class in *cbc\_sdk.platform.observations*), 419

older\_run\_id() (DifferentialQuery method), 171

one() (AffectedAssetQuery method), 485

one() (AlertSearchQuery method), 261

one() (AssetGroupQuery method), 352

one() (AsyncProcessQuery method), 460

one() (AuthEventQuery method), 216

one() (AWSComputeResourceQuery method), 519

one() (BaseComputeResourceQuery method), 528

one() (DeviceSearchQuery method), 368

one() (DifferentialQuery method), 171

one() (EnrichedEventQuery method), 181

one() (EventQuery method), 384

one() (FacetQuery method), 135

one() (FeedQuery method), 226

one() (GrantQuery method), 395

one() (GroupedAlertSearchQuery method), 304

one() (IterableQueryMixin method), 558

one() (JobQuery method), 398

one() (ObservationQuery method), 421

[one\(\)](#) (*PaginatedQuery* method), 561  
[one\(\)](#) (*PolicyQuery* method), 438  
[one\(\)](#) (*Query* method), 564  
[one\(\)](#) (*RecommendationQuery* method), 191  
[one\(\)](#) (*ReportQuery* method), 237  
[one\(\)](#) (*ReputationOverrideQuery* method), 475  
[one\(\)](#) (*ResultQuery* method), 145  
[one\(\)](#) (*RunHistoryQuery* method), 155  
[one\(\)](#) (*SensorKitQuery* method), 512  
[one\(\)](#) (*SimpleQuery* method), 570  
[one\(\)](#) (*TemplateHistoryQuery* method), 166  
[one\(\)](#) (*USBDeviceApprovalQuery* method), 199  
[one\(\)](#) (*USBDeviceBlockQuery* method), 203  
[one\(\)](#) (*USBDeviceQuery* method), 205  
[one\(\)](#) (*UserQuery* method), 483  
[one\(\)](#) (*VCenterComputeResourceQuery* method), 539  
[one\(\)](#) (*VulnerabilityAssetViewQuery* method), 495  
[one\(\)](#) (*VulnerabilityQuery* method), 503  
[one\(\)](#) (*WatchlistQuery* method), 245  
[OpenKey\(\)](#) (in module *cbc\_sdk.credential\_providers.registry\_credential\_provider*), 175  
[OperationCancelled](#), 587  
[or\\_\(\)](#) (*AffectedAssetQuery* method), 486  
[or\\_\(\)](#) (*AlertSearchQuery* method), 261  
[or\\_\(\)](#) (*AssetGroupQuery* method), 352  
[or\\_\(\)](#) (*AsyncProcessQuery* method), 460  
[or\\_\(\)](#) (*AuthEventQuery* method), 216  
[or\\_\(\)](#) (*AWSComputeResourceQuery* method), 519  
[or\\_\(\)](#) (*BaseComputeResourceQuery* method), 528  
[or\\_\(\)](#) (*DeviceSearchQuery* method), 369  
[or\\_\(\)](#) (*EnrichedEventQuery* method), 181  
[or\\_\(\)](#) (*EventFacetQuery* method), 380  
[or\\_\(\)](#) (*EventQuery* method), 384  
[or\\_\(\)](#) (*FacetQuery* method), 136, 555  
[or\\_\(\)](#) (*GroupedAlertSearchQuery* method), 305  
[or\\_\(\)](#) (*ObservationQuery* method), 421  
[or\\_\(\)](#) (*Query* method), 564  
[or\\_\(\)](#) (*QueryBuilder* method), 568  
[or\\_\(\)](#) (*QueryBuilderSupportMixin* method), 569  
[or\\_\(\)](#) (*ReputationOverrideQuery* method), 476  
[or\\_\(\)](#) (*ResultQuery* method), 145  
[or\\_\(\)](#) (*RunHistoryQuery* method), 155  
[or\\_\(\)](#) (*SummaryQuery* method), 471  
[or\\_\(\)](#) (*TemplateHistoryQuery* method), 166  
[or\\_\(\)](#) (*USBDeviceApprovalQuery* method), 199  
[or\\_\(\)](#) (*USBDeviceQuery* method), 205  
[or\\_\(\)](#) (*VCenterComputeResourceQuery* method), 540  
[or\\_\(\)](#) (*VulnerabilityAssetViewQuery* method), 495  
[or\\_\(\)](#) (*VulnerabilityQuery* method), 503  
[org\\_urn](#) (*CBCloudAPI* property), 129  
[org\\_urn](#) (*User* property), 481

## P

[PaginatedQuery](#) (class in *cbc\_sdk.base*), 561  
[parameter\\_names](#) (*BypassRuleConfig* property), 442  
[parameter\\_names](#) (*CorePreventionRuleConfig* property), 444  
[parameter\\_names](#) (*DataCollectionRuleConfig* property), 447  
[parameter\\_names](#) (*HostBasedFirewallRuleConfig* property), 453  
[parameter\\_names](#) (*PolicyRuleConfig* property), 456  
[parents](#) (*Process* property), 467  
[pause\(\)](#) (*BackoffHandler.BackoffOperation* method), 613  
[perform\\_action\(\)](#) (*Vulnerability* method), 493  
[PlatformModel](#) (class in *cbc\_sdk.platform.base*), 249  
[Policy](#) (class in *cbc\_sdk.platform.policies*), 424  
[policy](#) (*Policy* property), 434  
[Policy.PolicyBuilder](#) (class in *cbc\_sdk.platform.policies*), 426  
[policy\\_id\(\)](#) (*RunQuery* method), 158  
[PolicyQuery](#) (class in *cbc\_sdk.platform.policies*), 437  
[PolicyRule](#) (class in *cbc\_sdk.platform.policies*), 439  
[PolicyRuleConfig](#) (class in *cbc\_sdk.platform.policy\_ruleconfigs*), 454  
[poll\\_status\(\)](#) (in module *cbc\_sdk.live\_response\_api*), 612  
[pop\(\)](#) (*Vulnerability.AssetView* method), 491  
[post\(\)](#) (*Connection* method), 581  
[post\\_multipart\(\)](#) (*BaseAPI* method), 575  
[post\\_multipart\(\)](#) (*CBCloudAPI* method), 129  
[post\\_object\(\)](#) (*BaseAPI* method), 576  
[post\\_object\(\)](#) (*CBCloudAPI* method), 129  
[preview\\_add\\_members\(\)](#) (*AssetGroup* method), 346  
[preview\\_add\\_members\\_to\\_groups\(\)](#) (*AssetGroup* class method), 347  
[preview\\_add\\_policy\\_override\(\)](#) (*Policy* method), 434  
[preview\\_add\\_policy\\_override\\_for\\_devices\(\)](#) (*Device* class method), 360  
[preview\\_create\\_asset\\_group\(\)](#) (*AssetGroup* class method), 347  
[preview\\_delete\(\)](#) (*AssetGroup* method), 347  
[preview\\_delete\\_asset\\_groups\(\)](#) (*AssetGroup* class method), 348  
[preview\\_policy\\_rank\\_changes\(\)](#) (*Policy* class method), 434  
[preview\\_rank\\_change\(\)](#) (*Policy* method), 435  
[preview\\_remove\\_members\(\)](#) (*AssetGroup* method), 348  
[preview\\_remove\\_members\\_from\\_groups\(\)](#) (*AssetGroup* class method), 348  
[preview\\_remove\\_policy\\_override\(\)](#) (*Device* method), 361

- preview\_remove\_policy\_override\_for\_devices() (Device class method), 361  
 preview\_save() (AssetGroup method), 349  
 preview\_update\_asset\_groups() (AssetGroup class method), 349  
 priorityLevel (Policy property), 435  
 Process (class in *cbc\_sdk.platform.processes*), 463  
 Process.Summary (class in *cbc\_sdk.platform.processes*), 464  
 Process.Tree (class in *cbc\_sdk.platform.processes*), 464  
 process\_limits() (CBCloudAPI method), 129  
 process\_md5 (Process property), 467  
 process\_pids (Process property), 467  
 process\_sha256 (EnrichedEvent property), 177  
 process\_sha256 (Process property), 467  
 ProcessFacet (class in *cbc\_sdk.platform.processes*), 468  
 ProcessFacet.Ranges (class in *cbc\_sdk.platform.processes*), 468  
 ProcessFacet.Terms (class in *cbc\_sdk.platform.processes*), 469  
 profiles\_ (Grant property), 393  
 proxy\_headers() (CBCSDKSessionAdapter method), 578  
 proxy\_manager\_for() (CBCSDKSessionAdapter method), 578  
 put() (Connection method), 581  
 put\_file() (CbLRSessionBase method), 597  
 put\_file() (LiveResponseSession method), 609  
 put\_object() (BaseAPI method), 576  
 put\_object() (CBCloudAPI method), 130
- ## Q
- quarantine() (Device method), 361  
 quarantine() (DeviceSearchQuery method), 369  
 Query (class in *cbc\_sdk.base*), 562  
 query\_device\_summaries() (Result method), 140  
 query\_device\_summaries() (Run method), 151  
 query\_device\_summaries() (RunHistory method), 152  
 query\_device\_summaries() (Template method), 161  
 query\_device\_summaries() (TemplateHistory method), 163  
 query\_device\_summary\_facets() (Result method), 141  
 query\_devices() (DeviceFacet.DeviceFacetValue method), 364  
 query\_facets() (Run method), 151  
 query\_facets() (RunHistory method), 152  
 query\_facets() (Template method), 161  
 query\_facets() (TemplateHistory method), 163  
 query\_result\_facets() (Result method), 141  
 query\_results() (Run method), 151  
 query\_results() (RunHistory method), 153  
 query\_results() (Template method), 162  
 query\_results() (TemplateHistory method), 164  
 query\_runs() (Template method), 162  
 query\_runs() (TemplateHistory method), 164  
 QueryBuilder (class in *cbc\_sdk.base*), 567  
 QueryBuilderSupportMixin (class in *cbc\_sdk.base*), 568  
 QuerySyntaxError, 587  
 QueryValueEx() (in module *cbc\_sdk.credential\_providers.registry\_credential\_provider*), 175
- ## R
- ranges\_ (AuthEventFacet property), 213  
 ranges\_ (EnrichedEventFacet property), 179  
 ranges\_ (EventFacet property), 377  
 ranges\_ (ObservationFacet property), 418  
 ranges\_ (ProcessFacet property), 470  
 RawErrorCode (class in *cbc\_sdk.winerror*), 616  
 read\_iocs() (in module *cbc\_sdk.helpers*), 590  
 Recommendation (class in *cbc\_sdk.endpoint\_standard.recommendation*), 184  
 Recommendation.RecommendationApplication (class in *cbc\_sdk.endpoint\_standard.recommendation*), 185  
 Recommendation.RecommendationImpact (class in *cbc\_sdk.endpoint\_standard.recommendation*), 185  
 Recommendation.RecommendationNewRule (class in *cbc\_sdk.endpoint\_standard.recommendation*), 186  
 Recommendation.RecommendationWorkflow (class in *cbc\_sdk.endpoint\_standard.recommendation*), 187  
 RecommendationQuery (class in *cbc\_sdk.endpoint\_standard.recommendation*), 190  
 refresh() (Alert method), 255  
 refresh() (Alert.Note method), 251  
 refresh() (AssetGroup method), 349  
 refresh() (AuditLog method), 354  
 refresh() (AuthEvent method), 210  
 refresh() (AuthEventFacet method), 213  
 refresh() (AuthEventFacet.Ranges method), 212  
 refresh() (AuthEventFacet.Terms method), 212  
 refresh() (AWSComputeResource method), 514  
 refresh() (BaseComputeResource method), 525  
 refresh() (Binary method), 247  
 refresh() (Binary.Summary method), 247  
 refresh() (BypassRuleConfig method), 442  
 refresh() (CBAnalyticsAlert method), 284  
 refresh() (CBAnalyticsAlert.Note method), 280

- `refresh()` (*ComputeResourceFacet* method), 531
- `refresh()` (*ComputeResourceFacet.ComputeResourceFacetValue* method), 530
- `refresh()` (*ContainerRuntimeAlert* method), 291
- `refresh()` (*ContainerRuntimeAlert.Note* method), 287
- `refresh()` (*CorePreventionRuleConfig* method), 444
- `refresh()` (*DataCollectionRuleConfig* method), 447
- `refresh()` (*Device* method), 361
- `refresh()` (*DeviceControlAlert* method), 298
- `refresh()` (*DeviceControlAlert.Note* method), 294
- `refresh()` (*DeviceFacet* method), 364
- `refresh()` (*DeviceFacet.DeviceFacetValue* method), 364
- `refresh()` (*DeviceSummary* method), 132
- `refresh()` (*DeviceSummary.Metrics* method), 132
- `refresh()` (*DeviceSummaryFacet* method), 133
- `refresh()` (*DeviceSummaryFacet.Values* method), 133
- `refresh()` (*Differential* method), 168
- `refresh()` (*Downloads* method), 249
- `refresh()` (*Downloads.FoundItem* method), 248
- `refresh()` (*EnrichedEvent* method), 177
- `refresh()` (*EnrichedEventFacet* method), 179
- `refresh()` (*EnrichedEventFacet.Ranges* method), 178
- `refresh()` (*EnrichedEventFacet.Terms* method), 178
- `refresh()` (*Event* method), 375
- `refresh()` (*EventFacet* method), 377
- `refresh()` (*EventFacet.Ranges* method), 376
- `refresh()` (*EventFacet.Terms* method), 376
- `refresh()` (*Feed* method), 222
- `refresh()` (*FeedModel* method), 224
- `refresh()` (*Grant* method), 393
- `refresh()` (*Grant.Profile* method), 390
- `refresh()` (*GroupedAlert* method), 301
- `refresh()` (*HostBasedFirewallAlert* method), 327
- `refresh()` (*HostBasedFirewallAlert.Note* method), 323
- `refresh()` (*HostBasedFirewallRuleConfig* method), 453
- `refresh()` (*HostBasedFirewallRuleConfig.FirewallRule* method), 449
- `refresh()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup* method), 451
- `refresh()` (*IntrusionDetectionSystemAlert* method), 334
- `refresh()` (*IntrusionDetectionSystemAlert.Note* method), 330
- `refresh()` (*IOC* method), 227
- `refresh()` (*IOC\_V2* method), 230
- `refresh()` (*Job* method), 397
- `refresh()` (*MutableBaseModel* method), 559
- `refresh()` (*NetworkThreatMetadata* method), 413
- `refresh()` (*NewBaseModel* method), 560
- `refresh()` (*Observation* method), 416
- `refresh()` (*ObservationFacet* method), 418
- `refresh()` (*ObservationFacet.Ranges* method), 417
- `refresh()` (*ObservationFacet.Terms* method), 417
- `refresh()` (*PlatformModel* method), 250
- `refresh()` (*Policy* method), 435
- `refresh()` (*PolicyRule* method), 440
- `refresh()` (*PolicyRuleConfig* method), 456
- `refresh()` (*Process* method), 467
- `refresh()` (*Process.Summary* method), 464
- `refresh()` (*Process.Tree* method), 465
- `refresh()` (*ProcessFacet* method), 470
- `refresh()` (*ProcessFacet.Ranges* method), 469
- `refresh()` (*ProcessFacet.Terms* method), 469
- `refresh()` (*Recommendation* method), 189
- `refresh()` (*Recommendation.RecommendationApplication* method), 185
- `refresh()` (*Recommendation.RecommendationImpact* method), 186
- `refresh()` (*Recommendation.RecommendationNewRule* method), 187
- `refresh()` (*Recommendation.RecommendationWorkflow* method), 188
- `refresh()` (*Report* method), 235
- `refresh()` (*ReportSeverity* method), 239
- `refresh()` (*ReputationOverride* method), 474
- `refresh()` (*Result* method), 141
- `refresh()` (*Result.Device* method), 139
- `refresh()` (*Result.Fields* method), 139
- `refresh()` (*Result.Metrics* method), 140
- `refresh()` (*ResultFacet* method), 142
- `refresh()` (*ResultFacet.Values* method), 142
- `refresh()` (*Run* method), 151
- `refresh()` (*RunHistory* method), 153
- `refresh()` (*SensorKit* method), 510
- `refresh()` (*Template* method), 162
- `refresh()` (*TemplateHistory* method), 164
- `refresh()` (*UnrefreshableModel* method), 571
- `refresh()` (*USBDevice* method), 194
- `refresh()` (*USBDeviceApproval* method), 197
- `refresh()` (*USBDeviceBlock* method), 202
- `refresh()` (*User* method), 481
- `refresh()` (*VCenterComputeResource* method), 533
- `refresh()` (*Vulnerability* method), 493
- `refresh()` (*Vulnerability.OrgSummary* method), 492
- `refresh()` (*Watchlist* method), 243
- `refresh()` (*WatchlistAlert* method), 341
- `refresh()` (*WatchlistAlert.Note* method), 337
- `RegistryCredentialProvider` (class in *cbc\_sdk.credential\_providers.registry\_credential\_provider*), 175
- `reject()` (*Recommendation* method), 189
- `remove()` (*HostBasedFirewallRuleConfig.FirewallRule* method), 449
- `remove()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup* method), 451



- `remove()` (*Vulnerability.AssetView method*), 492
- `remove_from_groups()` (*Device method*), 362
- `remove_from_groups_by_id()` (*Device method*), 362
- `remove_iocs()` (*Report method*), 235
- `remove_iocs_by_id()` (*Report method*), 235
- `remove_members()` (*AssetGroup method*), 350
- `replace_exclusions()` (*BypassRuleConfig method*), 442
- `replace_exclusions()` (*CorePreventionRuleConfig method*), 444
- `replace_reports()` (*Feed method*), 223
- `replace_reports_rawdata()` (*Feed method*), 223
- `replace_rule()` (*Policy method*), 435
- `replace_rule_config()` (*Policy method*), 435
- `Report` (*class in cbc\_sdk.enterprise\_edr.threat\_intelligence*), 231
- `Report.ReportBuilder` (*class in cbc\_sdk.enterprise\_edr.threat\_intelligence*), 232
- `ReportQuery` (*class in cbc\_sdk.enterprise\_edr.threat\_intelligence*), 237
- `reports` (*Feed property*), 223
- `reports` (*Watchlist property*), 243
- `ReportSeverity` (*class in cbc\_sdk.enterprise\_edr.threat\_intelligence*), 238
- `reputation_override()` (*Recommendation method*), 189
- `ReputationOverride` (*class in cbc\_sdk.platform.reputation*), 472
- `ReputationOverrideQuery` (*class in cbc\_sdk.platform.reputation*), 474
- `request_session()` (*CbLRManagerBase method*), 590
- `request_session()` (*LiveResponseSessionManager method*), 610
- `request_url()` (*CBCSDKSessionAdapter method*), 579
- `requires_boolean_value()` (*CredentialValue method*), 582
- `requires_integer_value()` (*CredentialValue method*), 582
- `reset()` (*AssetGroup method*), 350
- `reset()` (*BackoffHandler.BackoffOperation method*), 613
- `reset()` (*BypassRuleConfig method*), 442
- `reset()` (*CorePreventionRuleConfig method*), 445
- `reset()` (*DataCollectionRuleConfig method*), 447
- `reset()` (*Feed method*), 223
- `reset()` (*FeedModel method*), 225
- `reset()` (*Grant method*), 393
- `reset()` (*Grant.Profile method*), 390
- `reset()` (*HostBasedFirewallRuleConfig method*), 453
- `reset()` (*HostBasedFirewallRuleConfig.FirewallRule method*), 449
- `reset()` (*HostBasedFirewallRuleConfig.FirewallRuleConfig.FirewallRuleGroup method*), 451
- `reset()` (*IOC method*), 227
- `reset()` (*IOC\_V2 method*), 230
- `reset()` (*MutableBaseModel method*), 559
- `reset()` (*Policy method*), 436
- `reset()` (*PolicyRule method*), 440
- `reset()` (*PolicyRuleConfig method*), 456
- `reset()` (*Recommendation method*), 189
- `reset()` (*Report method*), 235
- `reset()` (*ReportSeverity method*), 239
- `reset()` (*USBDeviceApproval method*), 197
- `reset()` (*User method*), 481
- `reset()` (*Watchlist method*), 243
- `reset_google_authenticator_registration()` (*User method*), 481
- `Result` (*class in cbc\_sdk.audit\_remediation.base*), 138
- `Result.Device` (*class in cbc\_sdk.audit\_remediation.base*), 138
- `Result.Fields` (*class in cbc\_sdk.audit\_remediation.base*), 139
- `Result.Metrics` (*class in cbc\_sdk.audit\_remediation.base*), 139
- `ResultFacet` (*class in cbc\_sdk.audit\_remediation.base*), 141
- `ResultFacet.Values` (*class in cbc\_sdk.audit\_remediation.base*), 141
- `ResultFromCode()` (*in module cbc\_sdk.winerror*), 616
- `ResultQuery` (*class in cbc\_sdk.audit\_remediation.base*), 142
- `results` (*EventFacetQuery property*), 380
- `results` (*FacetQuery property*), 555
- `results` (*FeedQuery property*), 226
- `results` (*ReportQuery property*), 238
- `results` (*SimpleQuery property*), 570
- `results` (*SummaryQuery property*), 471
- `results` (*WatchlistQuery property*), 245
- `reverse()` (*Vulnerability.AssetView method*), 492
- `rule_groups` (*HostBasedFirewallRuleConfig property*), 454
- `rules_` (*HostBasedFirewallRuleConfig.FirewallRuleGroup property*), 451
- `Run` (*class in cbc\_sdk.audit\_remediation.base*), 149
- `run()` (*GetFileJob method*), 599
- `run()` (*JobWorker method*), 600
- `run()` (*LiveResponseJobScheduler method*), 602
- `run()` (*LRUCacheDict.EmptyCacheThread method*), 547
- `run_id()` (*FacetQuery method*), 136
- `run_id()` (*ResultQuery method*), 146
- `run_job()` (*JobWorker method*), 600
- `RunHistory` (*class in cbc\_sdk.audit\_remediation.base*), 152
- `RunHistoryQuery` (*class in cbc\_sdk.audit\_remediation.base*), 153

RunQuery (class in *cbc\_sdk.audit\_remediation.base*), 156

## S

save() (*AssetGroup* method), 350  
 save() (*BypassRuleConfig* method), 442  
 save() (*CorePreventionRuleConfig* method), 445  
 save() (*DataCollectionRuleConfig* method), 447  
 save() (*Feed* method), 223  
 save() (*FeedModel* method), 225  
 save() (*Grant* method), 393  
 save() (*Grant.Profile* method), 390  
 save() (*HostBasedFirewallRuleConfig* method), 454  
 save() (*HostBasedFirewallRuleConfig.FirewallRule* method), 449  
 save() (*HostBasedFirewallRuleConfig.FirewallRuleGroup* method), 451  
 save() (*IOC* method), 227  
 save() (*IOC\_V2* method), 230  
 save() (*MutableBaseModel* method), 559  
 save() (*Policy* method), 436  
 save() (*PolicyRule* method), 440  
 save() (*PolicyRuleConfig* method), 456  
 save() (*Report* method), 235  
 save() (*ReportSeverity* method), 239  
 save() (*USBDeviceApproval* method), 197  
 save() (*User* method), 481  
 save() (*Watchlist* method), 243  
 save\_watchlist() (*Report* method), 236  
 schedule() (*RunQuery* method), 158  
 SCORE\_CODE() (in module *cbc\_sdk.winerror*), 616  
 SCORE\_FACILITY() (in module *cbc\_sdk.winerror*), 616  
 SCORE\_SEVERITY() (in module *cbc\_sdk.winerror*), 616  
 scroll() (*DeviceSearchQuery* method), 369  
 scroll() (*ResultQuery* method), 146  
 search\_suggestions() (*Alert* static method), 255  
 search\_suggestions() (*AuthEvent* static method), 210  
 search\_suggestions() (*CBAnalyticsAlert* static method), 284  
 search\_suggestions() (*ContainerRuntimeAlert* static method), 291  
 search\_suggestions() (*DeviceControlAlert* static method), 298  
 search\_suggestions() (*HostBasedFirewallAlert* static method), 327  
 search\_suggestions() (*IntrusionDetectionSystemAlert* static method), 334  
 search\_suggestions() (*Observation* static method), 416  
 search\_suggestions() (*WatchlistAlert* static method), 341  
 select() (*BaseAPI* method), 576  
 select() (*CBCloudAPI* method), 130

select\_class\_instance() (in module *cbc\_sdk.connection*), 581  
 send() (*CBCSDKSessionAdapter* method), 579  
 SensorKit (class in *cbc\_sdk.workload.sensor\_lifecycle*), 509  
 SensorKitQuery (class in *cbc\_sdk.workload.sensor\_lifecycle*), 510  
 ServerError, 588, 629  
 session\_status() (*LiveResponseSessionManager* method), 611  
 set\_ad\_group\_ids() (*DeviceSearchQuery* method), 369  
 set\_alert\_ids() (*AlertSearchQuery* method), 261  
 set\_alert\_ids() (*GroupedAlertSearchQuery* method), 305  
 set\_alert\_ids() (*LegacyAlertSearchQueryCriterionMixin* method), 399  
 set\_alert\_notes\_present() (*AlertSearchQuery* method), 261  
 set\_alert\_notes\_present() (*GroupedAlertSearchQuery* method), 305  
 set\_alerttable() (*Feed.FeedBuilder* method), 220  
 set\_alerts\_enabled() (*Watchlist.WatchlistBuilder* method), 240  
 set\_appliance\_uuid() (*VCenterComputeResourceQuery* method), 540  
 set\_assignment\_mode() (*CorePreventionRuleConfig* method), 445  
 set\_auth\_event\_collection() (*Policy* method), 436  
 set\_auth\_method() (*User.UserBuilder* method), 478  
 set\_auto\_delete\_bad\_hash\_delay() (*Policy.PolicyBuilder* method), 428  
 set\_auto\_deregister\_interval() (*Policy.PolicyBuilder* method), 428  
 set\_auto\_scaling\_group\_name() (*AWSComputeResourceQuery* method), 520  
 set\_auto\_scaling\_group\_name() (*DeviceSearchQuery* method), 370  
 set\_availability\_zone() (*AWSComputeResourceQuery* method), 520  
 set\_avira\_protection\_cloud() (*Policy.PolicyBuilder* method), 428  
 set\_blocked\_threat\_categories() (*AlertSearchQuery* method), 262  
 set\_blocked\_threat\_categories() (*GroupedAlertSearchQuery* method), 305  
 set\_blocked\_threat\_categories() (*LegacyAlertSearchQueryCriterionMixin* method), 399  
 set\_categories() (*AlertSearchQuery* method), 262  
 set\_categories() (*GroupedAlertSearchQuery* method), 305  
 set\_categories() (*LegacyAlertSearchQueryCriterionMixin* method), 399  
 set\_category() (*Feed.FeedBuilder* method), 221

`set_cloud_provider_account_id()` (*AWSComputeResourceQuery method*), 520  
`set_cloud_provider_account_id()` (*DeviceSearchQuery method*), 370  
`set_cloud_provider_resource_id()` (*AWSComputeResourceQuery method*), 520  
`set_cloud_provider_tags()` (*AWSComputeResourceQuery method*), 520  
`set_cluster_name()` (*VCenterComputeResourceQuery method*), 540  
`set_cluster_names()` (*AlertSearchQuery method*), 262  
`set_cluster_names()` (*GroupedAlertSearchQuery method*), 306  
`set_cluster_names()` (*LegacyAlertSearchQueryCriterionMixin method*), 399  
`set_conditions()` (*Grant.ProfileBuilder method*), 391  
`set_create_time()` (*AlertSearchQuery method*), 262  
`set_create_time()` (*GroupedAlertSearchQuery method*), 306  
`set_create_time()` (*LegacyAlertSearchQueryCriterionMixin method*), 400  
`set_data_collection()` (*Policy method*), 436  
`set_datacenter_name()` (*VCenterComputeResourceQuery method*), 540  
`set_default_action()` (*HostBasedFirewallRuleConfig method*), 454  
`set_deployment_type()` (*AffectedAssetQuery method*), 486  
`set_deployment_type()` (*DeviceSearchQuery method*), 370  
`set_deployment_type()` (*VulnerabilityAssetViewQuery method*), 495  
`set_deployment_type()` (*VulnerabilityQuery method*), 503  
`set_description()` (*Policy.PolicyBuilder method*), 428  
`set_description()` (*Report.ReportBuilder method*), 232  
`set_description()` (*Watchlist.WatchlistBuilder method*), 241  
`set_device_guid()` (*VCenterComputeResourceQuery method*), 541  
`set_device_ids()` (*AlertSearchQuery method*), 262  
`set_device_ids()` (*DeviceSearchQuery method*), 370  
`set_device_ids()` (*DifferentialQuery method*), 171  
`set_device_ids()` (*FacetQuery method*), 136  
`set_device_ids()` (*GroupedAlertSearchQuery method*), 306  
`set_device_ids()` (*LegacyAlertSearchQueryCriterionMixin method*), 400  
`set_device_ids()` (*ResultQuery method*), 146  
`set_device_ids()` (*USBDeviceApprovalQuery method*), 200  
`set_device_locations()` (*AlertSearchQuery method*), 263  
`set_device_locations()` (*GroupedAlertSearchQuery method*), 307  
`set_device_locations()` (*LegacyAlertSearchQueryCriterionMixin method*), 400  
`set_device_names()` (*AlertSearchQuery method*), 263  
`set_device_names()` (*FacetQuery method*), 136  
`set_device_names()` (*GroupedAlertSearchQuery method*), 307  
`set_device_names()` (*LegacyAlertSearchQueryCriterionMixin method*), 400  
`set_device_names()` (*ResultQuery method*), 146  
`set_device_os()` (*AlertSearchQuery method*), 263  
`set_device_os()` (*FacetQuery method*), 136  
`set_device_os()` (*GroupedAlertSearchQuery method*), 307  
`set_device_os()` (*LegacyAlertSearchQueryCriterionMixin method*), 401  
`set_device_os()` (*ResultQuery method*), 146  
`set_device_os_versions()` (*AlertSearchQuery method*), 264  
`set_device_os_versions()` (*GroupedAlertSearchQuery method*), 307  
`set_device_os_versions()` (*LegacyAlertSearchQueryCriterionMixin method*), 401  
`set_device_type()` (*AffectedAssetQuery method*), 486  
`set_device_type()` (*VulnerabilityAssetViewQuery method*), 496  
`set_device_type()` (*VulnerabilityQuery method*), 503  
`set_device_username()` (*AlertSearchQuery method*), 264  
`set_device_username()` (*GroupedAlertSearchQuery method*), 308  
`set_device_username()` (*LegacyAlertSearchQueryCriterionMixin method*), 401  
`set_disabled()` (*Grant.Profile method*), 390  
`set_disabled()` (*Grant.ProfileBuilder method*), 391  
`set_egress_group_ids()` (*AlertSearchQuery method*), 264  
`set_egress_group_ids()` (*GroupedAlertSearchQuery method*), 308  
`set_egress_group_ids()` (*LegacyAlertSearchQueryCriterionMixin method*), 402  
`set_egress_group_names()` (*AlertSearchQuery method*), 264  
`set_egress_group_names()` (*GroupedAlertSearchQuery method*), 308  
`set_egress_group_names()` (*LegacyAlertSearchQueryCriterionMixin method*), 402  
`set_eligibility()` (*VCenterComputeResourceQuery method*), 541  
`set_eligibility_code()` (*VCenterComputeResourceQuery method*), 541

`set_email()` (*User.UserBuilder* method), 478  
`set_enabled()` (*HostBasedFirewallRuleConfig* method), 454  
`set_endpoint_names()` (*USBDeviceQuery* method), 206  
`set_esx_host_name()` (*VCenterComputeResourceQuery* method), 541  
`set_esx_host_uuid()` (*VCenterComputeResourceQuery* method), 541  
`set_exclude_sensor_versions()` (*DeviceSearchQuery* method), 370  
`set_expiration()` (*Grant.Profile* method), 390  
`set_expiration()` (*Grant.ProfileBuilder* method), 391  
`set_external_device_friendly_names()` (*AlertSearchQuery* method), 265  
`set_external_device_friendly_names()` (*GroupedAlertSearchQuery* method), 308  
`set_external_device_friendly_names()` (*LegacyAlertSearchQueryCriterionMixin* method), 402  
`set_external_device_ids()` (*AlertSearchQuery* method), 265  
`set_external_device_ids()` (*GroupedAlertSearchQuery* method), 309  
`set_external_device_ids()` (*LegacyAlertSearchQueryCriterionMixin* method), 402  
`set_fields()` (*AsyncProcessQuery* method), 461  
`set_fields()` (*AuthEventQuery* method), 217  
`set_fields()` (*EnrichedEventQuery* method), 181  
`set_fields()` (*EventQuery* method), 385  
`set_fields()` (*ObservationQuery* method), 421  
`set_fields()` (*Query* method), 565  
`set_first_name()` (*User.UserBuilder* method), 478  
`set_group_by()` (*AlertSearchQuery* method), 265  
`set_group_by()` (*GroupedAlertSearchQuery* method), 309  
`set_group_results()` (*AlertSearchQuery* method), 265  
`set_group_results()` (*GroupedAlertSearchQuery* method), 309  
`set_group_results()` (*LegacyAlertSearchQueryCriterionMixin* method), 403  
`set_hashes()` (*RecommendationQuery* method), 191  
`set_highest_risk_score()` (*AffectedAssetQuery* method), 486  
`set_highest_risk_score()` (*VulnerabilityAssetViewQuery* method), 496  
`set_highest_risk_score()` (*VulnerabilityQuery* method), 503  
`set_host_name()` (*VCenterComputeResourceQuery* method), 541  
`set_id()` (*AWSComputeResourceQuery* method), 521  
`set_installation_status()` (*AWSComputeResourceQuery* method), 521  
`set_installation_status()` (*VCenterComputeResourceQuery* method), 542  
`set_installation_type()` (*VCenterComputeResourceQuery* method), 542  
`set_ip_address()` (*VCenterComputeResourceQuery* method), 542  
`set_ip_reputations()` (*AlertSearchQuery* method), 265  
`set_ip_reputations()` (*GroupedAlertSearchQuery* method), 309  
`set_ip_reputations()` (*LegacyAlertSearchQueryCriterionMixin* method), 403  
`set_kill_chain_statuses()` (*AlertSearchQuery* method), 266  
`set_kill_chain_statuses()` (*GroupedAlertSearchQuery* method), 309  
`set_kill_chain_statuses()` (*LegacyAlertSearchQueryCriterionMixin* method), 403  
`set_last_contact_time()` (*DeviceSearchQuery* method), 371  
`set_last_name()` (*User.UserBuilder* method), 478  
`set_last_sync_ts()` (*AffectedAssetQuery* method), 487  
`set_last_sync_ts()` (*VulnerabilityAssetViewQuery* method), 496  
`set_last_sync_ts()` (*VulnerabilityQuery* method), 504  
`set_legacy_alert_ids()` (*AlertSearchQuery* method), 266  
`set_legacy_alert_ids()` (*GroupedAlertSearchQuery* method), 310  
`set_legacy_alert_ids()` (*LegacyAlertSearchQueryCriterionMixin* method), 403  
`set_link()` (*Report.ReportBuilder* method), 232  
`set_managed_detection_response_permissions()` (*Policy.PolicyBuilder* method), 428  
`set_max_rows()` (*DeviceSearchQuery* method), 371  
`set_max_rows()` (*USBDeviceQuery* method), 206  
`set_minimum_severity()` (*AlertSearchQuery* method), 266  
`set_minimum_severity()` (*GroupedAlertSearchQuery* method), 310  
`set_name()` (*AffectedAssetQuery* method), 487  
`set_name()` (*AWSComputeResourceQuery* method), 521  
`set_name()` (*Feed.FeedBuilder* method), 221  
`set_name()` (*Policy.PolicyBuilder* method), 428  
`set_name()` (*VCenterComputeResourceQuery* method), 542  
`set_name()` (*VulnerabilityAssetViewQuery* method), 496  
`set_name()` (*VulnerabilityQuery* method), 504  
`set_name()` (*Watchlist.WatchlistBuilder* method), 241  
`set_namespaces()` (*AlertSearchQuery* method), 266  
`set_namespaces()` (*GroupedAlertSearchQuery* method), 310



set\_namespaces() (*LegacyAlertSearchQueryCriterionMixin* method), 403  
 set\_not\_blocked\_threat\_categories() (*AlertSearchQuery* method), 267  
 set\_not\_blocked\_threat\_categories() (*GroupedAlertSearchQuery* method), 310  
 set\_not\_blocked\_threat\_categories() (*LegacyAlertSearchQueryCriterionMixin* method), 404  
 set\_on\_access\_scan() (*Policy.PolicyBuilder* method), 429  
 set\_on\_demand\_scan() (*Policy.PolicyBuilder* method), 429  
 set\_on\_demand\_scan\_schedule() (*Policy.PolicyBuilder* method), 429  
 set\_org() (*Grant.GrantBuilder* method), 388  
 set\_orgs() (*Grant.ProfileBuilder* method), 392  
 set\_os() (*DeviceSearchQuery* method), 371  
 set\_os\_arch() (*AffectedAssetQuery* method), 487  
 set\_os\_arch() (*VulnerabilityAssetViewQuery* method), 497  
 set\_os\_arch() (*VulnerabilityQuery* method), 504  
 set\_os\_architecture() (*VCenterComputeResourceQuery* method), 542  
 set\_os\_description() (*VCenterComputeResourceQuery* method), 543  
 set\_os\_name() (*AffectedAssetQuery* method), 487  
 set\_os\_name() (*VulnerabilityAssetViewQuery* method), 497  
 set\_os\_name() (*VulnerabilityQuery* method), 504  
 set\_os\_product\_id() (*AffectedAssetQuery* method), 487  
 set\_os\_type() (*AffectedAssetQuery* method), 488  
 set\_os\_type() (*VCenterComputeResourceQuery* method), 543  
 set\_os\_type() (*VulnerabilityAssetViewQuery* method), 497  
 set\_os\_type() (*VulnerabilityQuery* method), 505  
 set\_os\_version() (*AffectedAssetQuery* method), 488  
 set\_os\_version() (*VulnerabilityAssetViewQuery* method), 497  
 set\_os\_version() (*VulnerabilityQuery* method), 505  
 set\_override\_list() (*ReputationOverrideQuery* method), 476  
 set\_override\_type() (*ReputationOverrideQuery* method), 476  
 set\_parameter() (*BypassRuleConfig* method), 442  
 set\_parameter() (*CorePreventionRuleConfig* method), 445  
 set\_parameter() (*DataCollectionRuleConfig* method), 447  
 set\_parameter() (*HostBasedFirewallRuleConfig* method), 454  
 set\_parameter() (*PolicyRuleConfig* method), 456  
 set\_phone() (*User.UserBuilder* method), 479  
 set\_platform() (*AWSComputeResourceQuery* method), 521  
 set\_platform\_details() (*AWSComputeResourceQuery* method), 521  
 set\_policy\_applied() (*AlertSearchQuery* method), 267  
 set\_policy\_applied() (*GroupedAlertSearchQuery* method), 310  
 set\_policy\_applied() (*LegacyAlertSearchQueryCriterionMixin* method), 404  
 set\_policy\_ids() (*AlertSearchQuery* method), 267  
 set\_policy\_ids() (*DeviceSearchQuery* method), 371  
 set\_policy\_ids() (*FacetQuery* method), 137  
 set\_policy\_ids() (*GroupedAlertSearchQuery* method), 311  
 set\_policy\_ids() (*LegacyAlertSearchQueryCriterionMixin* method), 404  
 set\_policy\_ids() (*ResultQuery* method), 147  
 set\_policy\_names() (*AlertSearchQuery* method), 267  
 set\_policy\_names() (*FacetQuery* method), 137  
 set\_policy\_names() (*GroupedAlertSearchQuery* method), 311  
 set\_policy\_names() (*LegacyAlertSearchQueryCriterionMixin* method), 404  
 set\_policy\_names() (*ResultQuery* method), 147  
 set\_policy\_types() (*RecommendationQuery* method), 191  
 set\_ports() (*AlertSearchQuery* method), 268  
 set\_ports() (*GroupedAlertSearchQuery* method), 311  
 set\_ports() (*LegacyAlertSearchQueryCriterionMixin* method), 405  
 set\_principal\_name() (*Grant.GrantBuilder* method), 388  
 set\_priority() (*Policy.PolicyBuilder* method), 429  
 set\_process\_names() (*AlertSearchQuery* method), 268  
 set\_process\_names() (*GroupedAlertSearchQuery* method), 311  
 set\_process\_names() (*LegacyAlertSearchQueryCriterionMixin* method), 405  
 set\_process\_sha256() (*AlertSearchQuery* method), 268  
 set\_process\_sha256() (*GroupedAlertSearchQuery* method), 312  
 set\_process\_sha256() (*LegacyAlertSearchQueryCriterionMixin* method), 405  
 set\_product\_ids() (*AlertSearchQuery* method), 268  
 set\_product\_ids() (*GroupedAlertSearchQuery* method), 312  
 set\_product\_ids() (*LegacyAlertSearchQueryCriterionMixin* method), 405  
 set\_product\_names() (*AlertSearchQuery* method), 269

`set_product_names()` (*GroupedAlertSearchQuery method*), 312

`set_product_names()` (*LegacyAlertSearchQueryCriterionMixin method*), 406

`set_product_names()` (*USBDeviceApprovalQuery method*), 200

`set_product_names()` (*USBDeviceQuery method*), 206

`set_profile_expiration()` (*User method*), 481

`set_protocols()` (*AlertSearchQuery method*), 269

`set_protocols()` (*GroupedAlertSearchQuery method*), 313

`set_protocols()` (*LegacyAlertSearchQueryCriterionMixin method*), 406

`set_provider_url()` (*Feed.FeedBuilder method*), 221

`set_reason_code()` (*AlertSearchQuery method*), 269

`set_reason_code()` (*GroupedAlertSearchQuery method*), 313

`set_reason_code()` (*LegacyAlertSearchQueryCriterionMixin method*), 406

`set_region()` (*AWSComputeResourceQuery method*), 522

`set_registration_id()` (*VCenterComputeResourceQuery method*), 543

`set_registry_value()` (*CbLRSessionBase method*), 597

`set_registry_value()` (*LiveResponseSession method*), 609

`set_remote_domains()` (*AlertSearchQuery method*), 269

`set_remote_domains()` (*GroupedAlertSearchQuery method*), 313

`set_remote_domains()` (*LegacyAlertSearchQueryCriterionMixin method*), 406

`set_remote_ips()` (*AlertSearchQuery method*), 270

`set_remote_ips()` (*GroupedAlertSearchQuery method*), 313

`set_remote_ips()` (*LegacyAlertSearchQueryCriterionMixin method*), 407

`set_remote_is_private()` (*AlertSearchQuery method*), 270

`set_remote_is_private()` (*GroupedAlertSearchQuery method*), 314

`set_replica_ids()` (*AlertSearchQuery method*), 270

`set_replica_ids()` (*GroupedAlertSearchQuery method*), 314

`set_replica_ids()` (*LegacyAlertSearchQueryCriterionMixin method*), 407

`set_reputations()` (*AlertSearchQuery method*), 270

`set_reputations()` (*GroupedAlertSearchQuery method*), 314

`set_reputations()` (*LegacyAlertSearchQueryCriterionMixin method*), 407

`set_role()` (*User.UserBuilder method*), 479

`set_roles()` (*Grant.GrantBuilder method*), 389

`set_roles()` (*Grant.ProfileBuilder method*), 392

`set_rows()` (*AlertSearchQuery method*), 271

`set_rows()` (*AssetGroupQuery method*), 352

`set_rows()` (*AsyncProcessQuery method*), 461

`set_rows()` (*AuthEventQuery method*), 217

`set_rows()` (*EnrichedEventQuery method*), 182

`set_rows()` (*EventFacetQuery method*), 380

`set_rows()` (*EventQuery method*), 385

`set_rows()` (*FacetQuery method*), 555

`set_rows()` (*GroupedAlertSearchQuery method*), 314

`set_rows()` (*ObservationQuery method*), 422

`set_rows()` (*Query method*), 565

`set_rule_ids()` (*AlertSearchQuery method*), 271

`set_rule_ids()` (*GroupedAlertSearchQuery method*), 314

`set_rule_ids()` (*LegacyAlertSearchQueryCriterionMixin method*), 407

`set_rule_names()` (*AlertSearchQuery method*), 271

`set_rule_names()` (*GroupedAlertSearchQuery method*), 315

`set_rule_names()` (*LegacyAlertSearchQueryCriterionMixin method*), 408

`set_run_ids()` (*ResultQuery method*), 147

`set_run_states()` (*AlertSearchQuery method*), 271

`set_run_states()` (*GroupedAlertSearchQuery method*), 315

`set_run_states()` (*LegacyAlertSearchQueryCriterionMixin method*), 408

`set_sensor_actions()` (*AlertSearchQuery method*), 272

`set_sensor_actions()` (*GroupedAlertSearchQuery method*), 315

`set_sensor_actions()` (*LegacyAlertSearchQueryCriterionMixin method*), 408

`set_serial_numbers()` (*AlertSearchQuery method*), 272

`set_serial_numbers()` (*GroupedAlertSearchQuery method*), 316

`set_serial_numbers()` (*LegacyAlertSearchQueryCriterionMixin method*), 409

`set_serial_numbers()` (*USBDeviceQuery method*), 206

`set_severity()` (*AffectedAssetQuery method*), 488

`set_severity()` (*Report.ReportBuilder method*), 233

`set_severity()` (*VulnerabilityAssetViewQuery method*), 498

`set_severity()` (*VulnerabilityOrgSummaryQuery method*), 500

`set_severity()` (*VulnerabilityQuery method*), 505

`set_signature_update()` (*Policy.PolicyBuilder method*), 430

`set_signature_update_schedule()` (*Policy.PolicyBuilder method*), 430

- `set_source_label()` (*Feed.FeedBuilder* method), 221
- `set_start()` (*AsyncProcessQuery* method), 461
- `set_start()` (*AuthEventQuery* method), 217
- `set_start()` (*EnrichedEventQuery* method), 182
- `set_start()` (*EventQuery* method), 385
- `set_start()` (*ObservationQuery* method), 422
- `set_start()` (*Query* method), 565
- `set_status()` (*DeviceSearchQuery* method), 371
- `set_statuses()` (*FacetQuery* method), 137
- `set_statuses()` (*RecommendationQuery* method), 191
- `set_statuses()` (*ResultQuery* method), 147
- `set_statuses()` (*USBDeviceQuery* method), 207
- `set_subnet_id()` (*AWSComputeResourceQuery* method), 522
- `set_summary()` (*Feed.FeedBuilder* method), 221
- `set_sync_status()` (*AffectedAssetQuery* method), 488
- `set_sync_status()` (*VulnerabilityAssetViewQuery* method), 498
- `set_sync_status()` (*VulnerabilityQuery* method), 505
- `set_sync_type()` (*AffectedAssetQuery* method), 489
- `set_sync_type()` (*VulnerabilityAssetViewQuery* method), 498
- `set_sync_type()` (*VulnerabilityQuery* method), 506
- `set_system()` (*PolicyQuery* method), 439
- `set_tags()` (*AlertSearchQuery* method), 272
- `set_tags()` (*GroupedAlertSearchQuery* method), 316
- `set_tags()` (*LegacyAlertSearchQueryCriterionMixin* method), 409
- `set_tags_enabled()` (*Watchlist.WatchlistBuilder* method), 241
- `set_target_priorities()` (*AlertSearchQuery* method), 272
- `set_target_priorities()` (*DeviceSearchQuery* method), 372
- `set_target_priorities()` (*GroupedAlertSearchQuery* method), 316
- `set_target_priorities()` (*LegacyAlertSearchQueryCriterionMixin* method), 409
- `set_template_ids()` (*RunHistoryQuery* method), 155
- `set_threat_cause_vectors()` (*AlertSearchQuery* method), 273
- `set_threat_cause_vectors()` (*GroupedAlertSearchQuery* method), 316
- `set_threat_cause_vectors()` (*LegacyAlertSearchQueryCriterionMixin* method), 409
- `set_threat_ids()` (*AlertSearchQuery* method), 273
- `set_threat_ids()` (*GroupedAlertSearchQuery* method), 316
- `set_threat_ids()` (*LegacyAlertSearchQueryCriterionMixin* method), 410
- `set_threat_notes_present()` (*AlertSearchQuery* method), 273
- `set_threat_notes_present()` (*GroupedAlertSearchQuery* method), 317
- `set_time_range()` (*AlertSearchQuery* method), 273
- `set_time_range()` (*AsyncProcessQuery* method), 461
- `set_time_range()` (*AuthEventQuery* method), 217
- `set_time_range()` (*EnrichedEventQuery* method), 182
- `set_time_range()` (*EventFacetQuery* method), 380
- `set_time_range()` (*EventQuery* method), 385
- `set_time_range()` (*FacetQuery* method), 555
- `set_time_range()` (*GroupedAlertSearchQuery* method), 317
- `set_time_range()` (*ObservationQuery* method), 422
- `set_time_range()` (*Query* method), 565
- `set_time_range()` (*SummaryQuery* method), 471
- `set_time_received()` (*ResultQuery* method), 148
- `set_timestamp()` (*Report.ReportBuilder* method), 233
- `set_title()` (*Report.ReportBuilder* method), 233
- `set_types()` (*AlertSearchQuery* method), 274
- `set_types()` (*GroupedAlertSearchQuery* method), 318
- `set_types()` (*LegacyAlertSearchQueryCriterionMixin* method), 410
- `set_update_servers_offsite()` (*Policy.PolicyBuilder* method), 430
- `set_update_servers_onsite()` (*Policy.PolicyBuilder* method), 430
- `set_update_servers_override()` (*Policy.PolicyBuilder* method), 430
- `set_uuid()` (*VCenterComputeResourceQuery* method), 543
- `set_vcenter()` (*AffectedAssetQuery* method), 489
- `set_vcenter()` (*VulnerabilityAssetViewQuery* method), 498
- `set_vcenter()` (*VulnerabilityOrgSummaryQuery* method), 500
- `set_vcenter()` (*VulnerabilityQuery* method), 506
- `set_vcenter_host_url()` (*VCenterComputeResourceQuery* method), 543
- `set_vcenter_name()` (*VCenterComputeResourceQuery* method), 544
- `set_vcenter_uuid()` (*VCenterComputeResourceQuery* method), 544
- `set_vendor_ids()` (*AlertSearchQuery* method), 275
- `set_vendor_ids()` (*GroupedAlertSearchQuery* method), 318
- `set_vendor_ids()` (*LegacyAlertSearchQueryCriterionMixin* method), 410
- `set_vendor_names()` (*AlertSearchQuery* method), 275
- `set_vendor_names()` (*GroupedAlertSearchQuery* method), 319
- `set_vendor_names()` (*LegacyAlertSearchQueryCriterionMixin* method), 410
- `set_vendor_names()` (*USBDeviceApprovalQuery* method), 200
- `set_vendor_names()` (*USBDeviceQuery* method), 207
- `set_virtual_private_cloud_id()` (*AWSComputeResourceQuery* method), 522

`set_virtual_private_cloud_id()` (*DeviceSearchQuery method*), 372

`set_visibility()` (*AffectedAssetQuery method*), 489

`set_visibility()` (*Report.ReportBuilder method*), 233

`set_visibility()` (*VulnerabilityAssetViewQuery method*), 498

`set_visibility()` (*VulnerabilityOrgSummaryQuery method*), 500

`set_visibility()` (*VulnerabilityQuery method*), 506

`set_vm_id()` (*AffectedAssetQuery method*), 489

`set_vm_id()` (*VulnerabilityAssetViewQuery method*), 499

`set_vm_id()` (*VulnerabilityQuery method*), 506

`set_vmwaretools_version()` (*VCenterComputeResourceQuery method*), 544

`set_vuln_count()` (*AffectedAssetQuery method*), 490

`set_vuln_count()` (*VulnerabilityAssetViewQuery method*), 499

`set_vuln_count()` (*VulnerabilityQuery method*), 506

`set_watchlist_ids()` (*AlertSearchQuery method*), 275

`set_watchlist_ids()` (*GroupedAlertSearchQuery method*), 319

`set_watchlist_ids()` (*LegacyAlertSearchQueryCriterionMixin method*), 411

`set_watchlist_names()` (*AlertSearchQuery method*), 276

`set_watchlist_names()` (*GroupedAlertSearchQuery method*), 319

`set_watchlist_names()` (*LegacyAlertSearchQueryCriterionMixin method*), 411

`set_workflows()` (*AlertSearchQuery method*), 276

`set_workflows()` (*GroupedAlertSearchQuery method*), 319

`set_workflows()` (*LegacyAlertSearchQueryCriterionMixin method*), 411

`set_workload_ids()` (*AlertSearchQuery method*), 276

`set_workload_ids()` (*GroupedAlertSearchQuery method*), 320

`set_workload_ids()` (*LegacyAlertSearchQueryCriterionMixin method*), 412

`set_workload_kinds()` (*AlertSearchQuery method*), 276

`set_workload_kinds()` (*GroupedAlertSearchQuery method*), 320

`set_workload_kinds()` (*LegacyAlertSearchQueryCriterionMixin method*), 412

`set_workload_names()` (*AlertSearchQuery method*), 276

`set_workload_names()` (*GroupedAlertSearchQuery method*), 320

`set_workload_names()` (*LegacyAlertSearchQueryCriterionMixin method*), 412

`set_xdr_collection()` (*Policy method*), 436

`setDaemon()` (*JobWorker method*), 600

`setDaemon()` (*LiveResponseJobScheduler method*), 602

`setDaemon()` (*LRUCacheDict.EmptyCacheThread method*), 548

`setName()` (*JobWorker method*), 600

`setName()` (*LiveResponseJobScheduler method*), 602

`setName()` (*LRUCacheDict.EmptyCacheThread method*), 548

`severity_levels()` (*Vulnerability.OrgSummary method*), 492

`siblings` (*Process property*), 467

`SimpleQuery` (*class in cbc\_sdk.base*), 569

`sort()` (*FeedQuery method*), 226

`sort()` (*ReportQuery method*), 238

`sort()` (*SimpleQuery method*), 570

`sort()` (*Vulnerability.AssetView method*), 492

`sort()` (*WatchlistQuery method*), 245

`sort_by()` (*AffectedAssetQuery method*), 490

`sort_by()` (*AlertSearchQuery method*), 277

`sort_by()` (*AssetGroupQuery method*), 353

`sort_by()` (*AsyncProcessQuery method*), 461

`sort_by()` (*AuthEventQuery method*), 218

`sort_by()` (*AWSComputeResourceQuery method*), 522

`sort_by()` (*BaseComputeResourceQuery method*), 529

`sort_by()` (*DeviceSearchQuery method*), 372

`sort_by()` (*EnrichedEventQuery method*), 182

`sort_by()` (*EventQuery method*), 385

`sort_by()` (*GroupedAlertSearchQuery method*), 320

`sort_by()` (*ObservationQuery method*), 422

`sort_by()` (*Query method*), 565

`sort_by()` (*RecommendationQuery method*), 192

`sort_by()` (*ReputationOverrideQuery method*), 476

`sort_by()` (*ResultQuery method*), 148

`sort_by()` (*RunHistoryQuery method*), 155

`sort_by()` (*TemplateHistoryQuery method*), 166

`sort_by()` (*USBDeviceQuery method*), 207

`sort_by()` (*VCenterComputeResourceQuery method*), 544

`sort_by()` (*VulnerabilityAssetViewQuery method*), 499

`sort_by()` (*VulnerabilityQuery method*), 507

`start()` (*JobWorker method*), 600

`start()` (*LiveResponseJobScheduler method*), 602

`start()` (*LRUCacheDict.EmptyCacheThread method*), 548

`start_memdump()` (*CbLRSessionBase method*), 597

`start_memdump()` (*LiveResponseSession method*), 609

`start_request()` (*NSXRemediationJob class method*), 508

`status` (*NSXRemediationJob property*), 509

`stop()` (*Run method*), 151

`stop()` (*RunHistory method*), 153

`stop()` (*Template method*), 162

`stop()` (*TemplateHistory method*), 164



- [stop\\_keepalive\\_thread\(\)](#) (*CbLRManagerBase method*), 591  
[stop\\_keepalive\\_thread\(\)](#) (*LiveResponseSessionManager method*), 611  
[submit\(\)](#) (*DifferentialQuery method*), 172  
[submit\(\)](#) (*RunQuery method*), 159  
[submit\(\)](#) (*VulnerabilityOrgSummaryQuery method*), 501  
[submit\\_job\(\)](#) (*CbLRManagerBase method*), 591  
[submit\\_job\(\)](#) (*LiveResponseJobScheduler method*), 602  
[submit\\_job\(\)](#) (*LiveResponseSessionManager method*), 611  
[SUCCEEDED\(\)](#) (in module *cbc\_sdk.winerror*), 616  
[summarize\(\)](#) (*AWSComputeResourceQuery method*), 523  
[summary](#) (*Binary property*), 248  
[summary](#) (*Process property*), 467  
[SummaryQuery](#) (class in *cbc\_sdk.platform.processes*), 470  
[swagger\\_meta\\_file](#) (*AssetGroup attribute*), 350  
[swagger\\_meta\\_file](#) (*Device attribute*), 362  
[SwaggerLoader](#) (class in *cbc\_sdk.base*), 571  
[systemPolicy](#) (*Policy property*), 436
- ## T
- [Template](#) (class in *cbc\_sdk.audit\_remediation.base*), 159  
[TemplateHistory](#) (class in *cbc\_sdk.audit\_remediation.base*), 162  
[TemplateHistoryQuery](#) (class in *cbc\_sdk.audit\_remediation.base*), 164  
[terms\\_](#) (*AuthEventFacet property*), 213  
[terms\\_](#) (*EnrichedEventFacet property*), 179  
[terms\\_](#) (*EventFacet property*), 377  
[terms\\_](#) (*ObservationFacet property*), 418  
[terms\\_](#) (*ProcessFacet property*), 470  
[timeout](#) (*BackoffHandler property*), 613  
[timeout\(\)](#) (*AsyncProcessQuery method*), 462  
[timeout\(\)](#) (*AuthEventQuery method*), 218  
[timeout\(\)](#) (*EnrichedEventQuery method*), 183  
[timeout\(\)](#) (*EventFacetQuery method*), 381  
[timeout\(\)](#) (*FacetQuery method*), 556  
[timeout\(\)](#) (*ObservationQuery method*), 423  
[timeout\(\)](#) (*SummaryQuery method*), 472  
[TimeoutError](#), 588, 630  
[to\\_dict\(\)](#) (*Credentials method*), 583  
[to\\_json\(\)](#) (*Alert method*), 256  
[to\\_json\(\)](#) (*Alert.Note method*), 251  
[to\\_json\(\)](#) (*AssetGroup method*), 350  
[to\\_json\(\)](#) (*AuditLog method*), 354  
[to\\_json\(\)](#) (*AuthEvent method*), 211  
[to\\_json\(\)](#) (*AuthEventFacet method*), 213  
[to\\_json\(\)](#) (*AuthEventFacet.Ranges method*), 212  
[to\\_json\(\)](#) (*AuthEventFacet.Terms method*), 212  
[to\\_json\(\)](#) (*AWSComputeResource method*), 514  
[to\\_json\(\)](#) (*BaseComputeResource method*), 525  
[to\\_json\(\)](#) (*Binary method*), 248  
[to\\_json\(\)](#) (*Binary.Summary method*), 247  
[to\\_json\(\)](#) (*BypassRuleConfig method*), 442  
[to\\_json\(\)](#) (*CBAnalyticsAlert method*), 285  
[to\\_json\(\)](#) (*CBAnalyticsAlert.Note method*), 280  
[to\\_json\(\)](#) (*ComputeResourceFacet method*), 531  
[to\\_json\(\)](#) (*ComputeResourceFacet.ComputeResourceFacetValue method*), 530  
[to\\_json\(\)](#) (*ContainerRuntimeAlert method*), 291  
[to\\_json\(\)](#) (*ContainerRuntimeAlert.Note method*), 287  
[to\\_json\(\)](#) (*CorePreventionRuleConfig method*), 445  
[to\\_json\(\)](#) (*DataCollectionRuleConfig method*), 447  
[to\\_json\(\)](#) (*Device method*), 362  
[to\\_json\(\)](#) (*DeviceControlAlert method*), 298  
[to\\_json\(\)](#) (*DeviceControlAlert.Note method*), 294  
[to\\_json\(\)](#) (*DeviceFacet method*), 364  
[to\\_json\(\)](#) (*DeviceFacet.DeviceFacetValue method*), 364  
[to\\_json\(\)](#) (*DeviceSummary method*), 132  
[to\\_json\(\)](#) (*DeviceSummary.Metrics method*), 132  
[to\\_json\(\)](#) (*DeviceSummaryFacet method*), 133  
[to\\_json\(\)](#) (*DeviceSummaryFacet.Values method*), 133  
[to\\_json\(\)](#) (*Differential method*), 168  
[to\\_json\(\)](#) (*Downloads method*), 249  
[to\\_json\(\)](#) (*Downloads.FoundItem method*), 248  
[to\\_json\(\)](#) (*EnrichedEvent method*), 177  
[to\\_json\(\)](#) (*EnrichedEventFacet method*), 179  
[to\\_json\(\)](#) (*EnrichedEventFacet.Ranges method*), 178  
[to\\_json\(\)](#) (*EnrichedEventFacet.Terms method*), 178  
[to\\_json\(\)](#) (*Event method*), 375  
[to\\_json\(\)](#) (*EventFacet method*), 377  
[to\\_json\(\)](#) (*EventFacet.Ranges method*), 376  
[to\\_json\(\)](#) (*EventFacet.Terms method*), 376  
[to\\_json\(\)](#) (*Feed method*), 223  
[to\\_json\(\)](#) (*FeedModel method*), 225  
[to\\_json\(\)](#) (*Grant method*), 394  
[to\\_json\(\)](#) (*Grant.Profile method*), 390  
[to\\_json\(\)](#) (*GroupedAlert method*), 301  
[to\\_json\(\)](#) (*HostBasedFirewallAlert method*), 327  
[to\\_json\(\)](#) (*HostBasedFirewallAlert.Note method*), 323  
[to\\_json\(\)](#) (*HostBasedFirewallRuleConfig method*), 454  
[to\\_json\(\)](#) (*HostBasedFirewallRuleConfig.FirewallRule method*), 449  
[to\\_json\(\)](#) (*HostBasedFirewallRuleConfig.FirewallRuleGroup method*), 451  
[to\\_json\(\)](#) (*IntrusionDetectionSystemAlert method*), 334  
[to\\_json\(\)](#) (*IntrusionDetectionSystemAlert.Note method*), 330  
[to\\_json\(\)](#) (*IOC method*), 227  
[to\\_json\(\)](#) (*IOC\_V2 method*), 231

`to_json()` (*Job method*), 398  
`to_json()` (*MutableBaseModel method*), 559  
`to_json()` (*NetworkThreatMetadata method*), 413  
`to_json()` (*NewBaseModel method*), 560  
`to_json()` (*Observation method*), 416  
`to_json()` (*ObservationFacet method*), 418  
`to_json()` (*ObservationFacet.Ranges method*), 417  
`to_json()` (*ObservationFacet.Terms method*), 417  
`to_json()` (*PlatformModel method*), 250  
`to_json()` (*Policy method*), 436  
`to_json()` (*PolicyRule method*), 440  
`to_json()` (*PolicyRuleConfig method*), 456  
`to_json()` (*Process method*), 467  
`to_json()` (*Process.Summary method*), 464  
`to_json()` (*Process.Tree method*), 465  
`to_json()` (*ProcessFacet method*), 470  
`to_json()` (*ProcessFacet.Ranges method*), 469  
`to_json()` (*ProcessFacet.Terms method*), 469  
`to_json()` (*Recommendation method*), 189  
`to_json()` (*Recommendation.RecommendationApplication method*), 185  
`to_json()` (*Recommendation.RecommendationImpact method*), 186  
`to_json()` (*Recommendation.RecommendationNewRule method*), 187  
`to_json()` (*Recommendation.RecommendationWorkflow method*), 188  
`to_json()` (*Report method*), 236  
`to_json()` (*ReportSeverity method*), 239  
`to_json()` (*ReputationOverride method*), 474  
`to_json()` (*Result method*), 141  
`to_json()` (*Result.Device method*), 139  
`to_json()` (*Result.Fields method*), 139  
`to_json()` (*Result.Metrics method*), 140  
`to_json()` (*ResultFacet method*), 142  
`to_json()` (*ResultFacet.Values method*), 142  
`to_json()` (*Run method*), 151  
`to_json()` (*RunHistory method*), 153  
`to_json()` (*SensorKit method*), 510  
`to_json()` (*Template method*), 162  
`to_json()` (*TemplateHistory method*), 164  
`to_json()` (*UnrefreshableModel method*), 571  
`to_json()` (*USBDevice method*), 194  
`to_json()` (*USBDeviceApproval method*), 197  
`to_json()` (*USBDeviceBlock method*), 202  
`to_json()` (*User method*), 482  
`to_json()` (*VCenterComputeResource method*), 533  
`to_json()` (*Vulnerability method*), 493  
`to_json()` (*Vulnerability.OrgSummary method*), 492  
`to_json()` (*Watchlist method*), 244  
`to_json()` (*WatchlistAlert method*), 341  
`to_json()` (*WatchlistAlert.Note method*), 337  
`touch()` (*AssetGroup method*), 350  
`touch()` (*BypassRuleConfig method*), 443  
`touch()` (*CorePreventionRuleConfig method*), 445  
`touch()` (*DataCollectionRuleConfig method*), 447  
`touch()` (*Feed method*), 223  
`touch()` (*FeedModel method*), 225  
`touch()` (*Grant method*), 394  
`touch()` (*Grant.Profile method*), 390  
`touch()` (*HostBasedFirewallRuleConfig method*), 454  
`touch()` (*HostBasedFirewallRuleConfig.FirewallRule method*), 450  
`touch()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup method*), 451  
`touch()` (*IOC method*), 228  
`touch()` (*IOC\_V2 method*), 231  
`touch()` (*MutableBaseModel method*), 560  
`touch()` (*Policy method*), 436  
`touch()` (*PolicyRule method*), 440  
`touch()` (*PolicyRuleConfig method*), 457  
`touch()` (*Report method*), 236  
`touch()` (*ReportSeverity method*), 239  
`touch()` (*USBDeviceApproval method*), 197  
`touch()` (*User method*), 482  
`touch()` (*Watchlist method*), 244  
`tree` (*Process property*), 467  
`try_json()` (*in module cbc\_sdk.connection*), 581

## U

`UnauthorizedError`, 588  
`unignore()` (*IOC\_V2 method*), 231  
`unignore()` (*Report method*), 236  
`uninstall_sensor()` (*Device method*), 362  
`uninstall_sensor()` (*DeviceSearchQuery method*), 372  
`UnrefreshableModel` (*class in cbc\_sdk.base*), 571  
`update()` (*Alert method*), 256  
`update()` (*AlertSearchQuery method*), 277  
`update()` (*CBAnalyticsAlert method*), 285  
`update()` (*ContainerRuntimeAlert method*), 292  
`update()` (*DeviceControlAlert method*), 298  
`update()` (*Feed method*), 223  
`update()` (*GroupedAlertSearchQuery method*), 321  
`update()` (*HostBasedFirewallAlert method*), 328  
`update()` (*IntrusionDetectionSystemAlert method*), 335  
`update()` (*Report method*), 236  
`update()` (*Watchlist method*), 244  
`update()` (*WatchlistAlert method*), 342  
`update_criteria()` (*AlertSearchQuery method*), 278  
`update_criteria()` (*AssetGroupQuery method*), 353  
`update_criteria()` (*AsyncProcessQuery method*), 462  
`update_criteria()` (*AuthEventQuery method*), 218  
`update_criteria()` (*AWSComputeResourceQuery method*), 523

- `update_criteria()` (*BaseComputeResourceQuery method*), 529
  - `update_criteria()` (*CriteriaBuilderSupportMixin method*), 551
  - `update_criteria()` (*DeviceSearchQuery method*), 373
  - `update_criteria()` (*DifferentialQuery method*), 172
  - `update_criteria()` (*EnrichedEventQuery method*), 183
  - `update_criteria()` (*EventFacetQuery method*), 381
  - `update_criteria()` (*EventQuery method*), 386
  - `update_criteria()` (*FacetQuery method*), 137, 556
  - `update_criteria()` (*GroupedAlertSearchQuery method*), 321
  - `update_criteria()` (*ObservationQuery method*), 423
  - `update_criteria()` (*Query method*), 566
  - `update_criteria()` (*RecommendationQuery method*), 192
  - `update_criteria()` (*ResultQuery method*), 148
  - `update_criteria()` (*RunHistoryQuery method*), 156
  - `update_criteria()` (*SensorKitQuery method*), 512
  - `update_criteria()` (*TemplateHistoryQuery method*), 166
  - `update_criteria()` (*USBDeviceApprovalQuery method*), 200
  - `update_criteria()` (*USBDeviceQuery method*), 207
  - `update_criteria()` (*VCenterComputeResourceQuery method*), 545
  - `update_exclusions()` (*AlertSearchQuery method*), 278
  - `update_exclusions()` (*AsyncProcessQuery method*), 462
  - `update_exclusions()` (*AuthEventQuery method*), 219
  - `update_exclusions()` (*EnrichedEventQuery method*), 183
  - `update_exclusions()` (*EventFacetQuery method*), 382
  - `update_exclusions()` (*EventQuery method*), 386
  - `update_exclusions()` (*ExclusionBuilderSupportMixin method*), 552
  - `update_exclusions()` (*FacetQuery method*), 557
  - `update_exclusions()` (*GroupedAlertSearchQuery method*), 321
  - `update_exclusions()` (*ObservationQuery method*), 424
  - `update_exclusions()` (*Query method*), 566
  - `update_policy()` (*Device method*), 362
  - `update_policy()` (*DeviceSearchQuery method*), 373
  - `update_sensor_version()` (*Device method*), 362
  - `update_sensor_version()` (*DeviceSearchQuery method*), 373
  - `update_threat()` (*Alert method*), 257
  - `update_threat()` (*CBAnalyticsAlert method*), 285
  - `update_threat()` (*ContainerRuntimeAlert method*), 292
  - `update_threat()` (*DeviceControlAlert method*), 299
  - `update_threat()` (*HostBasedFirewallAlert method*), 328
  - `update_threat()` (*IntrusionDetectionSystemAlert method*), 335
  - `update_threat()` (*WatchlistAlert method*), 342
  - `url` (*BaseAPI property*), 577
  - `url` (*CBCloudAPI property*), 130
  - `urn` (*User property*), 482
  - `USBDevice` (*class in cbc\_sdk.endpoint\_standard.usb\_device\_control*), 193
  - `USBDeviceApproval` (*class in cbc\_sdk.endpoint\_standard.usb\_device\_control*), 195
  - `USBDeviceApprovalQuery` (*class in cbc\_sdk.endpoint\_standard.usb\_device\_control*), 198
  - `USBDeviceBlock` (*class in cbc\_sdk.endpoint\_standard.usb\_device\_control*), 201
  - `USBDeviceBlockQuery` (*class in cbc\_sdk.endpoint\_standard.usb\_device\_control*), 202
  - `USBDeviceQuery` (*class in cbc\_sdk.endpoint\_standard.usb\_device\_control*), 203
  - `User` (*class in cbc\_sdk.platform.users*), 477
  - `User.UserBuilder` (*class in cbc\_sdk.platform.users*), 477
  - `user_ids()` (*UserQuery method*), 483
  - `UserQuery` (*class in cbc\_sdk.platform.users*), 482
- ## V
- `valid_rule_configs()` (*Policy method*), 436
  - `validate()` (*AssetGroup method*), 350
  - `validate()` (*BypassRuleConfig method*), 443
  - `validate()` (*CorePreventionRuleConfig method*), 445
  - `validate()` (*DataCollectionRuleConfig method*), 447
  - `validate()` (*Feed method*), 224
  - `validate()` (*FeedModel method*), 225
  - `validate()` (*Grant method*), 394
  - `validate()` (*Grant.Profile method*), 390
  - `validate()` (*HostBasedFirewallRuleConfig method*), 454
  - `validate()` (*HostBasedFirewallRuleConfig.FirewallRule method*), 450
  - `validate()` (*HostBasedFirewallRuleConfig.FirewallRuleGroup method*), 451
  - `validate()` (*IOC method*), 228
  - `validate()` (*IOC\_V2 method*), 231
  - `validate()` (*MutableBaseModel method*), 560
  - `validate()` (*Policy method*), 437
  - `validate()` (*PolicyRule method*), 441
  - `validate()` (*PolicyRuleConfig method*), 457
  - `validate()` (*Report method*), 237



- validate() (*ReportSeverity* method), 239  
 validate() (*USBDeviceApproval* method), 197  
 validate() (*User* method), 482  
 validate() (*Watchlist* method), 244  
 validate\_process\_query() (*CBCloudAPI* method), 130  
 values (*ComputeResourceFacet* property), 531  
 values\_ (*DeviceFacet* property), 365  
 values\_ (*DeviceSummaryFacet* property), 133  
 values\_ (*ResultFacet* property), 142  
 VCenterComputeResource (class in *cbc\_sdk.workload.vm\_workloads\_search*), 531  
 VCenterComputeResourceQuery (class in *cbc\_sdk.workload.vm\_workloads\_search*), 533  
 Vulnerability (class in *cbc\_sdk.platform.vulnerability\_assessment*), 490  
 Vulnerability.AssetView (class in *cbc\_sdk.platform.vulnerability\_assessment*), 491  
 Vulnerability.OrgSummary (class in *cbc\_sdk.platform.vulnerability\_assessment*), 492  
 vulnerability\_refresh() (*Device* method), 363  
 VulnerabilityAssetViewQuery (class in *cbc\_sdk.platform.vulnerability\_assessment*), 494  
 VulnerabilityOrgSummaryQuery (class in *cbc\_sdk.platform.vulnerability\_assessment*), 500  
 VulnerabilityQuery (class in *cbc\_sdk.platform.vulnerability\_assessment*), 501  
 where() (*AsyncProcessQuery* method), 463  
 where() (*AuthEventQuery* method), 219  
 where() (*AWSComputeResourceQuery* method), 524  
 where() (*BaseComputeResourceQuery* method), 529  
 where() (*DeviceSearchQuery* method), 373  
 where() (*EnrichedEventQuery* method), 184  
 where() (*EventFacetQuery* method), 382  
 where() (*EventQuery* method), 387  
 where() (*FacetQuery* method), 138, 557  
 where() (*FeedQuery* method), 226  
 where() (*GroupedAlertSearchQuery* method), 322  
 where() (*ObservationQuery* method), 424  
 where() (*Query* method), 566  
 where() (*QueryBuilder* method), 568  
 where() (*QueryBuilderSupportMixin* method), 569  
 where() (*ReportQuery* method), 238  
 where() (*ReputationOverrideQuery* method), 477  
 where() (*ResultQuery* method), 149  
 where() (*RunHistoryQuery* method), 156  
 where() (*RunQuery* method), 159  
 where() (*SimpleQuery* method), 570  
 where() (*SummaryQuery* method), 472  
 where() (*TemplateHistoryQuery* method), 167  
 where() (*USBDeviceApprovalQuery* method), 201  
 where() (*USBDeviceQuery* method), 208  
 where() (*VCenterComputeResourceQuery* method), 545  
 where() (*VulnerabilityAssetViewQuery* method), 500  
 where() (*VulnerabilityQuery* method), 507  
 where() (*WatchlistQuery* method), 245  
 Win32Error (class in *cbc\_sdk.winerror*), 616  
 with\_traceback() (*ApiError* method), 584  
 with\_traceback() (*ClientError* method), 584  
 with\_traceback() (*ConnectionError* method), 584  
 with\_traceback() (*CredentialError* method), 585  
 with\_traceback() (*FunctionalityDecommissioned* method), 585  
 with\_traceback() (*InvalidHashError* method), 585  
 with\_traceback() (*InvalidObjectError* method), 586  
 with\_traceback() (*LiveResponseError* method), 601  
 with\_traceback() (*ModelNotFound* method), 586  
 with\_traceback() (*MoreThanOneResultError* method), 586  
 with\_traceback() (*NonQueryableModel* method), 587  
 with\_traceback() (*NSXJobError* method), 586  
 with\_traceback() (*ObjectNotFoundError* method), 587  
 with\_traceback() (*OperationCancelled* method), 587  
 with\_traceback() (*QuerySyntaxError* method), 588  
 with\_traceback() (*ServerError* method), 588  
 with\_traceback() (*TimeoutError* method), 588  
 with\_traceback() (*UnauthorizedError* method), 589  
 WorkerStatus (class in *cbc\_sdk.live\_response\_api*), 611  
 workflow\_ (*Alert* property), 257  
 workflow\_ (*CBAnalyticsAlert* property), 286

## W

`workflow_` (*ContainerRuntimeAlert* property), [293](#)  
`workflow_` (*DeviceControlAlert* property), [299](#)  
`workflow_` (*HostBasedFirewallAlert* property), [329](#)  
`workflow_` (*IntrusionDetectionSystemAlert* property),  
[336](#)  
`workflow_` (*Recommendation* property), [189](#)  
`workflow_` (*WatchlistAlert* property), [343](#)  
`WorkItem` (class in `cbc_sdk.live_response_api`), [611](#)