

# OIX Overdelivery Test Plan

## Information

### Test Plan Author:

[Sergey Gorelov](#)

### Also worked on test plan

[Sergey Gorelov](#) , [Albina Khyrullina](#) , [Mikhail Polyantsev](#)

Please, address all test plan related issues or questions to its author

Please note: if something wrong occurred with stats, verify generated stat processing from ad-server to oix. More detailed info is available [here](#).

- [Test 1. Overdelivery While Using Usual Budget](#)
  - [Test 1.1. Usual Budget](#)
    - [Preparing of Environment](#)
    - [Test 1.1.1. Full Spent and Overdelivery Budget](#)
    - [Test 1.1.2. Check Stats Correction after detection of Fraud Condition](#)
  - [Test 1.2. Usual Budget & Campaign Credit](#)
    - [Test 1.2.1 Full Spent and Overdelivery Budget \(see OUI-26289\)](#)
    - [Test 1.2.2. Check Stats Correction after detection of Fraud Condition](#)
- [Test 2. Overdelivery While Using I/O Budget](#)
  - [Test 2.1. I/O Budget](#)
    - [Test 2.1.1. Full Spent and Overdelivery Budget](#)
    - [Test 2.1.2. Check Stats Correction after detection of Fraud Condition](#)
  - [Test 2.2. Campaign Credit & I/O Budget \(see OUI-26289\)](#)
    - [Test 2.2.1. Full Spent and Overdelivery Budget](#)
    - [Test 2.2.2. Check Stats Correction after detection of Fraud Condition](#)
- [Test 3. I/O Budget Sharing \(Will Be Realised in 3.5\)](#)

## Common Rule

After all manipulations (adding budget, spending budget and adding new) **Campaign Budget** always must be equal to **Spent Budget + Available Budget**  
If it is not, it means there is a reason to create a new bug

## Note

To prepare and execute tests it is not necessary to use values specified in test plan. You can use your own, but they should not be very large. This is because of fact, that we have limited time to get statistics.

### Testing on attest colo

In some cases more convenient and faster to pass a test on attest colo. All too same as on test colo, but additional every chapter with the receipt of statistics contains an expanded block with description how to get stats on attest colo

## Test 1. Overdelivery While Using Usual Budget

### Test 1.1. Usual Budget

#### Preparing of Environment

1. Open any SSH client, similar to PuTTY or WinSCP
2. Login as [firstname\\_lastname@nnode0.ocslab.com](mailto:firstname_lastname@nnode0.ocslab.com)
3. Find **.bashrc** file in your personal directory (for example /home/sergey\_gorelov). Open it for editing and check presence of the next rows:

```
# User specific environment and startup programs

PATH=$PATH:$HOME/bin
. /opt/oix/server/etc/moscow-test-central/adcluster/environment.sh
. /opt/oix/server/etc/moscow-test-central/adcluster/nnode0/CurrentEnv/fe.sh
. /opt/oix/server/etc/moscow-test-central/adcluster/nnode0/CurrentEnv/be.sh
export PATH
```

4. If such rows not exist it is necessary to add them (if you use PuTTY it is best to do it in the Midnight Commander)
5. Close SSH client
6. Open three of different browsers **Firefox**, **Google Chrome** and **Internet Explorer**
7. In all browsers clear cach and cookies
8. Go to browser with opened **Firefox**, we will use it to send requests from **Opted Out** user
9. Open [enswitch](#) page, select **itm (apm1)** and click Submit
10. Enter into address bar the following request and execute it  
[http://cs.ocslab.com/cgi-bin/moo.cgi?op=out&success\\_url=http://cs.ocslab.com/opt-out/success.html&already\\_url=http://cs.ocslab.com/opt-out/already.html&fail\\_url=http://cs.ocslab.com/opt-out/fail.html](http://cs.ocslab.com/cgi-bin/moo.cgi?op=out&success_url=http://cs.ocslab.com/opt-out/success.html&already_url=http://cs.ocslab.com/opt-out/already.html&fail_url=http://cs.ocslab.com/opt-out/fail.html)
11. Open [enswitch](#) page again, select **direct** and click Submit
12. First browser ready for sending requests
13. Go to browser with opened **Google Chrome**, we will use it for send requests from user **Without Status** (it will be good if you first install Web Developer extention, you can find it on <http://chrispederick.com/work/web-developer/>)
14. Open **Web Developer** console, find **Cookies** tab and click **Disable** cookies
15. Close **Web Developer** console.
16. Second browser ready for sending requests
17. Go to browser with opened **Internet Explorer**, we will use it for send requests from **Opted In** user
18. Enter into address bar the following request and execute it  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&setuid=1>

19. Third browser ready for sending requests

### Test 1.1.1. Full Spent and Overdelivery Budget

#### Important Remark

Here and in the following tests we need to execute requests until overdelivery of the Campaign Budget will not happen. Be very careful, in order to **avoid** causing fraud condition triggering. For this it is recommended to do small pauses between requests, approximately from 1 to 3 seconds. Do not obtain more than 20 impressions in 1 browser without clearing cache and cookies. However, if you got a passback, generate uncompleted invoice, clear cache and cookies and then repeat everything again with correct pauses between requests.

If described method has not led to success, there is one more method to get overdelivery without fraud condition: it is needed to do the fixed number of requests to issue budget and not cause the fraud condition. Wait a while until changes will not applied to all hosts (It may take about 5 minutes), then execute one more request. You can receive passback with **Campaign eval status 'I'**

#### Getting stats on test colo

1. Login as [test@ocslab.com](mailto:test@ocslab.com)
2. Open [Campaign](#) view page and check its status
3. If Status is **Not Live - No Available Budget** do so that **Available Budget** became equal to \$5
4. Remember values of **Budget**, **Spent Budget** and **Available budget**
5. Fill in address bar of first browser from environment (**Firefox**) with adRequest and execute it arbitrary number of times, but not less than 10 (it need to receipt of some number of passbacks)  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217796&colo=58905&referer-kw=ouioverdelivery-us-1&loc.name=hr>
6. Just fill address bar of first browser from environment (**Firefox**) with adRequest, do not execute it:  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217796&colo=58905&referer-kw=ouioverdelivery-us-1&loc.name=us>
7. Fill address bar of second browser from environment (**Google Chrome**) with adRequest:  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217796&colo=58905&referer-kw=ouioverdelivery-us-2&loc.name=ru>
8. Fill address bar of third browser from environment (**IE**) with adRequest:  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217796&colo=58905&referer-kw=ouioverdelivery-us-1&loc.name=us>
9. Execute adRequests in first Browser 2 times, in second Browser 2 times and in third Browser so much times to get passback by overdelivery reason (see note at beginning of test)

#### ▼ Getting stats on attest colo

- Run <http://vauto.ocslab.com:8080/testfunctional/> application in any browser
- Select **Host**: [stat-attest.ocslab.com](http://stat-attest.ocslab.com) and **Database**: [stat](#)
- Use **Login**: test\_ui and **Password**: adserver
- Click **Begin**, "select test\_functional.saverequeststatshourly\_begin();" command should appear in the

text box

- Add one row with parameters:

<b>Campaign Creative ID:</b>	404934
<b>Impressions:</b>	2
<b>adv_amount:</b>	2

- Add one more row with parameters:

<b>Campaign Creative ID:</b>	404935
<b>Impressions:</b>	17
<b>adv_amount:</b>	17

- Actually you can add any number of rows in any sequence of CCID, but total amount of impressions must not exceed 19
- Click **Finalize**, "test\_functional.saverequeststatshourly\_finalize();" command should appear in the text box
- Click **Execute**
- If your request has been designed properly you should receive popup window with message "Request successfully executed", otherwise you will get error message
- Further you must execute the same verification as on test colo

## Verification of Results

10. Wait a while to stats appeared in a database
11. Open [Campaign](#) view page and check status
12. Make sure that it is **Not Live - No Available Budget**
13. Make sure that **Available Budget** = 0
14. Make sure that **Spent Budget** is equal to sum of **Total Value** for all **Creative Groups**
15. Check Invoices section
16. Make sure that new uncompleted invoice with **Total Amount Payable** = USD5.00 appeared
17. Open Invoice
18. Make sure that all unsecured budget impressions were compensated by a credit
19. Click [History](#) in **Campaign Allocations** section
20. Make sure that one or few lines with credit appeared on a missing amount
21. Open [Account](#) and check **Campaign Credits** section
22. Make sure that new Campaign Credit with **Type**: "Compensation" and **Description**: "Automatic credit for campaign 193774 budget overrun" was added
23. Make sure that **Amount** is equal to **Utilised Amount** from **Campaign Allocation History**
24. Open your **SSH** client, login and execute the command:  
  
CampaignAdmin campaign id=310802 -r [corbaloc:iiop:localhost:\\$campaign\\_manager\\_port/CampaignManager](#)  
[er](#)
25. Make sure that eval\_status = I
26. Make sure that Campaign Budget = cmp\_budget

## Adding a New Funds to Budget

27. Open [Campaign](#) edit page
28. Enter into **Budget** value equal to **Old Budget** + 5 and press **Save**

### Verification of Results

29. Go to **General Properties** section
30. Make sure that status became **Live**
31. Make sure that:  
**Available Budget = Budget - Spent Budget**
32. Wait a while until changes will not applied to all hosts
33. Open your **SSH** client, login and execute the command:  
  
CampaignAdmin campaign id=310802 -r [corbaloc:iop:localhost:\\$campaign\\_manager\\_port/CampaignManager](#)
34. Make sure that eval\_status = A
35. Make sure that Campaign Budget = cmp\_budget
36. Make sure that **adServer** successfully processed requests, for this execute any request in any browser from  
stes 5 to 6
37. Generate Uncompleted Invoice

### Check merger.log

38. Connect via ssh (for example, with PuTTY) to the **stat\_test** host
39. Go to /u01/oix/pgdb/var/log/stat\_test/merger folder and open **merger.log** file
40. If you got any record for the nearest period containing **ERROR** it means there is a reason to create a new  
major PGDB bug

▼ Example of such record

[Mon Oct 5 09:25:27 2015] ERROR: thread-actionstats encountered an error:

Traceback (most recent call last):

File "/opt/oix/pgdb/lib/merger/worker.py", line 180, in run  
self.\_process()

File "/opt/oix/pgdb/lib/merger/worker.py", line 160, in \_process  
self.\_process\_file(\*(entries[0]))

File "/opt/oix/pgdb/lib/merger/worker.py", line 139, in \_process\_file  
self.\_upload\_file(filepath, key)

File "/opt/oix/pgdb/lib/merger/worker.py", line 116, in \_upload\_file  
self.\_execute(statement)

File "/opt/oix/pgdb/lib/merger/worker.py", line 84, in \_execute  
cursor.execute(statement)

ProgrammingError: column "days\_since\_imp" does not exist

LINE 6: min(days\_since\_imp) min\_days\_since\_imp,  
^

QUERY: with input as (

```
select
  advertiser_action_date,
  action_id,
  action_date,
  min(days_since_imp) min_days_since_imp,
  min(days_since_click) min_days_since_click,
  count(distinct action_request_id) conversions,
```

```

sum(cur_value) cur_value
from ( select action_request_id,
        action_id,
        action_date,
        advertiser_action_date,
        min(days_since_imp) min_days_since_imp,
        min(days_since_click) min_days_since_click,
        count(distinct action_request_id) conversions,
        avg(cur_value) cur_value -- avg just for getting distinct cur_value
        from actionstats_new_and_history
        group by 1,2,3,4
    )_
group by 1, 2, 3
),
updated as (
    update actionstatsdailyadvertiser tgt
    set conversions = src.conversions,
        cur_value = src.cur_value,
        min_days_since_imp = src.min_days_since_imp,
        min_days_since_click = src.min_days_since_click
    from input src
    where
        tgt.advertiser_action_date = src.advertiser_action_date
        and tgt.action_id = src.action_id
        and tgt.action_date = src.action_date
    returning tgt.action_date, tgt.advertiser_action_date, tgt.action_id
)
insert into actionstatsdailyadvertiser (action_date, advertiser_action_date, action_id, min_days_since_imp,
min_days_since_click, conversions, cur_value)
select action_date, advertiser_action_date, action_id, min_days_since_imp, min_days_since_click,
conversions, cur_value
from input
left join updated using (action_date, advertiser_action_date, action_id)
where updated.action_date is null
CONTEXT: PL/pgSQL function statdml.save_action_stats() line 185 at SQL statement
SQL statement "SELECT statdml.save_action_stats()"
PL/pgSQL function inline_code_block line 23 at PERFORM

```

## Test 1.1.2. Check Stats Correction after detection of Fraud Condition

### Preparing Test

1. Open [Campaign](#) view page and remember values of **Budget**, **Spent Budget** and **Available budge**

### Getting Stats

2. Fill in address bar of **IE** browser (**Opted In** user) with adRequest and execute it one time:

<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&colo=58905&referer-kw=ouioverdelivery-us-1&loc.name=us>

3. Execute following Ad Request (as result you should receive 25 banners)

<http://cs.ocslab.com/cgi-bin/arbitrage.cgi?size=300x250&colo=58905&tid=217796&loc.name=us&t=fraud1.tm>

[pl&t.n=25](#)

▼ Getting stats on attest colo

- Run <http://vauto.ocslab.com:8080/testfunctional/> application in any browser
- Select **Host:** [stat-attest.ocslab.com](#) and **Database:** [stat](#)
- Use **Login:** test\_ui and **Password:** adserver
- Click **Begin**
- Add one row with parameters:

<b>Campaign Creative ID:</b>	404935
<b>Impressions:</b>	25
<b>adv_amount:</b>	25
<b>Fraud Correction:</b>	<input type="checkbox"/>

- Add one more row with parameters:

<b>Campaign Creative ID:</b>	404935
<b>Impressions:</b>	25
<b>adv_amount:</b>	25
<b>Fraud Correction:</b>	<input checked="" type="checkbox"/>

- Click **Finalize** than click **Execute**
- If your request has been designed properly you should receive popup window with message "Request successfully executed", otherwise you will get error message
- Further you must execute the same verification as on test colo

### Verification of Results

4. Open [Campaign](#) view page
5. Make sure that there is new **Uncompleted Invoice** with Total Amount Payable equal to 0 and with message "Nothing found to display" in **Invoice Data** section
6. Make sure that values of **Budget**, **Spent Budget** and **Available Budget** did not change

## Test 1.2. Usual Budget & Campaign Credit

### Test 1.2.1 Full Spent and Overdelivery Budget (see [OUI-26289](#))

#### Preparing Test

1. Open [Campaign](#) view page and check its status
2. If Status is **Not Live - No Available Budget** do so that **Available Budget** became equal to \$5 and consisted from two approximately equal parts, one **Campaign Credit** and two **Usual Campaign Budget**, for this open [Campaign Credit Allocation](#) edit page and add some amount (for example 2) then open [Campaign](#) edit page and add some amount (for example 3)  
Numbers can be any, but it is better to use small values

- Remember values of **Budget**, **Spent Budget** and **Available budget**

### Getting Stats

- Fill in address bar of first browser from environment (**Firefox**) with adRequest and execute it arbitrary number of times, but not less than 10 (it need to receipt of some number of passbacks)

<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217797&colo=58905&referer-kw=ouioverdelivery-us-3&loc.name=hr>

- Just fill address bar of first browser from environment (**Firefox**) with adRequest, do not execute it:

<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217797&colo=58905&referer-kw=ouioverdelivery-us-3&loc.name=cn>

- Fill address bar of second browser from environment (**Google Chrome**) with adRequest:

<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217797&colo=58905&referer-kw=ouioverdelivery-us-4&loc.name=gb>

- Fill address bar of third browser from environment (**IE**) with adRequest:

<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217797&colo=58905&referer-kw=ouioverdelivery-us-3&loc.name=cn>

- Execute adRequests in first Browser 2 times, in second Browser 2 times and in third Browser so much times to get passback by overdelivery reason (see note in **Test 1.1.1.**)

#### ▼ Getting stats on attest colo

- Run <http://vauto.ocslab.com:8080/testfunctional/> application in any browser
- Select **Host:** [stat-attest.ocslab.com](http://stat-attest.ocslab.com) and **Database:** [stat](#)
- Use **Login:** test\_ui and **Password:** adserver
- Click **Begin**
- Add one row with parameters:

<b>Campaign Creative ID:</b>	404936
<b>Impressions:</b>	2
<b>adv_amount:</b>	2

- Add one more row with parameters:

<b>Campaign Creative ID:</b>	404937
<b>Impressions:</b>	17
<b>adv_amount:</b>	17

- Actually you can add any number of rows in any sequence of CCID, but total amount of impressions must not exceed 19
- Click **Finalize** then click **Execute**
- If your request has been designed properly you should receive popup window with message "Request successfully executed", otherwise you will get error message
- Further you must execute the same verification as on test colo



## Verification of Results

9. Wait a while to stats appeared in a database
10. Open [Campaign](#) view page and check status
11. Make sure that it is **Not Live - No Available Budget**
12. Make sure that **Available Budget** = 0
13. Go to **Creative Groups** section, select **Entire History** in date picker
14. Make sure that **Spent Budget** is equal to sum of **Total Value** for all **Creative Groups**
15. Check **Invoices** section
16. Make sure that new uncompleted invoice with **Total Amount Payable** = **Usual Campaign Budget** appeared
17. Open Invoice
18. Make sure that all unsecured budget impressions were compensated by a credit
19. Click [History](#) in **Campaign Allocations** section
20. Make sure that one or few lines with credit appeared on a missing amount
21. Open [Account](#) and check **Campaign Credits** section
22. Make sure that new Campaign Credit with **Type**: "Compensation" and **Description**: "Automatic credit for campaign 193775 budget overrun" was added
23. Make sure that **Amount** is equal to **Utilised Amount** from **Campaign Allocation History**
24. Open your **SSH** client, login and execute the command:  
  
CampaignAdmin campaign id=310804 -r [corbaloc:iiop:localhost:\\$campaign\\_manager\\_port/CampaignManager](#)
25. Make sure that eval\_status = I
26. Make sure that Campaign Budget = cmp\_budget

## Adding a New Funds to Budget

27. Open last Uncompleted invoice and leave it open  
Make sure that **Invoice Data** contains two row - one with Unit Price > 0 related to **Usual Budget** and two with Unit Price = 0 related to **Campaign Credit**
28. Open [Campaign Credit Allocation](#) edit page and enter new **Amount** equal to **Old Amount + 3**
29. Open [Campaign](#) view page
30. Make sure that **Campaign Allocations** section contain one Campaign Credit row with Available Amount equal to 3
31. Open [Campaign](#) edit page
32. Enter into **Budget** value equal to **Old Amount + 2** and press **Save**

## Verification of Results

33. Go to **General Properties** section
34. Make sure that status became **Live**
35. Check **Available Budget**
36. Make sure that it equal to \$5 and can be calculated by formula **Available Budget = Budget - Spent Budget**
37. Wait a while until changes will not applied to all hosts
38. Open your **SSH** client, login and execute the command:

CampaignAdmin campaign id=310804 -r [corbaloc:iiop:localhost:\\$campaign\\_manager\\_port/CampaignManager](#)

39. Make sure that eval\_status = A
40. Make sure that Campaign Budget = cmp\_budget
41. Make sure that **adServer** successfully processed requests, for this execute any request in any browser from steps 5 to 6
42. Generate Uncompleted Invoice

## Test 1.2.2. Check Stats Correction after detection of Fraud Condition

### Preparing Test

1. Open [Campaign](#) view page and remember values of **Budget**, **Spent Budget** and **Available budget**
2. Open [Reports](#) tab and execute [ISP](#), [Campaign Overview](#) and [Publisher Overview](#) reports for today
3. Remember values or better to leave open the tabs with result

### Getting Stats

4. Fill in address bar of **IE** browser (**Opted In** user) with adRequest and execute it one time:  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&colo=58905&referer-kw=ouioverdelivery-us-4&loc.name=gb>
5. Execute following Ad Request (as result you should receive 25 banners)  
<http://cs.ocslab.com/cgi-bin/arbitrage.cgi?size=300x250&colo=58905&tid=217797&loc.name=gb&t=fraud1.tpl&t.n=25>

#### ▼ Getting stats on attest colo

- Run <http://vauto.ocslab.com:8080/testfunctional/> application in any browser
- Select **Host:** [stat-attest.ocslab.com](http://stat-attest.ocslab.com) and **Database:** [stat](#)
- Use **Login:** test\_ui and **Password:** adserver
- Click **Begin**
- Add one row with parameters:

<b>Campaign Creative ID:</b>	404937
<b>Impressions:</b>	25
<b>adv_amount:</b>	25
<b>Fraud Correction:</b>	<input type="checkbox"/>

- Add one more row with parameters:

<b>Campaign Creative ID:</b>	404937
<b>Impressions:</b>	25
<b>adv_amount:</b>	25
<b>Fraud Correction:</b>	<input checked="" type="checkbox"/>

- Click **Finalize** than click **Execute**
- If your request has been designed properly you should receive popup window with message "Request successfully executed", otherwise you will get error message
- Further you must execute the same verification as on test colo

## Verification of Results

6. Open [Campaign](#) view page
7. Make sure that **Available Budget** decreased by 7
8. Make sure that Campaign Allocations section is empty
9. Make sure that there is a new **Uncompleted Invoice**
10. Open it and make sure that Total Amount Payable is equal to 0 and **Invoice Data** section contain "Nothing found to display" message
11. Open [Reports](#) tab and execute [ISP](#), [Campaign Overview](#) and [Publisher Overview](#) reports for today
12. Compare values with the same in step 5
13. Make sure that values have not changed

## Test 2. Overdelivery While Using I/O Budget

### Test 2.1. I/O Budget

#### Test 2.1.1. Full Spent and Overdelivery Budget

##### Getting Stats

1. Open [Campaign](#) view page and check it status
2. If Status is **Not Live - No Available Budget** do so that **Available Budget** became equal to \$5: for this use [Opportunity](#) and [Campaign Allocations](#) edit page
3. Remember values of **Budget**, **Spent Budget** and **Available budget**
4. Fill in address bar of first browser from environment (**Firefox**) with adRequest and execute it arbitrary number of times, but not less than 10 (it need to receipt of some number of passbacks)  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217797&colo=58905&referer-kw=ouioverdelivery-us-7&loc.name=hr>
5. Just fill address bar of first browser from environment (**Firefox**) with adRequest, do not execute it:  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217797&colo=58905&referer-kw=ouioverdelivery-us-7&loc.name=ru>
6. Fill address bar of second browser from environment (**Google Chrome**) with adRequest:  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217797&colo=58905&referer-kw=ouioverdelivery-us-8&loc.name=us>
7. Fill address bar of third browser from environment (**IE**) with adRequest:  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217797&colo=58905&referer-kw=ouioverdelivery-us-7&loc.name=ru>
8. Execute adRequests in first Browser 2 times, in second Browser 2 times and in third Browser so much times to get passback by overdelivery reason (see note in **Test 1.1.1.**)
  - ▼ Getting stats on attest colo
    - Run <http://vauto.ocslab.com:8080/testfunctional/> application in any browser

- Select **Host:** [stat-attest.ocslab.com](http://stat-attest.ocslab.com) and **Database:** [stat](#)
- Use **Login:** test\_ui and **Password:** adserver
- Click **Begin**
- Add one row with parameters:

<b>Campaign Creative ID:</b>	405074
<b>Impressions:</b>	2
<b>adv_amount:</b>	2

- Add one more row with parameters:

<b>Campaign Creative ID:</b>	405075
<b>Impressions:</b>	17
<b>adv_amount:</b>	17

- Actually you can add any number of rows in any sequence of CCID, but total amount of impressions must not exceed 19
- Click **Finalize** then click **Execute**
- If your request has been designed properly you should receive popup window with message "Request successfully executed", otherwise you will get error message
- Further you must execute the same verification as on test colo

### Verification of Results

9. Wait a while to stats appeared in a database
10. Open [Campaign](#) view page and check status
11. Make sure that it is **Not Live - No Available Budget**
12. Make sure that **Available Budget** = 0
13. Go to **Creative Groups** section, select **Entire History** in date picker
14. Make sure that **Spent Budget** is equal to sum of **Total Value** for all **Creative Groups**
15. Check **Invoices** section
16. Make sure that new uncompleted invoice with **Total Amount Payable = I/O Budget** appeared
17. Open Invoice
18. Make sure that all unsecured budget impressions were compensated by a credit
19. Make sure that **Campaign Allocations** section contain "No Campaign Allocations" message
20. Click [History](#)
21. Make sure that one or few lines with credit appeared on a missing amount
22. Open [Account](#) and check **Campaign Credits** section
23. Make sure that new Campaign Credit with **Type:** "Compensation" and **Description:** "Automatic credit for campaign 193834 budget overrun" was added
24. Make sure that **Amount** is equal to **Utilised Amount** from **Campaign Allocation History**
25. Open your **SSH** client, login and execute the command:  
  
CampaignAdmin campaign id=310862 -r [corbaloc:iiop:localhost:\\$campaign\\_manager\\_port/CampaignManager](#)
26. Make sure that eval\_status = I
27. Make sure that Campaign Budget = cmp\_budget

## Adding a New Funds to Budget

28. Use [Opportunity](#) and [Campaign Allocations](#) edit page to make available campaign budget equal to 5

### Verification of Results

29. Go to **General Properties** section
30. Make sure that status became **Live**
31. Make sure that:  
**Available Budget = Budget - Spent Budget**
32. Wait a while until changes will not applied to all hosts
33. Open your **SSH** client, login and execute the command:  
  
CampaignAdmin campaign id=310862 -r [corbaloc:iiop:localhost:\\$campaign\\_manager\\_port/CampaignManager](#)
34. Make sure that eval\_status = A
35. Make sure that Campaign Budget = cmp\_budget
36. Make sure that **adServer** successfully processed requests, for this execute any request in any browser from steps 5 to 6
37. Generate Uncompleted Invoice

## Test 2.1.2. Check Stats Correction after detection of Fraud Condition

### Preparing Test

1. Open [Campaign](#) view page and remember values of **Budget**, **Spent Budget** and **Available budget**

### Getting Stats

2. Fill in address bar of **IE** browser (**Opted In** user) with adRequest and execute it one time:  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&colo=58905&referer-kw=ouioverdelivery-us-7&loc.name=ru>
3. Execute following Ad Request (as result you should receive 25 banners)  
<http://cs.ocslab.com/cgi-bin/arbitrage.cgi?size=300x250&colo=58905&tid=217797&loc.name=ru&t=fraud1.tpl&t.n=25>

▼ Getting stats on attest colo

- Run <http://vauto.ocslab.com:8080/testfunctional/> application in any browser
- Select **Host**: [stat-attest.ocslab.com](http://stat-attest.ocslab.com) and **Database**: [stat](#)
- Use **Login**: test\_ui and **Password**: adserver
- Click **Begin**
- Add one row with parameters:

<b>Campaign Creative ID:</b>	405075
<b>Impressions:</b>	25
<b>adv_amount:</b>	25
<b>Fraud Correction:</b>	<input type="checkbox"/>

- Add one more row with parameters:

<b>Campaign Creative ID:</b>	405075
<b>Impressions:</b>	25
<b>adv_amount:</b>	25
<b>Fraud Correction:</b>	<input checked="" type="checkbox"/>

- Click **Finalize** than click **Execute**
- If your request has been designed properly you should receive popup window with message "Request successfully executed", otherwise you will get error message
- Further you must execute the same verification as on test colo

### Verification of Results

4. Open [Campaign](#) view page
5. Make sure that there is new **Uncompleted Invoice** with Total Amount Payable equal to 0 and with message "Nothing found to display" in **Invoice Data** section
6. Make sure that values of **Budget**, **Spent Budget** and **Available Budget** did not change

## Test 2.2. Campaign Credit & I/O Budget (see [OUI-26289](#))

### Test 2.2.1. Full Spent and Overdelivery Budget

#### Getting Stats

1. Open [Campaign](#) view page and check its status
2. If Status is **Not Live - No Available Budget** do so that **Available Budget** became equal to \$5 and consisted of two parts (IO and Credit), for this use [Campaign Allocations](#) edit page (Use I/O# 1) and [Campaign Credit Allocation](#) edit page
3. Remember values of **Budget**, **Spent Budget** and **Available budget**
4. Fill in address bar of first browser from environment (**Firefox**) with adRequest and execute it arbitrary number of times, but not less than 10 (it need to receipt of some number of passbacks)  
  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217797&colo=58905&referer-kw=ouioverdelivery-us-5&loc.name=hr>
5. Just fill address bar of first browser from environment (**Firefox**) with adRequest, do not execute it:  
  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217797&colo=58905&referer-kw=ouioverdelivery-us-5&loc.name=cn>
6. Fill address bar of second browser from environment (**Google Chrome**) with adRequest:  
  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217797&colo=58905&referer-kw=ouioverdelivery-us-6&loc.name=ru>
7. Fill address bar of third browser from environment (**IE**) with adRequest:  
  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217797&colo=58905&referer-kw=ouioverdelivery-us-5&loc.name=cn>
8. Execute adRequests in first Browser 2 times, in second Browser 2 times and in third Browser so much times

to get passback by overdelivery reason (see note in **Test 1.1.1.**)

▼ Getting stats on attest colo

- Run <http://vauto.ocslab.com:8080/testfunctional/> application in any browser
- Select **Host:** [stat-attest.ocslab.com](http://stat-attest.ocslab.com) and **Database:** [stat](#)
- Use **Login:** test\_ui and **Password:** adserver
- Click **Begin**
- Add one row with parameters:

<b>Campaign Creative ID:</b>	404877
<b>Impressions:</b>	2
<b>adv_amount:</b>	2

- Add one more row with parameters:

<b>Campaign Creative ID:</b>	404878
<b>Impressions:</b>	17
<b>adv_amount:</b>	17

- Actually you can add any number of rows in any sequence of CCID, but total amount of impressions must not exceed 19
- Click **Finalize** then click **Execute**
- If your request has been designed properly you should receive popup window with message "Request successfully executed", otherwise you will get error message
- Further you must execute the same verification as on test colo

## Verification of Results

9. Wait a while to stats appeared in a database
10. Open [Campaign](#) view page and check status
11. Make sure that it is **Not Live - No Available Budget**
12. Make sure that **Available Budget** = 0
13. Go to **Creative Groups** section, select **Entire History** in date picker
14. Make sure that **Spent Budget** is equal to sum of **Total Value** for all **Creative Groups**
15. Check **Invoices** section
16. Make sure that new uncompleted invoice with **Total Amount Payable** = **I/O Budget** appeared
17. Open Invoice
18. Make sure that all unsecured budget impressions were compensated by a credit
19. Make sure that **Campaign Allocations** section contain "No Campaign Allocations" message
20. Click [History](#)
21. Make sure that one or few lines with credit appeared on a missing amount
22. Open [Account](#) and check **Campaign Credits** section
23. Make sure that new Campaign Credit with **Type:** "Compensation" and **Description:** "Automatic credit for campaign 193716 budget overrun" was added
24. Make sure that **Amount** is equal to **Utilised Amount** from **Campaign Allocation History**
25. Open your **SSH** client, login and execute the command:  
  
CampaignAdmin campaign id=310746 -r [corbaloc:iiop:localhost:\\$campaign\\_manager\\_port/CampaignManager](#)

26. Make sure that eval\_status = I
27. Make sure that Campaign Budget = cmp\_budget

### Adding Missing Funds to Budget and Check of Overdelivery Compensation

28. Open [Campaign Allocations](#) edit page
29. Add new Campaign Allocation with I/O#1
30. Enter into **Allocation Amount** value equal to **Spent Budget** (from **General Properties** section of [Campaign](#)) + 2 and press **Save**
31. Open [Campaign Credit Allocation](#) edit page
32. Enter into **Amount** value equal to **Spent Budget** (from **General Properties** section of [Campaign](#)) + 3 and press **Save**

### Verification of Results

33. Go to **General Properties** section
34. Make sure that status became **Live**
35. Make sure that:  
**Available Budget = Budget - Spent Budget**
36. Wait a while until changes will not applied to all hosts
37. Open your **SSH** client, login and execute the command:  
  
CampaignAdmin campaign id=310746 -r [corbaloc:iiop:localhost:\\$campaign\\_manager\\_port/CampaignManager](#)
38. Make sure that eval\_status = A
39. Make sure that Campaign Budget = cmp\_budget
40. Make sure that **adServer** successfully processed requests, for this execute any request in any browser from  
stes 5 to 6
41. Generate Uncompleted Invoice

## Test 2.2.2. Check Stats Correction after detection of Fraud Condition

### Preparing Test

1. Open [Campaign](#) view page and remember values of **Budget**, **Spent Budget** and **Available budget**

### Getting Stats

2. Fill in address bar of **IE** browser (**Opted In** user) with adRequest and execute it one time:  
<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&colo=58905&referrer-kw=ouioverdelivery-us-5&loc.name=cn>
3. Execute following Ad Request (as result you should receive 25 banners)  
<http://cs.ocslab.com/cgi-bin/arbitrage.cgi?size=300x250&colo=58905&tid=217797&loc.name=cn&t=fraud1.tmpl&t.n=25>

▼ Getting stats on attest colo

- Run <http://vauto.ocslab.com:8080/testfunctional/> application in any browser
- Select **Host**: [stat-attest.ocslab.com](http://stat-attest.ocslab.com) and **Database**: **stat**



- Use **Login:** test\_ui and **Password:** adserver
- Click **Begin**
- Add one row with parameters:

<b>Campaign Creative ID:</b>	404877
<b>Impressions:</b>	25
<b>adv_amount:</b>	25
<b>Fraud Correction:</b>	<input type="checkbox"/>

- Add one more row with parameters:

<b>Campaign Creative ID:</b>	404877
<b>Impressions:</b>	25
<b>adv_amount:</b>	25
<b>Fraud Correction:</b>	<input checked="" type="checkbox"/>

- Click **Finalize** than click **Execute**
- If your request has been designed properly you should receive popup window with message "Request successfully executed", otherwise you will get error message
- Further you must execute the same verification as on test colo

### Verification of Results

4. Open [Campaign](#) view page
5. Make sure that **Campaign Allocations** contain only one row with Insertion Order
6. Make sure that there is a new **Uncompleted Invoice**
7. Open it and make sure that Total Amount Payable is equal to 0 and **Invoice Data** section contain one row

Unit of Measure	Unit Price	Quantity	Amount
impressions	USD0.00	N	USD0.00

8. Make sure that **Spent Budget** increased on  $\$1 * \text{Quantity}$  and **Available Budget** decreased on the same value

## Test 3. I/O Budget Sharing (Will Be Realised in 3.5)

### Getting Stats

1. Open [Opportunity](#) view page and remember **Spent Amount** value
2. Click **Edit** and set **Amount** as **Spent Amount + 5**
3. Open [Campaign Allocations](#) edit page and add new Campaign Allocation with ID# = [I/O# Sharing](#) and **Allocation Amount = 4**

4. Open [Campaign Allocations](#) edit page and add new Campaign Allocation with ID# = [I/O# Sharing](#) and **Allocation Amount = 4**
5. Wait a while until changes will not applied to all hosts
6. Open your **SSH** client, login and execute the command:

CampaignAdmin campaign id=310943 -r [corbaloc:iiop:localhost:\\$campaign\\_manager\\_port/CampaignManager](#)

7. Make sure that eval\_status = A
8. Make sure that Campaign Budget = cmp\_budget
9. Execute the command:

CampaignAdmin campaign id=310945 -r [corbaloc:iiop:localhost:\\$campaign\\_manager\\_port/CampaignManager](#)

10. Make sure that eval\_status = A
11. Make sure that Campaign Budget = cmp\_budget
12. Close SSH client
13. Remember values of **Budget**, **Spent Budget** and **Available budget** on both campaign view pages
14. Execute the following two adRequests for 4 times each:

<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217797&colo=58905&referer-kw=ouioverdelivery-us-10&loc.name=us>

<http://a.oix-rubyttest.net/services/nslookup?require-debug-info=body&tid=217796&colo=58905&referer-kw=ouioverdelivery-us-12&loc.name=us>

#### ▼ Getting stats on attest colo

- Run <http://vauto.ocslab.com:8080/testfunctional/> application in any browser
- Select **Host:** [stat-attest.ocslab.com](#) and **Database:** [stat](#)
- Use **Login:** test\_ui and **Password:** adserver
- Click **Begin**
- Add one row with parameters:

<b>Campaign Creative ID:</b>	405077
<b>Impressions:</b>	4
<b>adv_amount:</b>	4

- Add one more row with parameters:

<b>Campaign Creative ID:</b>	405079
<b>Impressions:</b>	4
<b>adv_amount:</b>	4

- Actually you can add any number of rows in any sequence of CCID, but total amount of impressions must not exceed 19
- Click **Finalize** then click **Execute**
- If your request has been designed properly you should receive popup window with message "Request

- successfully executed", otherwise you will get error message
- Further you must execute the same verification as on test colo



### Important Note

In some cases the use of this application can lead to incorrect results. This is explained by the fact that data is written directly into memory, bypassing the various checks on adServer and Merger.

## Verification of Results

15. Wait a while to stats appeared in a database



So, Opportunity Balance was \$5, and in two campaigns was allocated Insertion Order of \$4 in each. After execution of requests, all funds from opportunity must be outspent, and campaign credit should be automatically created on missing amount, and then immediately spent. It is unknown, what campaign will spend a credit first, and in what campaign credit will be created. Therefore, two variants are possible and they both will be described further

**First variant (first campaign completely spends funds from I/O - \$4, second campaign spend \$1 and then on the overspent funds, credit is created)**

16. Open [Campaign](#) view page and check status
17. Make sure that it is **Not Live - No Available Budget**
18. Make sure that **Available Budget** now has a value 0
19. Make sure that Campaign Allocations table doesn't contain any row
20. Make sure that there is new Uncompleted Invoice with Total Amount Payable equal to \$4
21. Click [History](#) link in **Campaign Allocation** section
22. Make sure that there is no new record about credit
23. Open [Campaign](#) view page and check status
24. Make sure that it is **Not Live - No Available Budget**
25. Make sure that **Available Budget** now has a value 0
26. Make sure that Campaign Allocations table doesn't contain any row
27. Make sure that there is new Uncompleted Invoice with Total Amount Payable equal to \$1
28. Click [History](#) link in **Campaign Allocation** section
29. Make sure that Details Table have a new record containing row

Start Date	End Date	Type	ID #	Utilised Amount
		Campaign Credit	CC# 1603	USD3.00

30. It means that excess value were credited back to the advertiser
31. Make sure that excess value and value are returned to the advertiser as a credit is equal

**Second variant (first campaign not completely spends funds from I/O - \$3, second campaign spend \$2 and then on the overspent funds, credit is created in each campaign)**

32. Open [Campaign](#) view page and check status
33. Make sure that it is **Not Live - No Available Budget**
34. Make sure that **Available Budget** now has a value 0

35. Make sure that Campaign Allocations table doesn't contain any row
36. Make sure that there is new Uncompleted Invoice with Total Amount Payable equal to \$3
37. Click [History](#) link in **Campaign Allocation** section
38. Make sure that Details Table have a new record containing row

Start Date	End Date	Type	ID #	Utilised Amount
		Campaign Credit	CC# 1603	USD1.00

39. It means that part of the excess value were credited back to the advertiser
40. Open [Campaign](#) view page and check status
41. Make sure that it is **Not Live - No Available Budget**
42. Make sure that **Available Budget** now has a value 0
43. Make sure that Campaign Allocations table doesn't contain any row
44. Make sure that there is new Uncompleted Invoice with Total Amount Payable equal to \$2
45. Click [History](#) link in **Campaign Allocation** section
46. Make sure that Details Table have a new record containing row

Start Date	End Date	Type	ID #	Utilised Amount
		Campaign Credit	CC# 1603	USD2.00

47. It means that the rest of excess value were credited back to the advertiser
48. Open your **SSH** client, login and execute the command:  
 CampaignAdmin campaign id=310943 -r [corbaloc:iiop:localhost:\\$campaign\\_manager\\_port/CampaignManager](#)
49. Make sure that eval\_status = I
50. Make sure that Campaign Budget = cmp\_budget
51. Open your **SSH** client, login and execute the command:  
 CampaignAdmin campaign id=310945 -r [corbaloc:iiop:localhost:\\$campaign\\_manager\\_port/CampaignManager](#)
52. Make sure that eval\_status = I
53. Make sure that Campaign Budget = cmp\_budget
54. Generate Uncompleted Invoice in both campaigns



### Important

If you found out the interesting case which isn't considered in this test plan, please report it to Sergey Gorelov or Mikhail Polyantsev