Hanshow ESL-Working API

Wellcom to Hanshow ESL system, the ESL-Working API is based on XML-RPC(for more details: http://en.wikipedia.org/wiki/XML-RPC), and it is easy to integrate and gives you full control of Hanshow ESL system.

1. Downlink command

Downlink command is the command from yours to ESL-Working(EW for short). The default post port is 9000, and '/RPC2' for default path.

1.1 HELLO

SYNOPSIS:

```
("HELLO",[])
```

DESCRIPTION:

Hello command

RETURN VALUE:

OK

EXAMPLE(PYTHON):

Send:

```
server.send_cmd("HELLO", [])
```

Return:

```
OK
```

NETWORK PACKAGE:

```
POST /RPC2 HTTP/1.1
Host: 127.0.0.1:9000
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 218
<?xml version='1.0'?>
<methodCall>
<methodName>send_cmd</methodName>
<params>
<param>
<value><string>HELLO</string></value>
</param>
<param>
<value><array><data>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.6
Date: Thu, 19 Mar 2015 02:12:01 GMT
Content-type: text/xml
Content-length: 128
<?xml version='1.0'?>
<methodResponse>
<params>
<param>
<value><string>OK</string></value>
</param>
</params>
</methodResponse>
```

1.2 ESL_ADD

SYNOPSIS:

```
('ESL_ADD',
    [
        {'nw1':'53-64-39-66','eslid':'5D-85-A8-99','nw2':'52-56-78-53',
        'nw3':'240','nw4':'DOT20','op1':'1','op2':'1','op3':'HS_EL_5103','
op4':'promo_temp'},
        {…},
    ]
)
```

DESCRIPTION:

Add information of price tag(s) if ESL ID does NOT exist, or update information of price tag(s) if ESL ID exists.

RETURN VALUE:

```
['OK',
    [
        {'nw4':'DOT20','op4':'promo_temp','eslid':'5D-85-A8-99','nw1':'53-
64-39-66',
        'nw2':'52-56-78-53','nw3':'240','ack':'00','op2':'1','op3':'HS_EL_
5103','bak':'00','op1':'1'},
        {…},
    ]
]
```

Upon successful completion this command return 'bak':'00'. Otherwise, an error code is returned.

SEE ALSO:

Appendix for argument description.

EXAMPLE(PYTHON):

Send:

```
server.send_cmd('ESL_ADD',
    [
        {'nw1':'53-64-39-66','eslid':'5D-85-A8-99','nw2':'52-56-78-53','nw
3':'240',
        'nw4':'DOT20','op1':'1','op2':'1','op3':'HS_EL_5103','op4':'promo_
temp'}
    ]
)
```

Return:

```
['OK',
    [
        {'nw4':'DOT20','op4':'promo_temp','eslid':'5D-85-A8-99','nw1':'53-
64-39-66',
        'nw2':'52-56-78-53','nw3':'240','ack':'00','op2':'1','op3':'HS_EL_
5103','bak':'00','op1':'1'}
    ]
]
```

NETWORK PACKAGE:

```
POST /RPC2 HTTP/1.1
Host: 127.0.0.1:9000
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 941
<?xml version='1.0'?>
<methodCall>
<methodName>send_cmd</methodName>
<params>
<param>
<value><string>ESL_ADD</string></value>
</param>
<param>
<value><array><data>
<value><struct>
<member>
<name>nw4</name>
<value><string>DOT20</string></value>
</member>
<member>
<name>op4</name>
<value><string>promo_temp</string></value>
</member>
<member>
<name>eslid</name>
<value><string>5D-85-A8-99</string></value>
</member>
<member>
<name>nw1</name>
<value><string>53-64-39-66</string></value>
</member>
```

```xml
<member>
<name>nw2</name>
<value><string>52-56-78-53</string></value>
</member>
<member>
<name>nw3</name>
<value><string>240</string></value>
</member>
<member>
<name>op2</name>
<value><string>1</string></value>
</member>
<member>
<name>op3</name>
<value><string>HS_EL_5103</string></value>
</member>
<member>
<name>op1</name>
<value><string>1</string></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.6
Date: Fri, 20 Mar 2015 02:30:55 GMT
Content-type: text/xml
Content-length: 1080
<?xml version='1.0'?>
<methodResponse>
<params>
<param>
<value><array><data>
<value><string>OK</string></value>
<value><array><data>
<value><struct>
<member>
<name>nw4</name>
<value><string>DOT20</string></value>
</member>
<member>
<name>op4</name>
```

```
<value><string>promo_temp</string></value>
</member>
<member>
<name>eslid</name>
<value><string>5D-85-A8-99</string></value>
</member>
<member>
<name>nw1</name>
<value><string>53-64-39-66</string></value>
</member>
<member>
<name>nw2</name>
<value><string>52-56-78-53</string></value>
</member>
<member>
<name>nw3</name>
<value><string>240</string></value>
</member>
<member>
<name>ack</name>
<value><string>00</string></value>
</member>
<member>
<name>op2</name>
<value><string>1</string></value>
</member>
<member>
<name>op3</name>
<value><string>HS_EL_5103</string></value>
</member>
<member>
<name>bak</name>
<value><string>00</string></value>
</member>
<member>
<name>op1</name>
<value><string>1</string></value>
</member>
</struct></value>
</data></array></value>
</data></array></value>
</param>
</params>
</methodResponse>
```

1.3 ESL_DEL

SYNOPSIS:

```
('ESL_DEL',[{'eslid':'5D-85-A8-99'}, {…}])
```

DESCRIPTION:

Delete price tag(s)

RETURN VALUE:

```
['OK', [{'eslid': '5D-85-A8-99', 'bak': '00'}, {…}]]
```

Upon successful completion this command return 'bak':'00'. Otherwise, an error code is returned.

SEE ALSO:

Appendix for argument description.

EXAMPLE(PYTHON):

Send:

```
server.send_cmd('ESL_DEL', [{'eslid':'5D-85-A8-99'}])
```

Return:

```
['OK', [{'eslid': '5D-85-A8-99', 'bak': '00'}]]
```

NETWORK PACKAGE:

```
POST /RPC2 HTTP/1.1
Host: 127.0.0.1:9000
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 337
<?xml version='1.0'?>
<methodCall>
<methodName>send_cmd</methodName>
<params>
<param>
```

```
<value><string>ESL_DEL</string></value>
</param>
<param>
<value><array><data>
<value><struct>
<member>
<name>eslid</name>
<value><string>5D-85-A8-99</string></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.6
Date: Fri, 20 Mar 2015 02:28:46 GMT
Content-type: text/xml
Content-length: 405
<?xml version='1.0'?>
<methodResponse>
<params>
<param>
<value><array><data>
<value><string>OK</string></value>
<value><array><data>
<value><struct>
<member>
<name>eslid</name>
<value><string>5D-85-A8-99</string></value>
</member>
<member>
<name>bak</name>
<value><string>00</string></value>
</member>
</struct></value>
</data></array></value>
</data></array></value>
</param>
</params>
</methodResponse>
```

1.4 ESL_QUERY

SYNOPSIS:

```
('ESL_QUERY',[{'eslid':'5A-B0-04-99'}, {…}])
```

esl_status **DESCRIPTION:**

Query price tag(s) information, the price tag(s) must binding to a product.

RETURN VALUE:

```
['OK',
    [
        {'salesname':'full frame','eslid':'5A-B0-04-99','salesno':'2000','
bak':'00',
        'apid':'1','Updata_status':'fail','Price3':'3','Price2':'64','Pric
e1':'666',
        'Price':'8','esl_status':'offline'},
        {…},
    ]
]
```

Upon successful completion this command return 'bak':'00'. Otherwise, an error code is returned.

SEE ALSO:

Appendix for argument description.

EXAMPLE(PYTHON):

Send:

```
server.send_cmd('ESL_QUERY', [{'eslid':'5A-B0-04-99'}])
```

Return:

```
['OK',
    [
        {'salesname':'full frame','eslid':'5A-B0-04-99','salesno':'2000','
bak':'00',
        'apid':'1','Updata_status':'fail','Price3':'3','Price2':'64','Pric
e1':'666',
        'Price':'8','esl_status':'offline'}
    ]
]
```

NETWORK PACKAGE:

```
POST /RPC2 HTTP/1.1
Host: 127.0.0.1:9000
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 339
<?xml version='1.0'?>
<methodCall>
<methodName>send_cmd</methodName>
<params>
<param>
<value><string>ESL_QUERY</string></value>
</param>
<param>
<value><array><data>
<value><struct>
<member>
<name>eslid</name>
<value><string>5A-B0-04-99</string></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.6
Date: Fri, 20 Mar 2015 05:49:46 GMT
Content-type: text/xml
Content-length: 1189
<?xml version='1.0'?>
<methodResponse>
```

```
<params>
<param>
<value><array><data>
<value><string>OK</string></value>
<value><array><data>
<value><struct>
<member>
<name>salesname</name>
<value><string>full frame</string></value>
</member>
<member>
<name>eslid</name>
<value><string>5A-B0-04-99</string></value>
</member>
<member>
<name>salesno</name>
<value><string>2000</string></value>
</member>
<member>
<name>Updata_time</name>
<value><string>20150320134532</string></value>
</member>
<member>
<name>Price</name>
<value><string>8</string></value>
</member>
<member>
<name>apid</name>
<value><string>1</string></value>
</member>
<member>
<name>Updata_status</name>
<value><string>fail</string></value>
</member>
<member>
<name>Price3</name>
<value><string>3</string></value>
</member>
<member>
<name>Price2</name>
<value><string>64</string></value>
</member>
<member>
<name>Price1</name>
<value><string>666</string></value>
```

```
</member>
<member>
<name>bak</name>
<value><string>00</string></value>
</member>
<member>
<name>esl_status</name>
<value><string>offline</string></value>
</member>
</struct></value>
</data></array></value>
</data></array></value>
</param>
</params>
</methodResponse>
```

1.5 AP_ADD

Access point(s) will be automatically added into ESL system by built-in heartbeat functionality.

1.6 AP_DEL

SYNOPSIS:

```
('AP_DEL',[{'apid':'1'}, {…}])
```

DESCRIPTION:

Delete access point(s)

RETURN VALUE:

```
['OK', [{'apid':'1','bak':'00'}, {…}]]
```

Upon successful completion this command return 'bak':'00'. Otherwise, an error code is returned.

SEE ALSO:

Appendix for argument description.

EXAMPLE(PYTHON):

Send:

```
server.send_cmd('AP_DEL', [{'apid':'1'}])
```

Return:

```
['OK', [{'apid':'1','bak':'00'}]]
```

NETWORK PACKAGE:

```
POST /RPC2 HTTP/1.1
Host: 127.0.0.1:9000
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 325
<?xml version='1.0'?>
<methodCall>
<methodName>send_cmd</methodName>
<params>
<param>
<value><string>AP_DEL</string></value>
</param>
<param>
<value><array><data>
<value><struct>
<member>
<name>apid</name>
<value><string>1</string></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.6
Date: Fri, 20 Mar 2015 06:28:58 GMT
Content-type: text/xml
Content-length: 394
<?xml version='1.0'?>
<methodResponse>
<params>
<param>
```

```
<value><array><data>
<value><string>OK</string></value>
<value><array><data>
<value><struct>
<member>
<name>apid</name>
<value><string>1</string></value>
</member>
<member>
<name>bak</name>
<value><string>00</string></value>
</member>
</struct></value>
</data></array></value>
</data></array></value>
</param>
</params>
</methodResponse>
```

1.7 AP_QUERY

SYNOPSIS:

```
('AP_QUERY',[{'apid':'1'}, {…}])
```

DESCRIPTION:

Query access point(s) status

RETURN VALUE:

```
['OK', [{'apid':'1','apip':'192.168.1.199','bak':'00','bs_status':'online'
}, {…}]]
```

Upon successful completion this command return 'bak':'00'. Otherwise, an error code is returned.

SEE ALSO:

Appendix for argument description.

EXAMPLE(PYTHON):

Send:

```
server.send_cmd('AP_QUERY', [{'apid':'1'}])
```

Return:

```
['OK', [{'apid':'1','apip':'192.168.1.199','bak':'00','bs_status':'online'
}]]
```

NETWORK PACKAGE:

```
POST /RPC2 HTTP/1.1
Host: 127.0.0.1:9000
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 327
<?xml version='1.0'?>
<methodCall>
<methodName>send_cmd</methodName>
<params>
<param>
<value><string>AP_QUERY</string></value>
</param>
<param>
<value><array><data>
<value><struct>
<member>
<name>apid</name>
<value><string>1</string></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.6
Date: Fri, 20 Mar 2015 06:16:40 GMT
Content-type: text/xml
Content-length: 558
<?xml version='1.0'?>
<methodResponse>
<params>
<param>
<value><array><data>
```

```
<value><string>OK</string></value>
<value><array><data>
<value><struct>
<member>
<name>apid</name>
<value><string>1</string></value>
</member>
<member>
<name>apip</name>
<value><string>192.168.1.199</string></value>
</member>
<member>
<name>bak</name>
<value><string>00</string></value>
</member>
<member>
<name>bs_status</name>
<value><string>online</string></value>
</member>
</struct></value>
</data></array></value>
</data></array></value>
</param>
</params>
</methodResponse>
```

1.8 BIND

SYNOPSIS:

```
("BIND", [{"eslid":"5A-B0-04-99","salesno":"2000", "apid":"1"}, {…}])
```

DESCRIPTION:

Build relation between product and price tag(s); one single product can bind to multiple tags, however, one tag is only allowed to bind to one product.

RETURN VALUE:

```
['OK', [{'salesname':'full frame','eslid':'5A-B0-04-99','salesno':'2000','
apid':'1','bak': '00','wait': '0'}, {…}]]
```

Upon successful completion this command return 'bak':'00'. Otherwise, an error code is

returned.

SEE ALSO:

Appendix for argument description.

EXAMPLE(PYTHON):

Send:

```
server.send_cmd("BIND", [{"eslid":"5A-B0-04-99","salesno":"2000", "apid":"1"}])
```

Return:

```
['OK', [{'salesname':'full frame','eslid':'5A-B0-04-99','salesno':'2000','apid':'1','bak': '00','wait': '0'}]]
```

NETWORK PACKAGE:

```
POST /RPC2 HTTP/1.1
Host: 127.0.0.1:9000
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 482
<?xml version='1.0'?>
<methodCall>
<methodName>send_cmd</methodName>
<params>
<param>
<value><string>BIND</string></value>
</param>
<param>
<value><array><data>
<value><struct>
<member>
<name>apid</name>
<value><string>1</string></value>
</member>
<member>
<name>eslid</name>
<value><string>5A-B0-04-99</string></value>
</member>
<member>
```

```
<name>salesno</name>
<value><string>2000</string></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.6
Date: Fri, 20 Mar 2015 06:44:58 GMT
Content-type: text/xml
Content-length: 709
<?xml version='1.0'?>
<methodResponse>
<params>
<param>
<value><array><data>
<value><string>OK</string></value>
<value><array><data>
<value><struct>
<member>
<name>salesname</name>
<value><string>full frame</string></value>
</member>
<member>
<name>eslid</name>
<value><string>5A-B0-04-99</string></value>
</member>
<member>
<name>salesno</name>
<value><string>2000</string></value>
</member>
<member>
<name>apid</name>
<value><string>1</string></value>
</member>
<member>
<name>bak</name>
<value><string>00</string></value>
</member>
<member>
<name>wait</name>
<value><string>0</string></value>
```

```
</member>
</struct></value>
</data></array></value>
</data></array></value>
</param>
</params>
</methodResponse>
```

1.9 UNBIND

SYNOPSIS:

```
("UNBIND", [{"eslid":"5A-B0-04-99"}, {…}])
```

DESCRIPTION:

Release relation between product and price tag(s).

RETURN VALUE:

```
['OK', [{'eslid': '5A-B0-04-99', 'bak': '00', 'wait': '0'}, {…}]]
```

Upon successful completion this command return 'bak':'00'. Otherwise, an error code is returned.

SEE ALSO:

Appendix for argument description.

EXAMPLE(PYTHON):

Send:

```
server.send_cmd("UNBIND", [{"eslid":"5A-B0-04-99"}])
```

Return:

```
['OK', [{'eslid': '5A-B0-04-99', 'bak': '00', 'wait': '0'}]]
```

NETWORK PACKAGE:

```
POST /RPC2 HTTP/1.1
Host: 127.0.0.1:9000
```

```
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 336
<?xml version='1.0'?>
<methodCall>
<methodName>send_cmd</methodName>
<params>
<param>
<value><string>UNBIND</string></value>
</param>
<param>
<value><array><data>
<value><struct>
<member>
<name>eslid</name>
<value><string>5A-B0-04-99</string></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.6
Date: Fri, 20 Mar 2015 07:08:19 GMT
Content-type: text/xml
Content-length: 476
<?xml version='1.0'?>
<methodResponse>
<params>
<param>
<value><array><data>
<value><string>OK</string></value>
<value><array><data>
<value><struct>
<member>
<name>eslid</name>
<value><string>5A-B0-04-99</string></value>
</member>
<member>
<name>bak</name>
<value><string>00</string></value>
</member>
```

```
<member>
<name>wait</name>
<value><string>0</string></value>
</member>
</struct></value>
</data></array></value>
</data></array></value>
</param>
</params>
</methodResponse>
```

1.10 SALES_UPDATA_BUF

SYNOPSIS:

```
("SALES_UPDATA_BUF", [{"sid":"100","salesno":"2000","Price":"333"}, {…}])
```

DESCRIPTION:

Update product(s) information and price tag(s) related. The price tag(s) will NOT update if it's in unbind stage.

RETURN VALUE:

```
['OK', [{'salesno':'2000','Price':'333','sid':'100','bak':'00'}, {…}]]
```

Upon successful completion this command return 'bak':'00'. Otherwise, an error code is returned.

SEE ALSO:

Appendix for argument description.

EXAMPLE(PYTHON):

Send:

```
server.send_cmd("SALES_UPDATA_BUF", [{"sid":"100", "salesno":"2000","Price":"333"}])
```

Return:

```
['OK', [{'salesno':'2000','Price':'333','sid':'100','bak':'00'}]]
```

NETWORK PACKAGE:

```
POST /RPC2 HTTP/1.1
Host: 127.0.0.1:9000
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 709
<?xml version='1.0'?>
<methodCall>
<methodName>send_cmd</methodName>
<params>
<param>
<value><string>SALES_UPDATA_BUF</string></value>
</param>
<param>
<value><array><data>
<value><struct>
<member>
<name>salesno</name>
<value><string>2000</string></value>
</member>
<member>
<name>sid</name>
<value><string>100</string></value>
</member>
<member>
<name>Price3</name>
<value><string>3</string></value>
</member>
<member>
<name>Price2</name>
<value><string>64</string></value>
</member>
<member>
<name>Price1</name>
<value><string>666</string></value>
</member>
<member>
<name>Price</name>
<value><string>333</string></value>
</member>
</struct></value>
</data></array></value>
</param>
```

```
</params>
</methodCall>

HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.6
Date: Fri, 20 Mar 2015 07:14:28 GMT
Content-type: text/xml
Content-length: 768
<?xml version='1.0'?>
<methodResponse>
<params>
<param>
<value><array><data>
<value><string>OK</string></value>
<value><array><data>
<value><struct>
<member>
<name>salesno</name>
<value><string>2000</string></value>
</member>
<member>
<name>bak</name>
<value><string>00</string></value>
</member>
<member>
<name>sid</name>
<value><string>100</string></value>
</member>
<member>
<name>Price3</name>
<value><string>3</string></value>
</member>
<member>
<name>Price2</name>
<value><string>64</string></value>
</member>
<member>
<name>Price1</name>
<value><string>666</string></value>
</member>
<member>
<name>Price</name>
<value><string>333</string></value>
</member>
</struct></value>
```

```
</data></array></value>
</data></array></value>
</param>
</params>
</methodResponse>
```

1.11 SALES_QUERY_ACK

SYNOPSIS:

```
("SALES_QUERY_ACK", [{"sid":"100","salesno":"2000","Price":"333","salesname":"Apple","bak":"00"}, {…}])
```

DESCRIPTION:

Add information of product(s) if "salesno" does NOT exist, or update information of product(s) if "salesno" exists. Any product(s) information feild(s) can be customized in configuration file named "config.ini".

RETURN VALUE:

```
OK
```

Upon successful completion this command return 'OK'. Otherwise, 'failed' is returned.

SEE ALSO:

Appendix for argument description.

EXAMPLE(PYTHON):

Send:

```
server.send_cmd("SALES_QUERY_ACK", [{"sid":"100","salesno":"2000","Price":"333","salesname":"Apple","bak":"00"}])
```

Return:

```
OK
```

NETWORK PACKAGE:

```
POST /RPC2 HTTP/1.1
```

Host: 127.0.0.1:9000
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 566
<?xml version='1.0'?>
<methodCall>
<methodName>send_cmd</methodName>
<params>
<param>
<value><string>SALES_QUERY_ACK</string></value>
</param>
<param>
<value><array><data>
<value><struct>
<member>
<name>salesname</name>
<value><string>Apple</string></value>
</member>
<member>
<name>salesno</name>
<value><string>2000</string></value>
</member>
<member>
<name>Price</name>
<value><string>333</string></value>
</member>
<member>
<name>sid</name>
<value><string>100</string></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.0 200 OK
Server: BaseHTTP/0.3 Python/2.7.6
Date: Fri, 20 Mar 2015 07:35:02 GMT
Content-type: text/xml
Content-length: 559
<?xml version='1.0'?>
<methodResponse>
<params>

```
<param>
<value><array><data>
<value><string>failed</string></value>
<value><array><data>
<value><struct>
<member>
<name>salesname</name>
<value><string>Apple</string></value>
</member>
<member>
<name>salesno</name>
<value><string>2000</string></value>
</member>
<member>
<name>Price</name>
<value><string>333</string></value>
</member>
<member>
<name>sid</name>
<value><string>100</string></value>
</member>
</struct></value>
</data></array></value>
</data></array></value>
</param>
</params>
</methodResponse>
```

1.12 SALES_DEL

SYNOPSIS:

```
("SALES_DEL", [{"salesno":"2000"}, {…}])
```

DESCRIPTION:

Delete product(s).

RETURN VALUE:

```
['OK', [{'salesno': '2000', 'bak': '00'}, {…}]]
```

Upon successful completion this command return 'bak':'00'. Otherwise, an error code is

returned.

SEE ALSO:

Appendix for argument description.

EXAMPLE(PYTHON):

Send:

```
server.send_cmd("SALES_DEL", [{"salesno":"2000"}])
```

Return:

```
['OK', [{'salesno': '2000', 'bak': '00'}]]
```

NETWORK PACKAGE:

```
POST /RPC2 HTTP/1.1
Host: 127.0.0.1:9000
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 334
<?xml version='1.0'?>
<methodCall>
<methodName>send_cmd</methodName>
<params>
<param>
<value><string>SALES_DEL</string></value>
</param>
<param>
<value><array><data>
<value><struct>
<member>
<name>salesno</name>
<value><string>2000</string></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.0 200 OK
```

```
Server: BaseHTTP/0.3 Python/2.7.6
Date: Fri, 20 Mar 2015 07:42:36 GMT
Content-type: text/xml
Content-length: 400
<?xml version='1.0'?>
<methodResponse>
<params>
<param>
<value><array><data>
<value><string>OK</string></value>
<value><array><data>
<value><struct>
<member>
<name>salesno</name>
<value><string>2000</string></value>
</member>
<member>
<name>bak</name>
<value><string>00</string></value>
</member>
</struct></value>
</data></array></value>
</data></array></value>
</param>
</params>
</methodResponse>
```

2. Uplink command

Uplink command is the command from ESL-Working to yours. The default post port is 8088, and '/shop-web/stationHandler' for default path.

2.1 ESL_ACTIVE

SYNOPSIS:

```
("ESL_ACTIVE",
    [
        {'nw4':'DOT20','op4':'TEMP_DOT_29','eslid':'5A-B0-04-99','op1':'1'
,'nw2':'52-56-78-53',
        'nw3': '11', 'lastworktime': 'Fri Mar 20 15:14:32 2015', 'op2': '1
', 'op3': 'HS_EL_5103',
        'apid': '1', 'status': 'online','salesno': '2000', 'nw1': u'5A-0B-
0B-66'},
        {…},
    ]
)
```

DESCRIPTION:

Send back ESL parameters, related product ID and access point ID when binding relationship changed.

RETURN VALUE:

```
OK
```

Upon successful completion, return 'OK' to ESL-Working. Otherwise, this command will send repeatly.

SEE ALSO:

Appendix for argument description.

NETWORK PACKAGE:

```
POST /shop-web/stationHandler HTTP/1.1
Host: 192.168.1.106:8088
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 1493
<?xml version='1.0'?>
<methodCall>
<methodName>stationHandler.send_cmd</methodName>
<params>
<param>
<value><string>ESL_ACTIVE</string></value>
</param>
<param>
<value><array><data>
```

```xml
<value><struct>
<member>
<name>nw4</name>
<value><string>DOT20</string></value>
</member>
<member>
<name>op4</name>
<value><string>TEMP_DOT_29</string></value>
</member>
<member>
<name>eslid</name>
<value><string>5A-B0-04-99</string></value>
</member>
<member>
<name>op1</name>
<value><string>1</string></value>
</member>
<member>
<name>nw2</name>
<value><string>52-56-78-53</string></value>
</member>
<member>
<name>nw3</name>
<value><string>11</string></value>
</member>
<member>
<name>lastworktime</name>
<value><string>Fri Mar 20 15:14:32 2015</string></value>
</member>
<member>
<name>op2</name>
<value><string>1</string></value>
</member>
<member>
<name>op3</name>
<value><string>HS_EL_5103</string></value>
</member>
<member>
<name>apid</name>
<value><string>1</string></value>
</member>
<member>
<name>status</name>
<value><string>online</string></value>
</member>
```

```
<member>
<name>salesno</name>
<value><string>2000</string></value>
</member>
<member>
<name>nw1</name>
<value><string>5A-0B-0B-66</string></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.1 200 OK
Server: Apache XML-RPC 1.0
Connection: close
Content-Type: text/xml
Content-Length: 181
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse xmlns:ex="http://ws.apache.org/xmlrpc/namespaces/extension
s">
<params>
<param>
<value>OK</value>
</param>
</params>
</methodResponse>
```

2.2 UPDATA_ACK

SYNOPSIS:

```
("UPDATA_ACK",
    [
        {'status': 'failed', 'ack': [{'status': 'offline', 'apid': '1', 'e
slid': '5A-B0-04-99'}],
        'salesno': '2000', 'sid': '100'},
        {…},
    ]
)
```

DESCRIPTION:

Send back update result and related information when ESL-Working finished "SALES_UPDATE_BUF" command transmission.

RETURN VALUE:

```
OK
```

Upon successful completion, return 'OK' to ESL-Working. Otherwise, this command will send repeatly.

SEE ALSO:

Appendix for argument description.

NETWORK PACKAGE:

```
POST /shop-web/stationHandler HTTP/1.1
Host: 192.168.1.106:8088
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 847
<?xml version='1.0'?>
<methodCall>
<methodName>stationHandler.send_cmd</methodName>
<params>
<param>
<value><string>UPDATA_ACK</string></value>
</param>
<param>
<value><array><data>
<value><struct>
<member>
<name>status</name>
<value><string>failed</string></value>
</member>
<member>
<name>ack</name>
<value><array><data>
<value><struct>
<member>
<name>status</name>
<value><string>offline</string></value>
</member>
<member>
```

```
<name>apid</name>
<value><string>1</string></value>
</member>
<member>
<name>eslid</name>
<value><string>5A-B0-04-99</string></value>
</member>
</struct></value>
</data></array></value>
</member>
<member>
<name>salesno</name>
<value><string>2000</string></value>
</member>
<member>
<name>sid</name>
<value><string>100</string></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.1 200 OK
Server: Apache XML-RPC 1.0
Connection: close
Content-Type: text/xml
Content-Length: 181
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse xmlns:ex="http://ws.apache.org/xmlrpc/namespaces/extension
s">
<params>
<param>
<value>OK</value>
</param>
</params>
</methodResponse>
```

2.3 AP_STATUS

SYNOPSIS:

```
("AP_STATUS", [{'apid': '1', 'status': 'online'}, {…}])
```

DESCRIPTION:

Regularly send access point(s) status following heartbeat functionality configuration.

RETURN VALUE:

```
OK
```

Upon successful completion, return 'OK' to ESL-Working. Otherwise, this command will send repeatly.

SEE ALSO:

Appendix for argument description.

NETWORK PACKAGE:

```
POST /shop-web/stationHandler HTTP/1.1
Host: 192.168.1.106:8088
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 421
<?xml version='1.0'?>
<methodCall>
<methodName>stationHandler.send_cmd</methodName>
<params>
<param>
<value><string>AP_STATUS</string></value>
</param>
<param>
<value><array><data>
<value><struct>
<member>
<name>apid</name>
<value><string>1</string></value>
</member>
<member>
<name>status</name>
<value><string>online</string></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.1 200 OK
Server: Apache XML-RPC 1.0
Connection: close
Content-Type: text/xml
Content-Length: 181
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse xmlns:ex="http://ws.apache.org/xmlrpc/namespaces/extension
s">
<params>
<param>
<value>OK</value>
</param>
</params>
</methodResponse>
```

2.4 ESL_STATUS

SYNOPSIS:

```
("ESL_STATUS", [{'status': 'offline', 'eslid': u'5A-B0-04-99'}, {…}]
```

DESCRIPTION:

Send ESL status when one update session finished or ESL status changed.

RETURN VALUE:

```
OK
```

Upon successful completion, return 'OK' to ESL-Working. Otherwise, this command will send repeatly.

SEE ALSO:

Appendix for argument description.

NETWORK PACKAGE:

```
POST /shop-web/stationHandler HTTP/1.1
Host: 192.168.1.106:8088
Accept-Encoding: gzip
User-Agent: xmlrpclib.py/1.0.1 (by www.pythonware.com)
Content-Type: text/xml
Content-Length: 434
<?xml version='1.0'?>
<methodCall>
<methodName>stationHandler.send_cmd</methodName>
<params>
<param>
<value><string>ESL_STATUS</string></value>
</param>
<param>
<value><array><data>
<value><struct>
<member>
<name>status</name>
<value><string>offline</string></value>
</member>
<member>
<name>eslid</name>
<value><string>5A-B0-04-99</string></value>
</member>
</struct></value>
</data></array></value>
</param>
</params>
</methodCall>

HTTP/1.1 200 OK
Server: Apache XML-RPC 1.0
Connection: close
Content-Type: text/xml
Content-Length: 181
<?xml version="1.0" encoding="UTF-8"?>
<methodResponse xmlns:ex="http://ws.apache.org/xmlrpc/namespaces/extension
s">
<params>
<param>
<value>OK</value>
/param>
</params>
</methodResponse>
```

Appendix 1: ESL argument description

ARGUMENT	DEFINITION	FORMAT	NOTE	LEVEL
eslid	ESL ID	XX-XX-XX-XX	keep default	necessary
nw1	communication parameter	XX-XX-XX-XX	keep default	optional
nw2	communication parameter	XX-XX-XX-XX	keep default	optional
nw3	communication parameter	number	keep default	optional
nw4	ESL type	DOT20 / NORMAL	DOT20: dot matrix labels NORMAL: segment labels	optional
op1	configuration parameter	number	keep default	optional
op2	configuration parameter	number	keep default	optional
op3	ESL type	HS_EL_5102 /	keep default	optional
op4	template name	text	must select from template file	optional
op5	configuration parameter	text	keep default	optional
op6	configuration parameter	text	keep defaul	optional
status	ESL status	offline / online / lowpower / updating	keep default	optional

Appendix 2: Access Point argument description

ARGUMENT	DEFINITION	FORMAT	NOTE	LEVEL
apid	access point ID	XX-XX-XX-XX	keep default	necessary
apip	access point IP	XXX.XXX.XXX.XXX	keep default	optional
status	access point status	offline / online	keep default	optional

Appendix 3: "bak" values definition

VALUE	DEFINITION
00	command excuted successfully
01	database error
02	ESL ID error
03	product ID error
04	busy
05	AP ID error
06	product needs to query