



# **Examples of use for the ECCP protocol (ELASTIX CALL CENTER PROTOCOL)**

**Elaborated by:** Elastix development department  
**Version:** 1.0 alpha

## VERSIONING

Version	Elastix version	Date	Edit by	Approved by	Details
1	2.0.4-beta2	13-Apr-2011	Eduardo Cueva	Bruno Macias	Initial Documentation

## Example of use for ECCP

The Elastix Call Center Protocol was developed to allow “client” applications to communicate with the CallCenter server through a TCP port to be able to make requests and to hear asynchronous events through that port.

The present document describes examples of implementation for the ECCP protocol using PHP language. This work is for illustration purposes only; it is not an example of an optimized application for production.

The examples used can be found at the following route:

<http://elx.ec/eccpexample>

The compress file contains the following PHP programs:

- *agentlogin.php*
- *hangup.php*
- *hold.php*
- *unhold.php*
- *libs/ECCP.class.php*

To expand thisy file just type: *“tar -xvzf eccp-test-scripts.tar.bz2”*

File ECCP.class.php is the main library which contains methods to establish connections with the server while other files are examples that uses that library.

## TEST PREPARATIONS

Before starting please verify that the Elastix call center version is equal or higher to 2.0.0-14 (verify `rpm -q elastix-callcenter`). This is necessary because older versions does not Support ECCP protocol.

If you don't have that version installed, you can update to the latest version executing *“yum -y update elastix-callcenter”*.

Once previous steps are performed, it's necessary to add a registry to callcenter's database, which will be needed to perform these tests. Registry will be added to “eccp\_authorized\_clients” table and it's needed because it has the credentials to perform the connection.

Commands to execute are as follows; assuming that username and password are “agentconsole” (can be changed if desired):

```
[user@example ~]# mysql -uroot -pCLAVEMYSQL call_center
mysql> insert into eccp_authorized_clients(username,md5_password)
values("agentconsole",md5("agentconsole"));
mysql> exit
```

After doing this, we can proceed with the tests:

## STARTING TESTS

To have a better idea of how this protocol's xml messages are passed, it's recommended to review callcenter's server documentation located at: */opt/elastix/dialer/Protocolo ECCP.txt*.

For our tests we can use the scripts included in the package, either on the same server or on another at the same or at an external network.

Before starting, we need to create an agent which we will use for protocol testing. An example agent will use extension "412" with password "1234", you can use Elastix's interface to create it.

### agentlogin.php

Content analysis:

```
1  #!/usr/bin/php
2  <?php
3  require_once ("libs/ECCP.class.php");
4  $x = new ECCP();
5  $host = "IP_SERVER_ELASTIX";
6  $user = "agentconsole";
7  $pass = "agentconsole";
8  $agent = "Agent/412";
9  $passAgent = "1234";
10 $logAgent = "412";
11 try {
12     print "Connect...\n";
13     $x->connect($host, $user, $pass);
14     $x->setAgentNumber($agent);
15     $x->setAgentPass($passAgent);
16     print_r($x->getAgentStatus());
17     print "Login agent\n";
18     $r = $x->loginagent($logAgent);
19     print_r($r);
20     $bFalloLogin = FALSE;
21     if (!isset($r->failure) && !isset($r->loginagent_response->failure)) while (!$bFalloLogin) {
22         $x->wait_response(1);
23         while ($e = $x->getEvent()) {
24             print_r($e);
25             foreach ($e->children() as $ee) $evt = $ee;
26             if ($evt->getName() == 'agentfailedlogin') {
27                 $bFalloLogin = TRUE;
28                 break;
29             }
30         }
31     }
32     print "Disconnect...\n";
33     $x->disconnect();
34 } catch (Exception $e) {
35     print_r($e);
36     print_r($x->getParseError());
37 }
38 ?>
```

Line 3: library "ECCP.class.php" is required, which is the class that contains functions that allow callcenter's communication.

Lines 5 to 10: have basic data like:

- \$host => Host to connect
- \$user y \$pass => Connection's user and password, which have been inserted on the database.
- \$agent => Agent number. Agent/\_\_\_\_ form
- \$passAgent => Agent password
- \$logAgent => number of extension that will be used by the agent to connect.

Lines 13 to 18: basic agent connection data.

Lines 21 to 31: successful connection data.

Line 33: Disconnection action.

Lines 34 to 37: Error handling

Testing from a console:

### **Successful Case:** Successful Login

```
[user@example ~]$ php agentlogin.php
Connect...
SimpleXMLElement Object
(
    [status] => offline
)
Login agent
SimpleXMLElement Object
(
    [status] => logging
)
SimpleXMLElement Object
(
    [agentloggedin] => SimpleXMLElement Object
        (
            [agent] => Agent/412
            [queues] => SimpleXMLElement Object
                (
                    [queue] => 9000
                )
        )
)
```

## Unsuccessful Case: Wrong Connection Credentials

```
[user@example ~]$ php agentlogin.php
Connect...
ECCPBadRequestException Object
(
    [message:protected] => Unauthorized
    [string:Exception:private] =>
    [code:protected] => 401
    [file:protected] => /path_to_folder/eccp-test-scripts/libs/ECCP.class.php
    [line:protected] => 118
    [trace:Exception:private] => Array
        (
            [0] => Array
                (
                    [file] => /path_to_folder/eccp-test-scripts/libs/ECCP.class.php
                    [line] => 287
                    [function] => send_request
                    [class] => ECCP
                    [type] => ->
                    [args] => Array
                        (
                            [0] => SimpleXMLElement Object
                                (
                                    [@attributes] => Array
                                        (
                                            [id] => 2
                                        )
                                    [getagentstatus] => SimpleXMLElement Object
                                        (
                                            [agent_number] => Agent/412
                                        )
                                )
                        )
                )
            [1] => Array
                (
                    [file] => /path_to_folder/eccp-test-scripts/agentlogin.php
                    [line] => 16
                    [function] => getagentstatus
                    [class] => ECCP
                    [type] => ->
                    [args] => Array
                        (
                        )
                )
        )
    [previous:Exception:private] =>
)
```

### **Unsuccessful Case:** Dialer service is turn off

```
[user@example ~]$ php agentlogin.php
Connect...
ECCPConnFailedException Object
(
    [message:protected] => tcp://IP_ELASTIX_SERVER:20005: (111) Connection refused
    [string:Exception:private] =>
    [code:protected] => 111
    [file:protected] => /path_to_folder/eccp-test-scripts/libs/ECCP.class.php
    [line:protected] => 86
    [trace:Exception:private] => Array
        (
            [0] => Array
                (
                    [file] => /path_to_folder/eccp-test-scripts/agentlogin.php
                    [line] => 13
                    [function] => connect
                    [class] => ECCP
                    [type] => ->
                    [args] => Array
                        (
                            [0] => IP_ELASTIX_SERVER
                            [1] => agentconsole
                            [2] => agentconsole
                        )
                )
        )
    [previous:Exception:private] =>
)
```

### **Unsuccessful Case:** extension does not exist

```
[user@example ~]$ php agentlogin.php
Connect...
SimpleXMLElement Object
(
    [status] => offline
)
Login agent
SimpleXMLElement Object
(
    [status] => logged-out
    [failure] => SimpleXMLElement Object
        (
            [code] => 404
            [message] => Specified extension not found
        )
)
Disconnect...
```

### **Unsuccessful Case: Agent doesn't exists**

```
[user@example ~]$ php agentlogin.php
Connect...
SimpleXMLElement Object
(
    [status] => offline
    [failure] => SimpleXMLElement Object
        (
            [code] => 404
            [message] => Invalid agent number
        )
)
Login agent
SimpleXMLElement Object
(
    [status] => logged-out
    [failure] => SimpleXMLElement Object
        (
            [code] => 404
            [message] => Specified agent not found
        )
)
Disconnect...
```

### **Unsuccessful Case: Agent already is logged on**

```
[user@example ~]$ php agentlogin.php
Connect...
SimpleXMLElement Object
(
    [status] => online
)
Login agent
SimpleXMLElement Object
(
    [status] => logged-out
    [failure] => SimpleXMLElement Object
        (
            [code] => 409
            [message] => Specified agent already connected to extension: 412
        )
)
Disconnect...
```

## hangup.php

Content analysis:

```
1  #!/usr/bin/php
2  <?php
3  require_once ("libs/ECCP.class.php");
4  $x = new ECCP();
5  $host = "IP_SERVER_ELASTIX";
6  $user = "agentconsole";
7  $pass = "agentconsole";
8  $agent = "Agent/412";
9  $passAgent = "1234";
10 try {
11     print "Connect...\n";
12     $x->connect($host, $user, $pass);
13     $x->setAgentNumber($agent);
14     $x->setAgentPass($passAgent);
15     print_r($x->getAgentStatus());
16     print "Hangup...\n";
17     $r = $x->hangup();
18     print_r($r);
19     print "Disconnect...\n";
20     $x->disconnect();
21 } catch (Exception $e) {
22     print_r($e);
23     print_r($x->getParseError());
24 }
25 ?>
26
```

Line 3: library "ECCP.class.php" is required, which is the class that contains functions that allow callcenter's communication.

Lines 5 to 10 have basic data like::

- \$host => host to connect
- \$user y \$pass => user and password, which have been inserted on the database at the beginning.
- \$agent => Agent number. Agent/\_\_\_\_ Form
- \$passAgent => Agent password
- \$logAgent => number of extension that will be used by the agent to connect.

Lines 12 to 14: basic agent connection data.

Line 15: Agent status is polled

Line 17: hangup action, which releases agent

Line 20: Disconnection action.

Lines 34 to 37: Error handling.

Testing from a console:

### Successful Case:

```
[user@example ~]$ php hangup.php
Connect...
SimpleXMLElement Object
(
    [status] => oncall
)
Hangup...
SimpleXMLElement Object
(
    [success] => SimpleXMLElement Object
        (
        )
    )
)
Disconnect...
```

**Unsuccessful Case:** Wrong connection credentials as described above.

**Unsuccessful Case:** Extension doesn't exist as described above

**Unsuccessful Case:** Agent doesn't exist as described above

### hold.php

Content analysis:

```
1  #!/usr/bin/php
2  <?php
3  require_once ("libs/ECCP.class.php");
4  $x = new ECCP();
5  $host = "IP_SERVER_ELASTIX";
6  $user = "agentconsole";
7  $pass = "agentconsole";
8  $agent = "Agent/412";
9  $passAgent = "1234";
10 try {
11     print "Connect...\n";
12     $x->connect($host, $user, $pass);
13     $x->setAgentNumber($agent);
14     $x->setAgentPass($passAgent);
15     print_r($x->getAgentStatus());
16     print "Hold...\n";
17     $r = $x->hold();
18     print_r($r);
19     print "Disconnect...\n";
20     $x->disconnect();
21 } catch (Exception $e) {
22     print_r($e);
23     print_r($x->getParseError());
24 }
25 ?>
26
```

At the 3rd line, library “ECCP.class.php” is required, which is the class that contains functions that allow callcenter’s communication.

Lines 5 to 10 have basic data like:

- \$host => Host to connect
- \$user y \$pass => user and password, which have been inserted on the database at the beginning
- \$agent => Agent number. Agent/\_\_\_\_ Form
- \$passAgent => Agent password
- \$logAgent => number of extension that will be used by the agent to connect.

Lines 12 to 14: basic agent connection data.

Line 15: Agent status is polled

Línea 17: hold action that puts actual call on hold

Line 20: Disconnection action.

Testing from a console:

**Successful Case:** hold event

```
[user@example ~]$ php hold.php
Connect...
SimpleXMLElement Object
(
    [status] => oncall
)
Hold...
SimpleXMLElement Object
(
    [success] => SimpleXMLElement Object
        (
        )
    )
)
Disconnect...
```

**Unsuccessful Case:** Agent is in a call

```
[user@example ~]$ php hold.php
Connect...
SimpleXMLElement Object
(
    [status] => oncall
)
Hold...
SimpleXMLElement Object
(
    [failure] => SimpleXMLElement Object
        (
        )
    )
)
```

```
[code] => 404
[message] => Agent not found or not logged in through ECCP
)
Disconnect...
```

**Unsuccessful Case:** Call Parking is disabled

([https://IP\\_ELASTIX\\_SERVER/?menu=pbxconfig&display=parking](https://IP_ELASTIX_SERVER/?menu=pbxconfig&display=parking)) and enable call parking

```
[user@example ~]$ php hold.php
Connect...
SimpleXMLElement Object
(
    [status] => online
)
Hold...
SimpleXMLElement Object
(
    [failure] => SimpleXMLElement Object
        (
            [code] => 500
            [message] => Parked call extension is disabled
        )
)
Disconnect...
```

**Unsuccessful Case:** There's no active call or hold is tried more than once

```
[user@example ~]$ php hold.php
Connect...
SimpleXMLElement Object
(
    [status] => online
)
Hold...
SimpleXMLElement Object
(
    [failure] => SimpleXMLElement Object
        (
            [code] => 417
            [message] => Agent currenty not handling a call
        )
)
Disconnect...
```

**Unsuccessful Case:** Wrong connection credentials as described above

**Unsuccessful Case:** Extension doesn't exist as described above

**Unsuccessful Case:** Agent doesn't exist as described above

## unhold.php

### Content Analysis:

```
1  #!/usr/bin/php
2  <?php
3  require_once ("libs/ECCP.class.php");
4  $x      = new ECCP();
5  $host    = "IP_SERVER_ELASTIX";
6  $user    = "agentconsole";
7  $pass    = "agentconsole";
8  $agent   = "Agent/412";
9  $passAgent = "1234";
10 try {
11     print "Connect...\n";
12     $x->connect($host, $user, $pass);
13     $x->setAgentNumber($agent);
14     $x->setAgentPass($passAgent);
15     print_r($x->getAgentStatus());
16     print "unhold...\n";
17     $r = $x->unhold();
18     print_r($r);
19     print "Disconnect...\n";
20     $x->disconnect();
21 } catch (Exception $e) {
22     print_r($e);
23     print_r($x->getParseError());
24 }
25 ?>
26
```

At the 3rd line, library “ECCP.class.php” is required, which is the class that contains functions that allow callcenter’s communication.

Lines 5 to 10 have basic data like:

- \$host => Host to connect
- \$user y \$pass => user and password, which have been inserted on the database at the beginning
- \$agent => Agent number. Agent/\_\_\_\_ Form
- \$passAgent => Agent password
- \$logAgent => number of extension that will be used by the agent to connect..

Lines 12 to 14: basic agent connection data.

Line 15: Agent status is polled

Line 17: Unhold action which recovers actual on hold call

Line 20: Disconnection action.

Testing from a console:

### **Successful Case:** event unhold

```
[user@example ~]$ php unhold.php
Connect...
SimpleXMLElement Object
(
    [status] => paused
)
unhold...
SimpleXMLElement Object
(
    [success] => SimpleXMLElement Object
        (
        )
    )
)
Disconnect...
```

### **Unsuccessful Case:** If an unhold action is performed for second time

```
[user@example ~]$ php unhold.php
Connect...
SimpleXMLElement Object
(
    [status] => oncall
)
unhold...
SimpleXMLElement Object
(
    [failure] => SimpleXMLElement Object
        (
            [code] => 417
            [message] => Agent currenty handling a call
        )
    )
)
Disconnect...
```

### **Unsuccessful Case:** Wrong connection credentials as described above

### **Unsuccessful Case:** Extension doesn't exist as described above

### **Unsuccessful Case:** Agent doesn't exist as described above