client-unblock

- Examples

This command can unblock, from a different connection, a client blocked in a blocking operation, such as for instance `BRPOP` or `XREAD` or `WAIT`.

By default the client is unblocked as if the timeout of the command was reached, however if an additional (and optional) argument is passed, it is possible to specify the unblocking behavior, that can be **TIMEOUT** (the default) or **ERROR**. If **ERROR** is specified, the behavior is to unblock the client returning as error the fact that the client was force-unblocked. Specifically the client will receive the following error:

```
-UNBLOCKED client unblocked via CLIENT UNBLOCK
```

Note: of course as usually it is not guaranteed that the error text remains the same, however the error code will remain `-UNBLOCKED`.

This command is useful especially when we are monitoring many keys with a limited number of connections. For instance we may want to monitor multiple streams with `XREAD` without using more than N connections. However at some point the consumer process is informed that there is one more stream key to monitor. In order to avoid using more connections, the best behavior would be to stop the blocking command from one of the connections in the pool, add the new key, and issue the blocking command again.

To obtain this behavior the following pattern is used. The process uses an additional *control connection* in order to send the `CLIENT UNBLOCK` command if needed. In the meantime, before running the blocking operation on the other connections, the process runs `CLIENT ID` in order to get the ID associated with that connection. When a new key should be added, or when a key should no longer be monitored, the relevant connection blocking command is aborted by sending `CLIENT UNBLOCK` in the control connection. The blocking command will return and can be finally reissued.

This example shows the application in the context of streams, however the pattern is a general one and can be applied to other cases.

Examples

```
Connection A (blocking connection):
> CLIENT ID
2934
> BRPOP key1 key2 key3 0
(client is blocked)

... Now we want to add a new key ...

Connection B (control connection):
> CLIENT UNBLOCK 2934
1

Connection A (blocking connection):
... BRPOP reply with timeout ...
NULL
> BRPOP key1 key2 key3 key4 0
(client is blocked again)
```

□□□ #1
□ □□□ □□□ 2026-10-31 15:11:10 UTC
□ □□□ □□□ 2026-10-31 15:11:10 UTC