



# Cb.Conf

CB v4.2.4.150206.1225

February 06, 2015

## Contents

Format . . . . .	2
Data storage settings . . . . .	3
DatastoreRootDir . . . . .	3
AllianceClientStorefilePurgeMax . . . . .	3
AllianceClientNoStorefileDelete . . . . .	3
EnableSolrBinaryInfoNotifications . . . . .	3
EnableSolrFeedNotifications . . . . .	3
CbSolrConnectionTimeout . . . . .	3
CbSolrSocketTimeout . . . . .	4
KeepAllModuleFiles . . . . .	4
MaxEventStoreSizeInDocs . . . . .	4
MaxEventStoreDays . . . . .	4
MaxEventStoreSizeInMB . . . . .	4
MaxEventStoreSizeInPercent . . . . .	4
MinAvailableSizeInMB . . . . .	4
ProcessDocumentSplitThreshold . . . . .	4
WatchlistEndTimeOffset . . . . .	5
WatchlistStartTimeOffset . . . . .	5
Operational Communication Settings . . . . .	5
AllowNullSensorHostRegister . . . . .	5
AllianceNoClientCert . . . . .	5
AllianceVerifyServerCert . . . . .	5
EnforceClientCerts . . . . .	5
Network Settings . . . . .	6
CoreServicesIP . . . . .	6
CoreServicesPort . . . . .	6
DatastorePort . . . . .	6
DatastoreIP . . . . .	6
NginxSensorHttpPort . . . . .	6
NginxWebApiHttpPort . . . . .	6
RedisHost . . . . .	6
RedisPort . . . . .	6
RedisStatsHost . . . . .	7
RedisStatsPort . . . . .	7
SolrIP . . . . .	7
SolrPort . . . . .	7
WebsocketPort . . . . .	7
SSL Certificate usage . . . . .	7
SSLCertFile . . . . .	7
SSLKeyFile . . . . .	8
AllianceCert . . . . .	8
AllianceCertKey . . . . .	8
Cb Internal Settings . . . . .	8
CbUser . . . . .	8
CbGroup . . . . .	8
CbFileDescriptorLimit . . . . .	8
CbLicenseFile . . . . .	8

CbServerTokenFile . . . . .	9
CbJavaHome . . . . .	9
CoreServicesSmallScaleSensorCount . . . . .	9
CoreServicesMaxCheckinInterval . . . . .	9
CoreServicesProcessSearchIntervalSeconds . . . . .	9
CoreServicesEnableFuzzyProcessFacets . . . . .	9
CoreServicesEnableFuzzyBinaryFacets . . . . .	9
CoreServicesProcessSearchOrder . . . . .	9
CoreServicesBinarySearchOrder . . . . .	9
CoreServicesProcessPageSize . . . . .	10
CoreServicesBinaryPageSize . . . . .	10
CoreServicesProcessAutocomplete . . . . .	10
CoreServicesBinaryAutocomplete . . . . .	10
TimestampDeltaThreshold . . . . .	10
CoreServicesPidFile . . . . .	10
SensorInstallerDir . . . . .	10
FlaskSecret . . . . .	10
FailedLogonLockoutCount . . . . .	10
AccountUnlockInterval . . . . .	11
UserActivityQuota . . . . .	11
UserActivityDelta . . . . .	11
AllianceClientPidFile . . . . .	11
AllianceSyncIntervalSecs . . . . .	11
AllianceURL . . . . .	11
DatastoreJvmMax . . . . .	11
DatastoreAllowUnregisteredSensor . . . . .	11
DatastoreShutdownTimeout . . . . .	12
DatastoreDisableJMXRemote . . . . .	12
DisableDatastoreCache . . . . .	12
EventStoreSolrCore . . . . .	12
ModInfoStoreSolrCore . . . . .	12
ModInfoStoreFlushInterval . . . . .	12
PgSqlDataDir . . . . .	12
PgSqlPidFile . . . . .	12
PgSqlLogfilePath . . . . .	12
PgSqlPort . . . . .	13
DatabaseURL . . . . .	13
ModstorePath . . . . .	13
RedisPidFile . . . . .	13
WatchlistSyslogTemplateProcess . . . . .	13
WatchlistSyslogTemplateBinary . . . . .	13
BinaryInfoSyslogTemplateObserved . . . . .	13
BinaryInfoSyslogTemplateGroupObserved . . . . .	14
BinaryInfoSyslogTemplateHostObserved . . . . .	14

`/etc/cb/cb.conf` is the primary spot for server configuration. There should be no need to modify this file in a typical production environment, but configuration options listed here could be helpful when troubleshooting the server or tailoring the configuration for local integration.

## Format

This file is consumed by CB services as well as bash shell. The formatting rules must be followed closely in order to avoid parsing errors.

1. Comments CAN ONLY be placed on their own line. There should not be any comments added to the end of the line that already has a property. Any line starting with `#` are considered a comment.
2. All properties must be expressed strictly as “Name=Value” pairs. Nothing else (besides comments) should be placed in this file.

3. There MUST NOT be any whitespace (spaces or tabs) around the equals = sign: `name=value <GOOD>`  
`name =value <BAD>` `name= value <BAD>`

## Data storage settings

### DatastoreRootDir

*Default:* `/var/cb`

Path to the root directory where all Carbon Black Enterprise Server will store all of its runtime data. This data includes Solr, PostgreSQL and flat file storage of module files. Each of these storage types has additional parameters which are covered in subsections below

*Note:* To change this value, you must rerun `cbinit` and reinitialize the server.

### AllianceClientStorefilePurgeMax

*Default:* 100

Specifies maximum number of store files alliance client will purge from local hard drive if it identifies those files should not be stored locally (i.e. if they were already uploaded to Alliance server)

### AllianceClientNoStorefileDelete

*Default:* 0

This option specifies whether or not alliance client should keep binaries locally after they have been uploaded to the central Alliance server. If set to 0, Alliance client will delete storefiles after upload in order to preserve local hard drive space. If set to 1, binary modules will not be deleted after upload. **WARNING:** Purge script will still erase binary files to recover disk space unless `KeepAllModuleFiles` has been set to 1

Binaries can always be downloaded from the Alliance, even if they have been deleted.

### EnableSolrBinaryInfoNotifications

*Default:* False

When set to True, this parameter enables notifications on new binaries, and new hosts and groups observing a particular binary. Notifications are sent to syslog as log messages

### EnableSolrFeedNotifications

*Default:* False

When set to True, this parameter enables notifications on commit of a document with a feed hit. Notifications are sent to UI as alerts and to syslog as log messages

### CbSolrConnectionTimeout

*Default:* 0

This parameter sets the connection timeout in milliseconds from datastore to Solr backend. A value of 0 means Solr client's internal defaults are used.

**CbSolrSocketTimeout**

*Default:* 0

This parameter sets the socket read timeout in milliseconds from datastore to Solr backend. A value of 0 means Solr client's internal defaults are used.

**KeepAllModuleFiles**

*Default:* 0

Default value of 0 indicates that module files will be erased a) after they are uploaded to the alliance server and b) when data is purged to free up room on storage volume. Changing this value to a 1 will tell the server to NEVER delete module files

**MaxEventStoreSizeInDocs**

*Default:* 120

This parameter configures the threshold that process document count in millions that would trigger clean up. This parameter would take precedence over all other storage size parameters.

**MaxEventStoreDays**

*Default:* n/a

By default, process data is purged automatically when disk space is required. If this value is set, any process with a `last_server_update` time older than this is deleted.

**MaxEventStoreSizeInMB**

*Default:* n/a

By default, process data is purged automatically when disk space is required. If this value is set, the oldest day of process data is deleted until the size of the process store is less than this value.

**MaxEventStoreSizeInPercent**

*Default:* 50

This parameter configures the threshold that disk usage would trigger clean up as percentage of total disk space available to event store. Total disk space available to event store is calculated as the sum of current event store size and free disk space

**MinAvailableSizeInMB**

*Default:*

This parameter is optional and can be used to set a hard limit on the size of available disk space that has to be maintained on the mount point where event store resides. This parameter would take precedence over all other storage size parameters except `MaxEventStoreSizeInDocs`

**ProcessDocumentSplitThreshold**

*Default:* 10000

This parameter sets the total number of events after which a process document will start splitting into multiple segments.

### WatchlistEndTimeOffset

*Default:* 0MINUTES

This parameter changes the search window end time offsets for watchlist searcher job. Available units are SECONDS, MINUTES, and HOURS. **WARNING:** these parameters are optimized based on the commit interval of Solr backend. Please contact technical support before changing the values specified here.

### WatchlistStartTimeOffset

*Default:* 12MINUTES

This parameter changes the search window start time offsets for watchlist searcher job. Available units are SECONDS, MINUTES, and HOURS. **WARNING:** these parameters are optimized based on the commit interval of Solr backend. Please contact technical support before changing the values specified here.

## Operational Communication Settings

These settings adjust the operational profile of the Carbon Black Enterprise Server communications with sensors and the Carbon Black Alliance server.

### AllowNullSensorHostRegister

*Default:* 0

The server requires the computer SID during the initial sensor registration. If this value is blank, the server will reject registration. If registration is rejected, the sensor will re-attempt registration in a few minutes, including another attempt to get the computer SID.

If the condition which causes the sensor to not get the SID is ephemeral, it will fix itself. If the condition is chronic, set this value to 0 to allow the sensor to register with an empty SID.

Sensors rejected for an empty SID will be logged in `/var/log/cb/coreservices/debug.log`.

### AllianceNoClientCert

*Default:* 0

The Alliance server uses SSL Client certificates to authenticate clients. Most SSL Inspection devices do not support client certificates and immediately end the connection upon their receipt.

Set this to 1 to *not* transmit the SSL client certificate. **NOTE:** contact your support representative for alternative authentication arrangements.

### AllianceVerifyServerCert

*Default:* 1

Indicates that Alliance Server's SSL certificate is to be validated against the Carbon Black certificate authority. If the server's SSL certificate was not signed by the Carbon Black CA, the connection will fail. If your network uses an SSL Inspection device, this must be disabled.

### EnforceClientCerts

*Default:* True

Cb sensors validate the server via SSL server certificates. The Cb server also validate sensors via a SSL client certificate. This setting specifies whether or not CB server will allow sensors who do not provide a SSL certificate to communicate with it.

This should generally be “True”, but can be disabled for troubleshooting, mismatched certificates or upgrading older sensors (pre v3.1.0) that did not support SSL client certificates.

## Network Settings

Review / update / adjust these settings to adjust the server listening IPs/ports.

### CoreServicesIP

*Default:* 0.0.0.0

The `coreservices` daemon will bind to this interface.

### CoreServicesPort

*Default:* 5000

The `coreservices` daemon will bind to this port.

### DatastorePort

*Default:* 9000

`cbfs-http` will bind to this port.

### DatastoreIP

*Default:* 127.0.0.1

`cbfs-http` will bind to this interface.

### NginxSensorHttpPort

*Default:* 443

The value of this property must match the configuration of ‘listen’ directive in `/etc/cb/nginx/conf.d/cb.conf`. This property isn’t used to control Nginx web server, as Nginx maintains its own configuration files. But it must be kept in sync so that other components (such as firewall management) know which ports are used for HTTP comms

### NginxWebApiHttpPort

*Default:* 443

See notes for `NginxSensorHttpPort` for more details regarding this property

### RedisHost

*Default:* localhost

Redis general cache host.

### RedisPort

*Default:* 6379 Redis general cache listen port (TCP).

## RedisStatsHost

*Default:* localhost

Redis statistics cache host.

## RedisStatsPort

*Default:* 6379 Redis statistics cache listen port (TCP).

## SolrIP

*Default:* 127.0.0.1

Tomcat will bind to this interface

## SolrPort

*Default:* 8080

Tomcat will bind to this port.

Identifies the HTTP port that is used for all external communications, which includes sensors as well as web UI.  
NOTE: If this value is modified, a corresponding change must also be made in `/etc/nginx/conf.d/cb.conf` file

## WebsocketPort

*Default:* 5006

The websocket daemon will bind to this port. NOTE: This configuration option is subject to change.

## SSL Certificate usage

Carbon Black makes heavy use of SSL Certificates:

- sensors validate they are talking to the correct Carbon Black server via the SSL server certificate
- the server validates it is talking to an authentic sensor via SSL client certificates
- the server validates it is talking to the correct Carbon Black Alliance server via the SSL server certificate
- the Carbon Black Alliance server validates it is talking to an authentic Cb server via SSL client certificates

These certificates are configured below.

## SSLCertFile

*Default:* `/etc/cb/certs/cb-server.crt`

SSL certificate and private key files to be used for HTTPS communications between sensors and the Carbon Black server.

**NOTE:** If these paths are modified, a corresponding change must also be made in `/etc/nginx/conf.d/cb.conf` file

**NOTE:** These certificates are generated during `cbinit` and unique to each Carbon Black server.

## SSLKeyFile

*Default:* /etc/cb/certs/cb-server.key

See SSLCertFile

## AllianceCert

*Default:* /etc/cb/certs/carbonblack-alliance-client.crt

SSL private key and certificate files which are used for client-side authentication when an HTTPS connection with Alliance Server is established. These files are placed onto the machine when Carbon Black Release RPM is installed. They are then used whenever enterprise server machine needs to communicate with central Carbon Black servers. This includes yum repositories for installing and upgrading enterprise server software as well as the alliance client service.

Please note that certificates are specific to each organization and should be treated with care (i.e. not shared with other organizations or people outside your company)

## AllianceCertKey

*Default:* /etc/cb/certs/carbonblack-alliance-client.key

See AllianceCert

## Cb Internal Settings

These settings are unlikely to be required.

### CbUser

*Default:* cb

These settings control the user and group the Carbon Black services run as. The `cb` user and group are created at RPM install. If you would like to use another user or group, create the user/group, update these values and restart `cb-enterprise`.

### CbGroup

*Default:* cb

See CbUser

### CbFileDescriptorLimit

*Default:* 80000

By default, CentOS allows only 1024 file descriptors per process. This is too low for Carbon Black. Cb will update the process file descriptor limit to this value with `ulimit -n` in the `cb-enterprise` init script.

### CbLicenseFile

*Default:* /etc/cb/server.lic

The path to the server license file. This is provided by Cb support.



## **CbServerTokenFile**

*Default:* /etc/cb/server.token

A random hex string used to uniquely identify this server installation.

## **CbJavaHome**

*Default:* /usr/lib/jvm/jre-1.7.0-openjdk.x86\_64/

Carbon Black requires JRE 1.7.0+. If the JRE is installed at a different location on your server, change this value.

## **CoreServicesSmallScaleSensorCount**

*Default:* 5

If the number of currently active sensors is less than this threshold, the sensor checkin interval is always 30 seconds. If it is greater, Carbon Black will calculate a dynamic checkin interval.

## **CoreServicesMaxCheckinInterval**

When present, configures the maximum interval, in seconds, between successive sensor checkins from a single sensor. By default, the maximum is 1335 seconds (22m 15s). Raising this value decreases load on the server, as there are fewer sensor checkins and fewer modifications to the event store.

## **CoreServicesProcessSearchIntervalSeconds**

When present, all process searches in the UI are limited to the most recent number of seconds specified in this setting. This applies to both process searches in the process search page and process watchlists from the watchlist page.

This setting applies only to the Carbon Black web UI. Direct API queries do not honor this setting.

## **CoreServicesEnableFuzzyProcessFacets**

### **CoreServicesEnableFuzzyBinaryFacets**

- Default\*: True

Use statistical sampling for calculation of facets terms. This provides significantly improved runtime performance and reduced memory usage.

## **CoreServicesProcessSearchOrder**

*Default:* "start desc"

The default sort order of process search results as seen in the UI. The format of this field is: fieldnamedirection

where `direction` is one of "asc" or "desc"

## **CoreServicesBinarySearchOrder**

*Default:* "server\_added\_timestamp desc"

The default sort order of binary search results as seen in the UI. The format of this field is: fieldnamedirection

where `direction` is one of "asc" or "desc"

**CoreServicesProcessPageSize**

*Default:* 10

The number of matching process documents to display per-page as seen in the UI.

**CoreServicesBinaryPageSize**

*Default:* 10

The number of matching binary documents to display per-page as seen in the UI.

**CoreServicesProcessAutocomplete**

*Default:* Suggester

The backend autocomplete method for process autocomplete. Valid values are: Suggester: Faster but does not include counts nor infrequent terms Terms: Slower but does include counts and all terms

**CoreServicesBinaryAutocomplete**

*Default:* Terms

The backend autocomplete method for binary autocomplete. Valid values are: Suggester: Faster but does not include counts nor infrequent terms Terms: Slower but does include counts and all terms

**TimestampDeltaThreshold**

*Default:* 5

The time, in seconds, used as a threshold for identifying sensors with unsynchronized clocks.

**CoreServicesPidFile**

*Default:* /var/run/cb/coreservices.pid

This file contains the current process id of the `coreservices` daemon.

**SensorInstallerDir**

*Default:* /usr/share/cb/coreservices/installers

This is the authoritative directory of sensor installers. The contents of this directory are loaded by `coreservices` at startup and used for the sensor versions, including the definition of 'latest' if sensors are configured to always "Latest" version.

**FlaskSecret**

*Default:* none

This value is a random string of ascii printables. It is unique per server and generated during `cbinit`. It is used to encrypt the session cookies used after authenticating to the Web UI.

**FailedLogonLockoutCount**

*Default:* 10

The number of times a user can fail authentication before the account is locked

**AccountUnlockInterval**

*Default:* 30

A locked account will unlock after this many minutes.

**UserActivityQuota**

*Default:* 10000

Carbon Black logs all user authentication in postgres. This setting defines the minimum number of authentication records kept.

**UserActivityDelta**

*Default:* .1

This setting defines when to start trimming user authentication records. It is a percentage of `UserActivityQuota`. e.g., if `UserActivityQuota` is set to 10000 and `UserActivityQuotaDelta` is set to .1 then the database will grow to 11000. Once it hits 11000 it will shrink down to 10000.

**AllianceClientPidFile**

*Default:* `/var/run/cb/allianceclient.pid`

Path to the PID file which is used for `cb-allianceclient` service control

**AllianceSyncIntervalSecs**

*Default:* 60

Number of seconds between periodic connection attempts to the alliance server.

**AllianceURL**

*Default:* `https://api.alliance.carbonblack.com`

URL of the Carbon Black Alliance Server

**DatastoreJvmMax**

*Default:* 20%

Maximum amount of RAM to be used for JVM's memory heap. This parameter can be specified either as a number of megabytes (e.g. 4096) or as a percentage of host machine's physical RAM by appending '%' on the end (e.g. 30%)

**DatastoreAllowUnregisteredSensor**

*Default:* 0

Controls whether or not data store will accept data from a sensor that has not previously been registered. There is generally no reason to enable this setting.

**DatastoreShutdownTimeout**

*Default:* 60

Specifies number of seconds to wait when the data store is being stopped for all buffers and cached data to be cleanly written to disk. After this time if the service is still running, it will be forcefully stopped

**DatastoreDisableJMXRemote**

*Default:* 0

JMX Remote allows an external java management/debugging process (only on local machine) to communicate with the datastore. If this setting is not 0, then the datastore process will be launched without JMX Remote

**DisableDatastoreCache**

*Default:* false

DisableDatastoreCache option disables ehcache in datastore and forces all process events to be pushed to Solr engine immediately

**EventStoreSolrCore**

*Default:* cbevents

Name of the Solr core to be used for process data

**ModInfoStoreSolrCore**

*Default:* cbmodules

Name of the Solr core to be used for module info storage

**ModInfoStoreFlushInterval**

*Default:* 1000

Time interval in milliseconds with which buffered module info events will be pushed to the module info solr core.

**PgSqlDataDir**

*Default:* /var/cb/pgsql

PostgreSQL data directory.

**PgSqlPidFile**

*Default:* /var/run/cb/cb-pgsql.pid

Path to the PID file which is used for cb-pgsql service control

**PgSqlLogfilePath**

*Default:* /var/log/cb/pgsql/startup.log

Path to cb-pgsql startup log file. This file captures any output that may be generated prior to logging framework being initialized

**PgSqlPort**

*Default:* 5002

Port on which cb-pgsql is to listen on

**DatabaseURL**

*Default:* postgresql+psycopg2://cb:(passwd)@localhost:5002/cb

SQLAlchemy DB URL to be used when connecting to PostgreSQL

**ModstorePath**

*Default:* /var/cb/modulestore

Flat file storage location for module file storage

**RedisPidFile**

*Default:* /var/run/cb/cb-redis.pid

Path to PID file used for cb-redis service control. This file must be writable by CbUser

**WatchlistSyslogTemplateProcess**

*Default:*

Path to the [Jinja2 Template](#) used to format Process watchlist hits before sending to syslog. Use `/usr/share/cb/cbsyslog` to modify and test; see Syslog Developers Guide for additional details.

If this option is not specified, the system default template is used. Use the cbsyslog tool to retrieve the system default template.

**WatchlistSyslogTemplateBinary**

*Default:*

Path to the [Jinja2 Template](#) used to format Binary watchlist hits before sending to syslog. Use `/usr/share/cb/cbsyslog` to modify and test; see Syslog Developers Guide for additional details.

If this option is not specified, the system default template is used. Use the cbsyslog tool to retrieve the system default template.

**BinaryInfoSyslogTemplateObserved**

*Default:*

Path to the [Jinja2 Template](#) used to format binary info events before sending to syslog. These events are fired the first time a binary, as identified by md5, is observed on any sensor attached to the Carbon Black Enterprise Server. See the Syslog Developers Guide and Carbon Black Server API (CBSAPI) documentation for additional details.

If this option is not specified, the system default template is used. Use the cbsyslog tool to retrieve the system default template.

**BinaryInfoSyslogTemplateGroupObserved***Default:*

Path to the [Jinja2 Template](#) used to format binary info new group events before sending to syslog. These events are fired the first time a binary, as identified by md5, is observed on a new sensor group. See the Syslog Developers Guide and Carbon Black Server API (CBSAPI) documentation for additional details.

If this option is not specified, the system default template is used. Use the cbsyslog tool to retrieve the system default template.

**BinaryInfoSyslogTemplateHostObserved***Default:*

Path to the [Jinja2 Template](#) used to format binary info new host events before sending to syslog. These events are fired the first time a binary, as identified by md5, is observed on a new sensor. See the Syslog Developers Guide and Carbon Black Server API (CBSAPI) documentation for additional details.

If this option is not specified, the system default template is used. Use the cbsyslog tool to retrieve the system default template.