

Automatic Reconnections



- [Advisories](#)
 - [Connection timeout attributes](#)
 - [Reconnection attributes](#)
-

All the client libraries maintained on the [nats.io GitHub page](#) will automatically attempt to re-connect if their current server connection gets disconnected for any reason. Upon re-connection the client library will automatically re-establish all the subscriptions, there is nothing for the application programmer to do.

Unless specifically [disabled](#) client will try to re-connect to one of the servers it knows about, either through the URLs provided in the `connect` call or the URLs provided by the NATS system during earlier connects. This feature allows NATS applications and the NATS system itself to self-heal and reconfigure itself with no additional configuration or intervention.

You can adjust the [wait time](#) between connections attempts, the [maximum](#) number of reconnection attempts, and adjust the size of the reconnection [buffer](#).

Advisories

Your application can register callback to receive [events](#) to be notified about the following connection events:

- `ClosedCB ConnHandler`

The ClosedCB handler is called when a client will no longer be connected.

- `DisconnectedCB ConnHandler`

The DisconnectedCB handler is called whenever the connection is disconnected. It will not be called if DisconnectedErrCB is set **DEPRECATED**: Use DisconnectedErrCB instead which passes error that caused the disconnect event.

- `DisconnectedErrCB ConnErrorHandler`

The `DisconnectedErrCB` handler is called whenever the connection is disconnected. Disconnected error could be nil, for instance when user explicitly closes the connection. **NOTE:** `DisconnectedCB` will not be called if `DisconnectedErrCB` is set

- `ReconnectedCB ConnHandler`

The `ReconnectedCB` handler is called whenever the connection is successfully reconnected.

- `DiscoveredServersCB ConnHandler`

The `DiscoveredServersCB` handler is called whenever a new server has joined the cluster.

- `AsyncErrorCB ErrHandler`

The `AsyncErrorCB` handler is called whenever asynchronous connection errors happen (e.g. slow consumer errors)

Connection timeout attributes

- `Timeout time.Duration`

`Timeout` sets the timeout for a Dial operation on a connection. Default is `2 * time.Second`

- `PingInterval time.Duration`

`PingInterval` is the period at which the client will be sending ping commands to the server, disabled if 0 or negative. Default is `2 * time.Minute`

- `MaxPingsOut int`

`MaxPingsOut` is the maximum number of pending ping commands that can be awaiting a response before raising an `ErrStaleConnection` error. Default is `2`

Reconnection attributes

Besides the error and advisory callbacks mentioned above you can also set a few reconnection attributes in the connection options:

- `AllowReconnect bool`

`AllowReconnect` enables reconnection logic to be used when we encounter a disconnect from the current server. Default is `true`

- `MaxReconnect int`

`MaxReconnect` sets the number of reconnect attempts that will be tried before giving up. If negative, then it will never give up trying to reconnect. Default is `60`

- `ReconnectWait time.Duration`

`ReconnectWait` sets the time to backoff after attempting to (and failing to) reconnect. Default is `2 * time.Second`

- `CustomReconnectDelayCB ReconnectDelayHandler`

`CustomReconnectDelayCB` is invoked after the library tried every URL in the server list and failed to reconnect. It passes to the user the current number of attempts. This function returns the amount of time the library will sleep before attempting to reconnect again. It is strongly recommended that this value contains some jitter to prevent all connections to attempt reconnecting at the same time.

- `ReconnectJitter time.Duration`

`ReconnectJitter` sets the upper bound for a random delay added to `ReconnectWait` during a reconnect when no TLS is used. Note that any jitter is capped with `ReconnectJitterMax`. Default is `100 * time.Millisecond`

- `ReconnectJitterTLS time.Duration`

`ReconnectJitterTLS` sets the upper bound for a random delay added to `ReconnectWait` during a reconnect when TLS is used. Note that any jitter is capped with `ReconnectJitterMax`. Default is `1 * time.Second`

- `ReconnectBufSize int`

`ReconnectBufSize` is the size of the backing bufio during reconnect. Once this has been exhausted publish operations will return an error. Default is `8 * 1024 * 1024`

- `RetryOnFailedConnect bool`

`RetryOnFailedConnect` sets the connection in reconnecting state right away if it can't connect to a server in the initial set. The `MaxReconnect` and `ReconnectWait` options are used for this process, similarly to when an established connection is disconnected. If a `ReconnectHandler` is set, it will be invoked on the first successful reconnect attempt (if the initial connect fails), and if a `ClosedHandler` is set, it will be invoked if it fails to connect (after exhausting the `MaxReconnect` attempts). Default is `false`

📄 #3

📄 cocobook-sync 📄 2026-10-28 11:12:25 UTC

📄 cocobook-sync 📄 2026-10-29 01:42:36 UTC