



# **Ejemplos de uso del protocolo ECCP (ELASTIX CALL CENTER PROTOCOL)**

**Elaborado por:** Departamento de Desarrollo Elastix  
**Versión:** 1.0 alpha

## VERSIONAMIENTO

Versión	Versión de Elastix	Fecha	Editado por	Aprobado Por	Detalles
1	2.0.4-beta2	13-Abr-2011	Ing. Eduardo Cueva	Ing. Bruno Macias	Documentación Inicial

## Uso del protocolo de Callcenter ECCP

El protocolo de Callcenter fue desarrollado para que aplicaciones “cliente” puedan comunicarse con el servidor de CallCenter a través de un puerto TCP para de este modo realizar peticiones, así como escuchar eventos asincrónicos a través de dicho puerto.

El presente documento describe ejemplos de implementación del protocolo ECCP usando lenguaje PHP. El propósito de los ejemplos es meramente ilustrativo y no han sido pensado como aplicaciones optimizadas para producción.

Los ejemplos usados se encuentran disponibles a través de Internet en la ruta:

<http://elx.ec/eccpexample>

El archivo comprimido contiene los siguientes programas en PHP:

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

Para descomprimirlo solo ejecute: *“tar -xvzf eccp-test-scripts.tar.bz2”*

El archivo ECCP.class.php es la librería base que contiene los métodos que permitirán establecer conexión con el servidor mientras que los demás archivos son ejemplos que usan dicha librería.

## PREPARACIÓN PARA LAS PRUEBAS

Antes de comenzar se debe verificar que la versión del módulo de call center de Elastix sea superior o igual a 2.0.0-14 (verificar rpm -q elastix-callcenter). Esto es necesario debido a que versiones menores no implementan el protocolo ECCP.

Si no se tiene dicha versión lo que se puede hacer es actualizar a la última ejecutando “yum -y update elastix-callcenter”.

Una vez realizado lo anterior es necesario agregar un registro a la base de datos de callcenter que será necesario que exista para realizar dichas pruebas. Dicho registro se hará en la tabla “eccp\_authorized\_clients” y es necesario porque contendrá las credenciales con la cual se autorizará la conexión.

Los comandos a ejecutar son los siguientes asumiendo que el usuario y clave de acceso son “agentconsole” aunque puede cambiar si se desea:

```
[user@example ~]# mysql -uroot -pCLAVEMYSQL call_center
```

```
mysql> insert into eccp_authorized_clients(username,md5_password)
values("agentconsole",md5("agentconsole"));
mysql> exit
```

Luego de haber realizado lo anterior se puede proseguir con las pruebas:

## COMENZANDO LAS PRUEBAS

Para tener una idea más clara de cómo deben ser estructurados los mensajes XML del protocolo ECCP se recomienda revisar la especificación del mismo, la cual se encuentra en su servidor, en la siguiente ruta: **/opt/elastix/dialer/Protocolo ECCP.txt**

Para las pruebas podemos usar los scripts mencionados anteriormente ya sea en el mismo servidor u otro que se encuentre en la misma red del servidor principal o externa.

Antes de comenzar necesitamos crear el agente que utilizaremos para las pruebas del protocolo. El agente de ejemplo usará la extensión "412" con la clave "1234". Debemos crear previamente dicha extensión desde la interfaz de Elastix.

### agentlogin.php

Analizamos su contenido el cual es el siguiente:

```
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 ?>
```

En la línea 3 se requiere la librería “ECCP.class.php” la cual es la clase con las funciones que permitirán comunicarse con el callcenter.

De la línea 5 a la 10 se tiene los datos básicos como:

- \$host => nombre de *host* a conectarse
- \$user y \$pass => usuario y clave de conexión, los cuales fueron insertados en la base de datos al inicio
- \$agent => número del agente. Formato Agent/\_\_\_\_
- \$passAgent => Contraseña del agente
- \$logAgent => número de extensión donde se va a conectar el agente.

De la línea 13 a 18: lo mínimo necesario para conectar al agente.

De la línea 21 a 31: en el caso de que sea exitosa la conexión.

Línea 33: acción de desconexión.

De la línea 34 a la 37: se maneja el caso de que hubiera un error.

Probando desde una consola:

### **Caso exitoso:** Login Exitoso

```
[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
                )
        )
)
```

## Caso no exitoso: credenciales de conexión incorrectas

```
[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] =>
)
```

### **Caso no exitoso:** servicio dialer está apagado

```
[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] =>
)
```

### **Caso no exitoso:** extensión no existe

```
[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...
```

### **Caso no exitoso: agente no existe**

```
[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...
```

### **Caso no exitoso: agente ya esta logueado**

```
[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

Analizamos su contenido el cual es el siguiente:

```
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
```

En la línea 3 se requiere la librería "ECCP.class.php" la cual es la clase con las funciones que permitirán comunicarse con el callcenter.

De la línea 5 a la 10 se tiene los datos básicos como:

- \$host => nombre de host a conectarse
- \$user y \$pass => usuario y clave de conexión, los cuales fueron insertados en la base de datos al inicio
- \$agent => número del agente. Formato Agent/\_\_\_\_
- \$passAgent => Contraseña del agente
- \$logAgent => número de extensión donde se va a conectar el agente.

De la línea 12 a 14: lo mínimo necesario para conectar al agente.

Línea 15: se pregunta el status del agente.

Línea 17: acción de hangup que libera a un agente.

Línea 20: es la acción de desconexión.

De la línea 34 a la 37: se maneja el caso de que hubiera un error.

Probando desde una consola:

### Caso exitoso:

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

**Caso no exitoso:** credenciales de conexión incorrectas mencionados anteriormente

**Caso no exitoso:** extensión no existe mencionados anteriormente

**Caso no exitoso:** agente no existe mencionados anteriormente

### hold.php

Analizamos su contenido el cual es el siguiente:

```
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
```

En la línea 3 se requiere la librería “ECCP.class.php” la cual es la clase con las funciones que permitirán comunicarse con el callcenter.

De la línea 5 a la 10 se tiene los datos básicos como:

- \$host => nombre de host a conectarse
- \$user y \$pass => usuario y clave de conexión, los cuales fueron insertados en la base de datos al inicio
- \$agent => número del agente. Formato Agent/\_\_\_\_
- \$passAgent => Contraseña del agente
- \$logAgent => número de extensión donde se va a conectar el agente.

De la línea 12 a 14: lo mínimo necesario para conectar al agente.

Línea 15: se pregunta el status del agente.

Línea 17: acción de hold que coloca la llamada actual en espera.

Línea 20: acción de desconexión.

Probando desde una consola:

#### **Caso exitoso:** evento hold

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

#### **Caso no exitoso:** agente en llamada

```
[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...

```

### **Caso no exitoso:** Parking esta deshabilitado

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

```

[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...

```

### **Caso no exitoso:** No hay ninguna llamada activa o se trata de ejecutar un hold más de una vez

```

[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...

```

**Caso no exitoso:** credenciales de conexión incorrectas mencionados anteriormente

**Caso no exitoso:** extensión no existe mencionados anteriormente

**Caso no exitoso:** agente no existe mencionados anteriormente

## unhold.php

Analizamos su contenido el cual es el siguiente:

```
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
```

En la línea 3 se requiere la librería "ECCP.class.php" la cual es la clase con las funciones que permitirán comunicarse con el callcenter.

De la línea 5 a la 10 se tiene los datos básicos como:

- \$host => nombre de host a conectarse
- \$user y \$pass => usuario y clave de conexión, los cuales fueron insertados en la base de datos al inicio
- \$agent => número del agente. Formato Agent/\_\_\_\_
- \$passAgent => Contraseña del agente
- \$logAgent => número de extensión donde se va a conectar el agente.

De la línea 12 a 14: lo mínimo necesario para conectar al agente.

Línea 15: se pregunta el status del agente.

Línea 17: acción de unhold que retoma la llamada actual en espera.

Línea 20: acción de desconexión.

Probando desde una consola:

### **Caso exitoso:** evento unhold

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

### **Caso no exitoso:** Si se ejecuta la acción unhold por segunda vez

```
[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...
```

**Caso no exitoso:** credenciales de conexión incorrectas mencionados anteriormente

**Caso no exitoso:** extensión no existe mencionados anteriormente

**Caso no exitoso:** agente no existe mencionados anteriormente