Integration Guide
# Adobe Test&Target Integration Guide

## Table of Contents

1. Preface ................................................................. 4
   1.1 Intended audience ............................................... 4
   1.2 Additional resources ......................................... 4
   1.3 Useful Firefox Extensions .................................. 4
2. How Test&Target Works .............................................. 5
   2.1 How mboxes works ............................................. 5
3. Steps to Site Set-up .................................................. 7
   3.1 Download and save mbox.js .................................... 8
4. Inserting Mboxes ...................................................... 8
   4.1 Include the mbox.js reference mboxes ..................... 8
   4.2 About default content ....................................... 8
   4.3 Prepare to insert mboxes .................................... 9
   4.4 Insert a single mbox .......................................... 9
   4.5 Tips for mbox insertion ..................................... 9
   4.6 Mbox around a table ......................................... 10
   4.7 Mbox around a table cell .................................... 11
   4.8 Multiple mboxes on a page ................................ 12
   4.9 Whole page as mbox ......................................... 12
   4.10 Dynamic Mboxes ............................................ 12
5. Validating Mboxes .................................................... 14
   5.1 Browse and confirm the mbox is listed in Test&Target  14
   5.2 mboxDebug .................................................... 14
   5.3 Troubleshoot mboxes ....................................... 14
   5.4 Mbox Troubleshooting Guide ................................. 15
   5.5 Troubleshooting resources ................................ 16
6. Using Parameters and Values in Test&Target .................... 17
   6.1 Parameters and values selection matrix .................... 18
7. Reserved mbox values .................................................. 19
   7.1 About the Place Order mbox ................................. 19
   7.2 Place Order mbox syntax ................................... 19
   7.3 Sales Data values ............................................. 19
   7.4 Validate Place Order mbox ................................ 19
   7.5 Passing Page Score .......................................... 20
8. Custom Targeting Groups ............................................ 20
   8.1 Built-In Targeting Groups .................................. 20
   8.2 Creating Custom Targeting Groups ......................... 20
9. Profile Parameters ................................................... 21
   9.1 In-mbox profile parameters ................................ 21
   9.2 In-mbox profile parameter syntax .......................... 21
   9.3 Script profile parameters ................................... 22
   9.4 Script profile parameter syntax ............................ 22
   9.5 Category Affinity ............................................ 23
10. URL Parameters ..................................................... 24
11. Managing Content (Offers) ....................................... 24
   11.1 About offers ................................................ 24
   11.2 Hosting offers ................................................ 25
1 Preface

1.1 Intended audience
This document is intended for page developers, Test&Target users and Test&Target consultants.

1.2 Additional resources
- **Test&Target Help** includes information on how to create offers, choose the right campaign or test, and detailed instructions to meet all of your campaign needs.
  
  Login to Test&Target and click on the Help link in the upper right corner.

- **JavaScript API reference:**
  
  [http://www.mozilla.org/js/scripting/](http://www.mozilla.org/js/scripting/)

- **Document Object Model (DOM) reference:**
  

1.3 Useful Firefox Extensions
- **Right-Click to remove cookies for site**
  

- **FireBug: View generated source with rendered mboxes**
  
2 How Test&Target Works

Mboxes are the connection between your HTML pages and Test&Target. They receive information from your pages and allow you to change the content where they are inserted.

When a visitor browses to a page with an mbox, there are three things the mbox can do:

- Display content
- Receive information
- Record an event

These may be done simultaneously. The mbox may also sit "dormant" on your site until it is added to a campaign in the Test&Target interface.

Mboxes can be placed anywhere on your site. Once they are recognized by Test&Target, Users can choose them to display content, record visits, and/or record conversion activities.

2.1 How mboxes works

Test&Target mboxes are specified using the mboxCreate() JavaScript function defined in the mbox.js function library. The diagram and steps below illustrate how mboxes work with a visitor's browser to communicate with the Test&Target system.

Figure 1 - How mboxes work

1. The visitor navigates to a page that calls the mbox.js function library.
2. The mbox.js library causes the browser to save a first party cookie, called mbox, to the visitor’s system. The cookie is used to uniquely identify the visitor during his or her visit. Since the mbox.js library is installed on your system, the cookie belongs to your domain.

3. The browser continues to create the DOM from the HTML and script code described on the page until it encounters the mboxCreate() function. The mboxCreate() function is designed to accept any number of parameter value pairs, including the name of the mbox and other important information about the visitor’s session.

   The mboxCreate() function gathers these variable values along with the current page URL and referring URL and creates a message to Test&Target called an mboxURL. The mboxURL is sent via an http(s) request to Test&Target.

4. The Test&Target server parses the mboxURL and evaluates the parameters against any campaign using the mbox named in the mboxCreate function. Based on the campaign or test rules set by the users (See Creating Offers and Campaigns section below), Test&Target returns the correct content - or offers- to the mbox. The new content appears seamlessly to the visitor.

   If there is no campaign that uses the mbox, or if the visitor does not meet the targeting conditions, Test&Target does not change the visitor’s display. The visitor continues to see the default content of the mbox, or the normal web content.

5. Test&Target also stores the data provided in the mboxURL string as appropriate to create reports and feed algorithms to optimize your campaign over time.
3 Steps to Site Set-up

Follow the steps below to prepare your web sites to host Test&Target campaigns and tests.

1. Download and save the mbox.js library to one of your web site directories.

2. Once your business strategy and design are complete, insert all mboxes needed to display content, log conversions and/or log pre-conversion visitor activity. This includes inserting a reference to the mbox.js on each page and defining default content for all mboxes.

3. Confirm the mboxes are rendering properly and calls are being made to Test&Target.

4. If you are tracking sales data, targeting audience segments for display, or segmenting reports, ensure all needed parameters are associated with the appropriate mboxes to support your campaigns or tests. This may include inserting Place Order mboxes and mbox parameters.
3.1 Download and save mbox.js

Log into your account and navigate to the mboxes tab>get mbox.js. Download and save the mbox.js to a directory on one of your servers.

Tip
- Ensure the directory and mbox.js file permissions are public.
- If you have multiple domains, consider using a central mbox.js location.

4 Inserting Mboxes

4.1 Include the mbox.js reference mboxes

Add the following source code into all of your pages with mboxes.

```html
<script type="text/javascript">
src="http://www.mycompany.com/myfolder/mbox.js">
</script>
```

Tip
- Add the reference to a global include in the head of your web pages
- Use an absolute or relative URL as best suits your architecture

4.2 About default content

Default content ensures visitors 100% positive experience. Default content displays when:
- no campaigns are assigned to use a given mbox
- a visitor is not targeted by a campaign i.e. the campaign is targeted to visitors coming from Google and a visitor comes from Yahoo
- a visitor's browser has Javascript disabled or does not support JavaScript or cookies
- a visitor's connection is very slow and there is a timeout
- a Test&Target user chooses to use default content as the control in a test

For every mbox you must wrap the default HTML content with the `<div class="mboxDefault"></div>` tags.
4.3 Prepare to insert mboxes

Decide the following based on the strategy and design of your campaign or test:

- The locations, on your website, of the mboxes needed to display content, log conversions, and/or log pre-conversion activities.
- The boundaries of the default content for each mbox. The default content of display mboxes should be a size and shape that supports the offers (creative) designed for the campaign or test.
- The mbox names, which users will recognize when setting up a campaign or test. Mbox names are case sensitive. Test&Target will recognize Mymbox and myMbox as two different mboxes.

4.4 Insert a single mbox

1. Ensure the web page contains a reference to mbox.js.
2. In the body of the HTML page where you wish to insert an mbox, define the beginning and end of the default content with the div tags as shown below. If there is no default content, create an empty div for the mbox.
3. Immediately follow the default content close </div> tag with the mboxCreate function. Give the mbox the unique name users will recognize when creating a campaign or test. In the example below, the mbox is named myMbox.

<table>
<thead>
<tr>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;head&gt;</td>
<td>&lt;head&gt;</td>
</tr>
<tr>
<td>&lt;/head&gt;</td>
<td>&lt;script src=&quot;http://www.mycompany.com/myfolder/mbox.js&quot; type=&quot;text/javascript&quot;&gt;&lt;/script&gt;</td>
</tr>
<tr>
<td>&lt;body&gt;</td>
<td>&lt;/head&gt;</td>
</tr>
<tr>
<td>My existing page.</td>
<td>&lt;body&gt;</td>
</tr>
<tr>
<td>Part that I’d like to overlay during a campaign or test.</td>
<td>My existing page.</td>
</tr>
<tr>
<td>&lt;/body&gt;</td>
<td>&lt;div class=&quot;mboxDefault&quot;&gt;</td>
</tr>
<tr>
<td>Part that I’d like to overlay during a campaign or test.</td>
<td>Part that I’d like to overlay during a campaign or test.</td>
</tr>
<tr>
<td>&lt;/div&gt;</td>
<td>&lt;/div&gt;</td>
</tr>
<tr>
<td>&lt;script type=&quot;text/javascript&quot;&gt;</td>
<td>&lt;script type=&quot;text/javascript&quot;&gt;</td>
</tr>
<tr>
<td>mboxCreate('myMbox');</td>
<td>mboxCreate('myMbox');</td>
</tr>
<tr>
<td>&lt;/script&gt;</td>
<td>&lt;/script&gt;</td>
</tr>
<tr>
<td>&lt;/body&gt;</td>
<td>&lt;/body&gt;</td>
</tr>
</tbody>
</table>

*Figure 2 - Page code for a single display mbox*

4.5 Tips for mbox insertion

- Default tags are case sensitive. Precisely follow the capitalization in the example above.
• Do not wrap individual words as default content. Wrap only paragraphs. By default, DIV tags create new lines in the document, so wrapping entire paragraphs avoids any issues.
• Do not use mbboxes to remove inputs from forms. Mbboxes are only capable of hiding or showing content. They do not remove nodes from the DOM.
• Do not insert any HTML between the close </div> tag and the mboxCreate() function. The mboxCreate() function locates the content it will replace by searching just above itself in the DOM for <div class="mboxDefault">.
• Do not use the page code to place an mbox within an mbox. You may use an offer to insert an mbox within an mbox. Ask your consultant for guidance to do this.
• Do not wrap an mbox around <tr></tr> tags as this will not work.

4.6  Mbox around a table

Wrap default around an entire table. Do not wrap rows.

<table>
<thead>
<tr>
<th>Right</th>
<th>Wrong</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;div class=&quot;mboxDefault&quot;&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;table&gt;</code></td>
<td><code>&lt;table&gt;</code></td>
</tr>
<tr>
<td><code>&lt;tr&gt;</code></td>
<td><code>&lt;div class=&quot;mboxDefault&quot;&gt;</code></td>
</tr>
<tr>
<td><code>&lt;td&gt;peaches&lt;/td&gt;</code></td>
<td><code>&lt;tr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;td&gt;cherries&lt;/td&gt;</code></td>
<td><code>&lt;td&gt;peaches&lt;/td&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/tr&gt;</code></td>
<td><code>&lt;td&gt;cherries&lt;/td&gt;</code></td>
</tr>
<tr>
<td><code>&lt;tr&gt;</code></td>
<td><code>&lt;/tr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;td&gt;walnuts&lt;/td&gt;</code></td>
<td><code>&lt;tr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;td&gt;almonds&lt;/td&gt;</code></td>
<td><code>&lt;td&gt;walnuts&lt;/td&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/tr&gt;</code></td>
<td><code>&lt;td&gt;almonds&lt;/td&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/table&gt;</code></td>
<td><code>&lt;/tr&gt;</code></td>
</tr>
<tr>
<td><code>&lt;/div&gt;</code></td>
<td><code>&lt;/table&gt;</code></td>
</tr>
<tr>
<td><code>&lt;script type=&quot;text/javascript&quot;&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>mboxCreate('myMbox');</code></td>
<td></td>
</tr>
</tbody>
</table>

If attempting to wrap a <tr></tr> section in an mbox, insert an additional table into the code around the <tr></tr> tags i.e.
4.7 Mbox around a table cell

Insert default within the cell, not outside the cell.

<table>
<thead>
<tr>
<th>Before</th>
<th>After</th>
</tr>
</thead>
</table>
| `<table>
  <tr>
    <td>
      CONTENT TO BE MBOXED
    </td>
    <td>
      CONTENT OUTSIDE OF MBOX
    </td>
  </tr>
</table>` | `<table>
  <tr>
    <td>
      CONTENT TO BE MBOXED
    </td>
    <td>
      CONTENT OUTSIDE OF MBOX
    </td>
  </tr>
</table>`

4.7 Mbox around a table cell

Insert default within the cell, not outside the cell.

<table>
<thead>
<tr>
<th>Right</th>
<th>Wrong</th>
</tr>
</thead>
</table>
| `<table>
  <tr>
    <td>
      Peaches
    </td>
  </tr>
<!-- script type="text/javascript" -->
  mboxCreate('myMbox');
  <td>
    cherries
  </td>
  <td>
    walnuts
  </td>
  <td>
    almonds
  </td>
  <td>
  </td>
</table>` | `<table>
  <tr>
    <td>
      peaches
    </td>
  </tr>
<!-- div class="mboxDefault" -->
  <td>
    cherries
  </td>
  <td>
    walnuts
  </td>
  <td>
    almonds
  </td>
  <td>
  </td>
</table>`
4.8  Multiple mboxes on a page

Since each mbox adds some load time to the page, try to limit the number of mboxes on each page to 10.

Follow the instructions for inserting a single mbox. Repeat steps 2 and 3 for each mbox. Only reference the mbox.js file once.

Tip

- Multiple test elements on a page can be combined into a single larger-sized mbox.

4.9  Whole page as mbox

Test&Target can replace all of the content between the <body></body> tags.

```
<html>
<body>
  <div class="mboxDefault">
    ...
  </div>
  <script type="text/javascript">
    mboxCreate('wholePageMbox');
  </script>
</body>
</html>
```

Page code for whole page as mbox

4.10  Dynamic Mboxes

Many Rich Internet Applications (RIAs) manipulate HTML after the page has already loaded by using technologies like DHTML and AJAX. For example, after clicking a button, your web page may display a new section of content. Test&Target supports this scenario, allowing you to define dynamic Mboxes through its mboxDefine() and mboxUpdate() functions.

For example, if you want Test&Target to serve content when an HTML node called "dynamicElement" appears on the page:
<div id="dynamicElement"></div>

then you could trigger the following script on any javascript event:

```javascript
<script type="text/javascript">
  mboxDefine('dynamicElement', 'mbox_dynamic', 'parameter1=value1');
  mboxUpdate('mbox_dynamic', 'parameter1=value1');
</script>
```

Of note:

* `mboxDefine()` defines an HTML element as a container for content to be served by Test&Target. It takes in the unique element id, the Test&Target mbox name, and any number of parameters. The parameters can be used for targeting by the active campaign, even if not passed in again with a later `mboxUpdate()` call.

  `mboxDefine()` does not actually serve content so it should be followed with `mboxUpdate()`.

* `mboxUpdate()` retrieves the content from Test&Target. This function may be called multiple times if you want to further change the content. Like `mboxCreate`, it takes in the mbox name and any number of parameters.

* The usual `mboxCreate()` function only works for HTML elements that exist on the page on the initial load.

* `mboxUpdate()` can also be used for mboxes created with `mboxCreate()` rather than `mboxDefine()`. It allows the page to update content dynamically after the initial page load.
5 Validating Mboxes

Once you have created your mboxes, verify they are communicating with Test&Target.

5.1 Browse and confirm the mbox is listed in Test&Target.

1. Using a browser, login to your Test&Target account. Click on the mboxes tab.
2. All mboxes on your site should appear this list. Expanding each mbox displays all of the known parameters for the mbox.

Tips
• If you do not have access to your company’s Test&Target Account, use the mboxDebug method below to validate the mbox.

5.2 mboxDebug

mboxDebug is a JavaScript pop-up window that helps you troubleshoot your mbox installation using a browser. You may also use it to verify the mbox is recognized by Test&Target.

1. In your browser, navigate to the page containing the mbox. Add mboxDebug=1 to the end of the URL in your browser’s navigation toolbar:

Example 1: http://www.yoursite.com/yourpage.html?mboxDebug=1
Example 2: http://www.yoursite.com/yourpage.html?param=value&mboxDebug=1

2. The pop-up window will list only the mboxes on the page that are communicating with Test&Target.
3. See Test&Target online help for a full description of the information returned in the mboxDebug pop-up.

Tips
• Ensure pop-ups are enabled on your browser.
• Use the close link in the page to keep the pop-up from coming back.
• See separate detailed document about the mboxDebug pop-up window.

5.3 Troubleshoot mboxes

a) If the Debug window does not appear:
• Confirm the mbox.js reference is correct on all web pages with the mboxes.
• Confirm you have downloaded the mbox.js into a folder with public permissions.

b) If the Debug window appears but enabled = false:
• This means that Test&Target was unable to set the “mbox” cookie.
• Delete your cookies and clear cache. Close and reopen a browser and reload the page.
• If enabled=false persists, seek and remove JavaScript errors. Some common errors include:
improper termination of quotes in mbox arguments
- spelling mistakes in your mbox functions
- script tags that are not invoked or that are not closed

- If enable=false persists, contact your consultant.

c) If the mboxes are not listed in the mboxDebug popup window, or if mboxes appear blank on the page, review your page code for the following:

- Confirm the mbox.js reference is correct on all web pages with the mboxes.
- Check that the mboxDefault class and mboxCreate scripts are written correctly.
- Confirm that any tag opened before the mbox script is closed after the mbox script.
- Remove JavaScript errors. See above paragraph for details.
- Scrub layout to support DOM rendering by all browser types. Mboxes insert new nodes into the DOM tree as the browser creates it. Since each brand of browser has its own implementation of the W3C DOM specification, mboxes can affect page rendering differently based on the browser type. Specify absolute sizes of table cells and images to help the browser more accurately display a page’s HTML layout.

5.4 Mbox Troubleshooting Guide
5.5 Troubleshooting resources

The Mozilla Firefox browser includes a JavaScript console that quickly finds and lists the JavaScript errors in your page.

6 Using Parameters and Values in Test&Target

Parameters and values passed during a visitor's session can be used to support these Test&Target features:

- Targeting or displaying unique content (on-site or off-site display ads) to visitor segments.
- Filtering campaign and test reports by visitor segments.
- Reporting sales, Average Order Value (AOV) or Revenue Per Visitor (RPV) for your tests.
- Score per page impression or click. Useful for tracking display ad costs or income from your ad publishing incomes.

Example of an mbox's parameters in Test&Target

Example of target condition set-up in Test&Target
6.1 Parameters and values selection matrix

Use the table below to determine which parameter types you need and where to find information for implementation.

<table>
<thead>
<tr>
<th>Campaign or Test Feature</th>
<th>Parameter Type</th>
<th>How to Implement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target or segment on:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New or returning visitor</td>
<td>profile</td>
<td>See Profiles section of this Integration Guide.</td>
</tr>
<tr>
<td>Values saved in a personalized profile</td>
<td>profile</td>
<td></td>
</tr>
<tr>
<td>URL parameters of page viewed, or referring page.</td>
<td>URL</td>
<td>See URL Parameters section of this Integration Guide.</td>
</tr>
<tr>
<td>Reporting of total sales average order value (AOV) or revenue per visit (RPV)</td>
<td>mbox</td>
<td>Place Order mbox</td>
</tr>
<tr>
<td>Costs per click (Such as Pay-Per-Click)</td>
<td>Mbox or URL</td>
<td>See Page Score in this Integration Guide</td>
</tr>
<tr>
<td>Revenue per impression, page view or click (such as ad publishing revenue)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
7 Reserved mbox values

7.1 About the Place Order mbox

Basic mboxes are sufficient to display offers and to log conversion or success metrics. The Place Order mbox is required in your implementation if you wish to report sales data, Average Order Value (AOV) or Revenue Per Visitor (RPV).

7.2 Place Order mbox syntax

The precise syntax for the mbox name and parameter names must follow the models below. The values in ALL CAPITALS must be passed to Test&Target from your application server.

7.3 Sales Data values

Sales data can be sent to Test&Target from any mbox, but the parameters must follow the strict syntax below. The parameters orderId and orderTotal are required; productPurchasedId is optional. Typically, the mbox name “orderConfirmPage” is used for sales data, but any mbox name is accepted. Multiple mboxes on the site can pass in order data.

```javascript
mboxCreate('orderConfirmPage',
    'productPurchasedId=LIST OF PRODUCT IDS FROM ORDER PAGE',
    'orderId=ORDER ID FROM ORDER PAGE',
    'orderTotal=ORDER TOTAL FROM ORDER PAGE');
</script>

Note: You must pass the orderId parameter value as it allows Test&Target to remove double calls (duplicate orders). The parameter names are also case-sensitive and must be followed as shown above.

7.4 Validate Place Order mbox

Once the reserved mboxes are inserted, verify they are passing values as expected to Test&Target.

1. Browse to a page with the sales data mbox. This will cause Test&Target to recognize the mbox, as well as its parameters and any values passed.
2. In the tool, click on the Mboxes tab.
3. Click the refresh button.

Tips

- See Appendix A for samples of a Place Order mbox integration with PHP and ATG.
4. Scroll down to the orderConfirmPage mbox
5. Press the plus sign to the left of the mbox name. Verify that your parameters and appropriate values are shown.

7.5 Passing Page Score

Costs and revenues or a general score per page, impression or click may be passed and tracked during your campaign or test. You may pass these values via URL or mboxCreate function. These will be summarized in reports.

The reserved parameter name is mboxPageValue. It is case sensitive.

- Insert costs with a negative number: mboxPageValue=-0.1
- Insert revenues with a positive number: mboxPageValue=0.5
- To insert in mboxCreate function: mboxCreate('homePageHero', 'mboxPageValue=0.5');
- To insert in the URL: The link to your page is mydomain.com?mboxPageValue=-0.1

Notes:
- A URL passing a cost or value will override the cost or value passed via an mbox.
- Score may be tracked as low as .00001 (1/100,000 of a cent). However, reports round up to .01.
- Only one value or cost per success metric is logged.

8 Custom Targeting Groups

Targeting groups provide a means for you to reuse commonly created targeting rules. You can use them anywhere you would normally target or segment: in your campaign population selection, in your experience population selection, or in your success metrics. To access targeting groups, go to the "mboxes" tab and then click the "targets" tab.

8.1 Built-In Targeting Groups

There are many built-in targeting groups that are defined here. They are all automatically available in the applicable sections of the targeting interface on the campaign pages. They are listed here to help describe the syntax to make it easier to create new custom ones for your own implementation.

8.2 Creating Custom Targeting Groups

You can also create your own reusable targeting groups. These can be even more powerful using regular expressions and JavaScript expressions.
8.2.1 To Create Expression Targeting Groups:

1. Create a target group by going to "my mboxes" and selecting the "targets" tab. Click the <add expression target> button.

2. Following the same syntax as with script profile parameters, enter JavaScript that returns either true or false.

   Example: Suppose we wanted to target customers on Saturdays and Sundays in order to offer them promotions only available on weekends.

   ```javascript
   return new Date().getDay() == 6 || new Date().getDay() == 0
   ```

   matches
   Saturday OR Sunday (In server time, 0 is Sunday and 6 is Saturday.)

3. Click <save>. You can now use your target in campaign and experience population selection, in segments, or in success metrics.

   Tips:
   - All standard JavaScript operators (== != < > && || etc.) can be used, and there are accessible variables (e.g. landing.page.url) to aid you as well.
   - You may reference both in-mbox and script profile parameters. By defining and collecting profile parameters including things like time-on-site, lifetime monetary value and purchase frequency, you can use them as building blocks in custom target group expressions. See the Test&Target Expression Targets and Script Profile Parameters Cheat Sheet in Online Help for a full list of accessible variables and more examples.

9 Profile Parameters

9.1 In-mbox profile parameters

In-mbox profile parameters are specific to the visitor and are passed to the visitor’s profile from an mbox call on the page. As the visitor continues to browse, or returns for another session, Test&Target can use the saved profile parameter values to personalize display or to better understand visitor preferences.

9.2 In-mbox profile parameter syntax

In-mbox Profile parameters have the prefix "profile." inserted before the parameter names.
A visitor's loan status on a banking site is an example of an in-mbox profile parameter. Confirm with your Consultant how best to utilize this feature in your optimization strategy.

### 9.3 Script profile parameters

Like in-mbox profile parameters, script profile parameters are specific to the visitor and are stored in the visitor's profile, connected by the cookie. As the visitor browses the site, or returns in another session, Test&Target uses the values to target display or to better understand visitor preferences. These profiles are denoted using the `user` prefix within Test&Target.

Unlike in-mbox profile parameters, script profile parameters are defined using Javascript in Test&Target itself, rather than being passed from the page. These scripts are executed on every mbox request and may reference profile parameters, mbox names and parameters, and current page and referring page urls. Test&Target's help section contains a cheatsheet with script profile parameter syntax and examples.

Once stored, these parameters can be referenced in custom targeters or in the targeting section on the campaign create/edit page.

### 9.4 Script profile parameter syntax

Scripts are written in Javascript and may return any type (string, integer, array, etc). Here is the Javascript for a script profile parameter named `user.frequency` that counts how often a consumer buys a product:

```javascript
if (mbox.name == 'orderConfirmPage') {
    return (user.get('frequency') | 0) + 1;
}
```

*A script profile parameter storing purchase frequency*
Notice that the script above referenced itself in order to act as a counter. The orderConfirmPage mbox is on the purchase confirmation page, so that is the signal to this script that the visitor has purchased a product.

Here is the Javascript for a script profile parameter named user.recency that calculates the number of days since a previous purchase:

```javascript
var dayInMs = 3600 * 24 * 1000;
if (mbox.name == 'orderConfirmPage') {
    user.setLocal('lastPurchaseTime', new Date().getTime());
}
var lastPurchaseTime = user.getLocal('lastPurchaseTime');
if (lastPurchaseTime) {
    return ((new Date()).getTime() - lastPurchaseTime)/dayInMs;
}
```

A script profile parameter storing purchase recency

The expression `user.setLocal('lastPurchaseTime', new Date().getTime())` stores the current time in a temporary variable, accessible to this script the next time it is executed for this user. The variable is stored cross-session, so it can be referenced on the next site visit using `user.getLocal('lastPurchaseTime')`.

9.5 Category Affinity

Whenever a user visits your site, Test&Target can automatically record which categories or sections of a site a particular user often browses to. You can record category information by passing it as the mbox parameter "user.categoryId" in any mbox (including a nested mbox), or as a URL parameter "user.categoryId". Please ask your consultant for more details.

Based on the frequency and recency of visits to your categories, Test&Target's algorithm determines what (if any) category affinity a user has. Category affinity can be used to target populations for your campaigns.

Example of Using Category Affinity

* Suppose you have a musical instrument online store and want to target sales promotions on bass guitars to visitors who have browsed the guitar section more than any other section. Using category affinity, you can create offers that display only to visitors with this criterion.

Preparation
To use category affinity, you must first turn on the Category Affinity Smart Targeter.

1. Go to the Mboxes Home Page>Profiles.
2. Press the <START> button next to user.categoryAffinity. The Category Affinity algorithm is now active and user data will be stored.

II. Using Category Affinity for a Targeted group

1. Create any type of campaign or test.
2. Choose ‘target this’ for a campaign or experience.
3. Choose “visitor behavior” and select user.categoryAffinity from the drop down.
4. Select the comparison type (contains, exactly matches, etc.) and enter the categoryId (you may also enter multiple categoryIds if you would like different sets of users to all receive the same content). Users whose top category matches any of these will fall into the population.
5. Click <done>. Your target rule will be summarized on the screen.
6. Continue creating or editing your campaign.

10 URL Parameters

All referring page and current page URL variables are automatically passed to an mbox whenever any visitor views the page with the mbox. These URL variables are then available to use through the for display targeting or segment filters in reports.

To force the parameters to load to Test&Target,
1. Browse to the page with the mbox.
2. Mimic the visitor behavior that causes the desired parameters and values to be passed. This browsing will cause Test&Target to associate the passed parameters and values to the mboxes on that page. You can also simply add the parameters manually to the URL and refresh the page through the browser.
3. View the mboxes in Test&Target. Click on the Mboxes Tab> List.
4. Click the <refresh> button. The URL parameters of the page with the mbox should be listed for that mbox. (NOTE: new parameters can take up to 30 minutes to appear in the list)

11 Managing Content (Offers)

11.1 About offers

Offers is the Test&Target term for any content used in a campaign or test. The user selects offers for display in an mbox. Offers may be Images, HTML, Flash, dynamic content or
anything found on a website.

11.2 Hosting offers

Test&Target allows you to host offers on your server, on Test&Target’s or a third party server.

11.3 Using offers to refer to content on your hosts

Users creating the offers will need to know the locations of the content. They can:

a. Use a relative URL if your web site uses a parallel folder structure across domains. A relative URL may be used if it will be accurate for all environments including development, staging and production.

b. Use an absolute URL if your files are in a centralized location on a separate domain.

12 Working with Multiple Domains

By default your mbox.js sets a new first party cookie and a new session each time a visitor changes domains. If you want the visitor to experience the same experience across multiple domains, and to be counted as the same visitor across domains, there are several methods for doing this. Work with your Consultant to implement a cross-domain campaign or test correctly. Also see Appendix A- About the Test&Target Cookie.
13 Ad Testing

13.1 About Ad Testing

With Test&Target, you can test content off your site using the Test&Target adbox. You can test the ad content using adboxes and test the landing page destinations and track clickthroughs using redirectors. Each implementation is described below.

Using Test&Target adboxes and redirectors, you can achieve the following:
- Monitor ad performance in real time, from first click to final conversion. Compare multiple ads in a test.
- Coordinate ad tests across several ad networks or ad publishers.
- Track view throughs, click throughs and impressions. A view through is when a user does not immediately click on the ad when he sees it, but later arrives at your site by another means.
- Combine ad testing with score tracking to quickly see real-time total revenue for that ad.
- Test and compare various landing locations of the Display ad, for example the home page, the category page and product page.

Note: The adbox and redirector use 3rd party cookies to track a visitor off the site. 3rd party cookies will automatically be created for a visitor when they view an adbox-powered ad. The "1st and 3rd party cookie" setting does NOT have to be enabled in your mbox.js for this work.

Flash Ads
By using adboxes within Flash (.swf) files, you can gather behavioral profile information about the visitors to your ads. This information includes the page they saw the ad on, how many times they saw the ad, and what placement within an ad network buy the ad was shown on. See the CS5 and Test&Target integration to learn more about Flash and Test&Target.

13.2 Ad Test Implementation Matrix

<table>
<thead>
<tr>
<th>Your Goal</th>
<th>How to Implement</th>
</tr>
</thead>
</table>
| Test versions of ad content. Use experiences to track conversion rates for each version. | 1. Create an AdBox.  
2. Create unique ad content and create a Redirect Offers for each content version.  
3. Create a campaign. Choose the AdBox as your display mbox. Create a experience for each content, load the unique Redirect Offers into the Adbox.  
4. Submit the AdBox URL to your Ad Network as the content |
13.3 Create an AdBox

1. Create the AdBox URL

In Test&Target, hover over the Display Ads tab, and click "adbox generator". Choose and enter a name for the adbox (or mbox as it says in the application), and paste the absolute URL of the default content in the next box. For adbox type, choose "Image", and click generate. The URL that is generated is the "adbox URL".

2. Validate the AdBox by copying the AdBox URL and pasting it into a browser and refreshing. Log in to your account, refresh your mbox list and verify the new "ad" mbox is listed (NOTE: it can take up to 30 minutes for the adbox to be listed). You may now submit this URL to your Ad network as the image reference.

13.4 Create a Redirector

Create the Redirector URL

In Test&Target, hover over the Display Ads tab, and click "adbox generator". Choose and enter a name for the redirector (or mbox as it says in the application), and paste the absolute URL of the default content in the next box. For adbox type, choose "Redirector", and click generate. The URL that is generated is the "redirector URL".

Tips:
- During QA the dummy page should have an <a href> link to the Redirector URL.
- Submit the Redirector URL to your Ad Network as the destination page for the image.
13.5 Ad Testing Constraints

Using mboxes inside Flash ads and submitting the .swf to your ad network can overcome many of the constraints below. For more information please contact your consultant.

- A few Ad servers cache content. If they do, you will not be able to track impressions or vary the content with the Adbox; you can only track click-throughs.
- There is no client side timeout as with standard mboxes. If Test&Target is completely down, (which almost never happens), visitors to the ad will not see content, not even default.
- 3rd party cookies are used to track the visits to the ad. If the PCIds are different, by default Test&Target will merge the visitor’s 3rd party with any existing 1st party profiles.
- To use first party cookies on the AdBox itself, you will need to pass the mBox session in the URL. Work with your consultant to implement this.

14 Email Testing

Test&Target can be used to dynamically test images in email, and even change those images on the fly when someone opens their email. By running an optimizing test on images in an email, early responders to your email can influence what delayed email openers see in their email. Redirectors can also be used in emails to track clicks and dynamically control which landing page people reach.

NOTE: A separate detailed integration document is available for email testing.

14.1 Email Image Testing Implementation

Email image testing is achieved through using modified versions of the adboxes described above. Since email clients do not allow cookies to be set, a unique identifier must be generated for each email. This number is appended to the adbox URL and to any redirectors used in the email to track clicks from the email.

Sample code for an email image adbox:

```
```

Where the bold values are specific to you:
clientcode is your company's Test&Target client code. Find this in your mbox.js listed as clientCode='yourclientcode'. This is all lower case and has no special characters.

image is the offer type. It is always "image" for graphic ads, it is "page" for redirects.

email_header is the name of the adbox.

http%3A%2F%2Fwww.domain.com%2Fheader.jpg is the mbox's default content. This must be an absolute reference and must be URL encoded.

mboxXDomain=disabled tells Test&Target to not attempt to set a cookie. Do not change this value.

mboxSessionId=12345 and mboxPC=12345 are 2 values required by Test&Target to merge this user's profile with their existing profile for your site. 12345 is the unique identifier generated per email. Dynamically insert this value into every adbox and redirector URL.

14.2 More information about the unique identifier

The unique identifier per email needs to be assigned to the mboxSessionId and mboxPC in each adbox and redirector URL. The recommended format for this identifier is timestamp-NNNNN where NNNNN is a random 5-digit number, but any alphanumeric format will work. Some mass e-mail services and any programming language are capable of generating this unique identifier.

15 Dynamic Content

Dynamic content can be part of any type of Test&Target test or campaign. You can test different versions of your dynamic messages, forms, interest rate updates, or other content that changes frequently. The visitor's session on your domain is preserved.

15.1 Using styles to test format of dynamic content

There are several ways to use dynamic content in your campaigns and tests. If you plan to test only the appearance of your existing dynamic content, use styles in an offer to hide, show, or reformat existing elements of your dynamic content. Use the Test&Target Help, or contact your Consultant for guidance.

15.2 Remote Offers: Offers Stored Outside Test&Target and Offers Stored on Your Site

If your test involves layout variations of your dynamic content, then offers stored outside Test&Target or offers stored on your site may be the right choice. Use the table below to help you choose the offer best suited for your data update frequency. Consult with your Client Services Manager if you have questions.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Offer Stored Outside Test&amp;Target (cached)</th>
<th>Offer Stored on Your Site (cached)</th>
<th>Offer Stored on Your Site (dynamic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updates each time a visitor makes a request</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Content updates</td>
<td>Cached every 2 hours</td>
<td>Cached every 2 hours</td>
<td>Immediately upon each request</td>
</tr>
<tr>
<td>Load time</td>
<td>Faster</td>
<td>Faster</td>
<td>Slower due to request processing</td>
</tr>
<tr>
<td>Can see JavaScript on page</td>
<td>yes</td>
<td>Yes</td>
<td>No, but can pass via URL</td>
</tr>
<tr>
<td>Offers may include JavaScript</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Restricted to the host that serves the mbox serving the offer</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Offer URL</td>
<td>Absolute</td>
<td>Relative</td>
<td>Relative</td>
</tr>
<tr>
<td>Requesting computer</td>
<td>tt.omtrdc.net</td>
<td>tt.omtrdc.net</td>
<td>The visitor’s computer, which carries the visitor’s cookies.</td>
</tr>
</tbody>
</table>

### 15.2.1 Examples of each offer type:

**Examples of Offers Stored on Your Site (dynamic)**
- * dynamic shopping cart messages served by your server
- * a templatized dynamic product page from your server
- * Anything with many variants or high volume output that cannot be output statically.

**Examples of Offers Stored Outside Test&Target**
- * content from an adserver updated a few times a week
- * content, such as images, served by a third party server
- * form supported by script self-contained within the web page
- * HTML code stored by a third party
- * output with fewer variations, or smaller volume outputs stored on a third party site

**Examples of Offers Stored on Your Site (cached)**
- * content, such as images, served by your site
- * form supported by script self-contained within the web page that you serve from your site
- * HTML code stored on the same site your adboxes are on
- * output with fewer variations, or smaller volume outputs
16 Host Management

16.1 About hosts in Test&Target
Test&Target recognizes any host serving your mboxes. You may use dozens of hosts to support the development, staging and/or production phases of your project.

16.2 About host groups
For ease of management, Test&Target allows you to bundle hosts into host groups. For example, you may use dozens of hosts grouped into two or three host groups.

By default, host groups are pre-named Production, Staging and Development. You may rename these or add host groups as needed. See the Test&Target Help system for details.

16.3 How Test&Target recognizes a host
In order for Test&Target to recognize a host, the following must happen:

- the mbox.js for your account must be saved in a public directory
- at least one mbox must exist on the host
- a page on the host must have
  - an accurate mbox.js reference
  - an mbox
- the page must have been viewed in a browser.

Once the page is viewed the host will be listed in Test&Target, allowing you to manage the host groups as well as preview and launch campaigns and tests. This includes your personal development boxes. Any server with a correct mbox will be recognized.

16.4 Confirm the host is recognized by Test&Target

1. Go to the mboxes tab> manage hosts.
2. If your host is not listed, click the <refresh> button.
3. By default, a newly recognized host is placed in the Production host group. This is the safest host group because it does not allow unapproved campaigns to be viewed from these hosts.
4. If needed, move the host into the Development or Staging host group.

Note: The Production host group cannot be deleted, even if you rename it. Test&Target assumes this is where you will serve final, approved campaigns and tests. The Production host group does not allow unapproved campaigns to be viewed.

Note: A domain will appear in this list if a call to your mbox.js is made from the domain. If someone copies one of your webpages and FTPs to their server, for example, that domain will appear in your host group. You may see domains from spider engines, language translator sites, or local disk drives. If the domain listed is not one your team is using, you can click the “delete” button to remove it.

17 Creating Offers and Campaigns

Once you have inserted and validated the mboxes, Users may select them for displaying offers, or to log visitor behavior. For detailed instructions see the Test&Target Help sections on Offers and Campaign and Test Set-up.

18 Campaign Quality Assurance

18.1 About quality assurance

Before approving new campaigns or tests for public viewing, it is critical to thoroughly test their display, functionality, and reports.

Below are some best practices for validating your campaign. Follow these in conjunction with your standard company quality assurance processes.

18.2 How to assure quality

1. Allow unapproved (pending) campaigns to be viewed from hosts in your Development or Staging host groups. This option cannot be set for the Production Host Group. You may only view approved campaigns from the Production Host Group. See Test&Target Help for instructions.
2. Preview all experiences and test their functionality. Use the Test&Target Help section on Campaign Quality Assurance for instructions, including a shortcut for previewing experiences and forwarding direct links to other approvers.
3. Test on all browser types.

Tip

- To minimize time needed for quality assurance, ensure hosts match in file structure, styles, and templates.
4. If using parameters or values in your campaign, mimic all expected visitor behavior and verify that the expected values are passed to Test&Target. Place orders, arrive from expected source URL and/or browse products, as appropriate for your test or campaign. View the mbox list and click on parameters to verify values are passed as expected.

5. Review the campaign reports to make sure the number of visits, clicks and conversions match the values you would expect based on your preview.
   1. From the Campaigns Home page, click the <Reports> button to the right of the campaign you are testing.
   2. Confirm reports log data as expected for your preview on staging or development hosts. If appropriate, view reports for success metrics, segment filters, optimizing offers and cost/revenue.

6. If the mbox used in your campaign already exist on a host in your Production host group, create a targeting condition that only you can meet. This exclusive condition frees you to approve the campaign, which then permits you complete QA on the hosts in the Default Host Group.

   For example, target the campaign to display only if the URL contains \texttt{testQA=true}.

   Append this to the end of the URL string to allow yourself to see the content. http://www.mydomain.com/mypage.html?\texttt{testQA=true}

   No visitors to hosts in the Production host group will see a campaign targeted in this way, except you, who can satisfy the targeting condition and complete the QA review.

19 Working with Analytics Packages

19.1 About Test&Target data

Test&Target data may vary from data reported by analytics software such as SiteCatalyst. The reasons for this:

Test&Target does not include a visitor in campaign data if the visitor's browser has cookies disabled, or Javascript disabled.

Note. Each visitor's browser may vary in acceptance of cookies and Javascript and cookies. For example, in Internet Explorer the default when Privacy settings are set to high is to disable Javascript, preventing the visitor from seeing the campaign or being counted.
Test&Target does not report on visitors who fail to enter the campaign because they do not meet a targeting condition. Once converted, a returning visitor will be counted as a new visitor in Test&Target reports by default. This allows him or her to convert again, giving you a more accurate picture of a experience's impact on total conversions. This default configuration can be changed within the campaign edit interface.

19.2 Compare Test&Target data with analytical package data

1. It's important to compare Test&Target and analytical package data early in your Integration. Understand differences before launching your campaign or test.

2. Set-up a Monitoring campaign with a single display mbox, the conversion mbox and at least one step in between. Serve default content so there is no change to the visitor's experience of your website.

3. Approve the campaign and run it for a few days.

4. When ready, compare Test&Target reports data and your analytical package data. Any differences should be due to the differences listed above. If there is an inexplicable mismatch, ask your consultant to do further tests to gather more information about the visitors.

20 Test&Target APIs

The following is a description of all public Test&Target APIs. This information is also available on developer.omniture.com and in Test&Target's online help.

20.1 Test&Target Campaign List REST API

20.1.1 Purpose
Extract high level campaign names and corresponding identifiers using a filter (time, state, etc). Typically, one writes a script to iterate through the results and invokes the campaign detail REST API for campaign performance reports.

20.1.2 When to Use
You have campaigns and want to programatically (without the admin interface) extract the ids, typically for reporting purposes.
20.1.3 Implement

1. Create the base URL

Example URL:

https://testandtarget.omniture.com/api?client=acme&email=john@acme.com&password=mysecret

Where:
* acme is your client name
* john@acme.com is your email login for your Test&Target account
* mysecret is your password

Note. Parameters and values are case sensitive.

2. Include parameters to filter the campaign list results

Add the parameters and values you need as a filter.

Example URL:

https://testandtarget.omniture.com/api?client=acme&email=john@acme.com&password=mysecret&operation=campaignList&environment=testEnvironment&name=test&state=saved,activated,library&labels=label1,library,label2&before=2007-01-27T12:00&after=2006-04-01T00:00

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Required or Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>campaignList</td>
<td>Required</td>
</tr>
<tr>
<td>Name</td>
<td>Campaign name (or part of the name) to match on</td>
<td>Optional</td>
</tr>
<tr>
<td>State</td>
<td>Comma separated list of the following states: saved, activated, library</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Default value: All possible states</td>
<td></td>
</tr>
<tr>
<td>Labels</td>
<td>Comma separated list of labels</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Default value: All possible labels</td>
<td></td>
</tr>
<tr>
<td>After</td>
<td>A campaign is included if it was active at least once after the specified date, formatted as yyyy-MM-ddTHH:mm. The time zone is assumed to be that of the browser.</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Default value: 1969-00-00T00:00</td>
<td></td>
</tr>
</tbody>
</table>
Before

A campaign is included if it was active at least once before the specified date, formatted as yyyy-MM-ddTHH:mm. The time zone is assumed to be that of the browser.

Default value: 2100-01-01T00:00

Optional

environment

Any host group defined in Test&Target.

Default value: Production

Optional

3. Inspect the results

A sample success response:

```xml
<campaigns>
  <campaign>
    <id>16</id>
    <name>Test Campaign A</name>
  </campaign>
  <campaign>
    <id>17</id>
    <name>Test Campaign B</name>
  </campaign>
</campaigns>
```

Elements are described below:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Sample Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>campaign</td>
<td>A single campaign</td>
<td>N/A</td>
</tr>
<tr>
<td>id</td>
<td>The campaign id. This can be used for the campaign performance report API</td>
<td>16</td>
</tr>
<tr>
<td>name</td>
<td>The campaign name as you defined it</td>
<td>Test Campaign A</td>
</tr>
</tbody>
</table>

A sample failure response:

```xml
<error>
  <message>
    Invalid email or password supplied. Email: my_email@abc.com
  </message>
  <code>401</code>
</error>
```

Elements are described below:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Sample Value</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Message</th>
<th>A description of the problem</th>
<th>Invalid email or password supplied. Email: <a href="mailto:my_email@abc.com">my_email@abc.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>The error code</td>
<td>401</td>
</tr>
</tbody>
</table>

4. Parse the result and retrieve individual Campaign Performance Reports (optional)

Using a scripting language (Perl, Python, Ruby, etc), iterate through the results and retrieve specific campaign detail reports using the Campaign Performance Report REST API.
20.2 Campaign Performance Report REST API

20.2.1 Purpose

Extract detailed performance information for a single campaign.

20.2.2 When to Use

You have campaigns and want to programmatically (without the admin interface) retrieve the performance reports for off-line analysis or to build your own reporting dashboard interface.

20.2.3 Implement

1. Create the base URL

   Example URL:

   https://testandtarget.omniture.com/api?client=acme&email=john@acme.com&password=mysecret

   Where:

   * acme is your client name
   * john@acme.com is your email login for your Test&Target account
   * mysecret is your password

   Note. Parameters and values are case sensitive.

2. Include parameters to filter the query

   For each operation add the parameters and values you need as a filter.

   Example URL:

   https://testandtarget.omniture.com/api?client=acme&email=john@acme.com&password=mysecret&operation=report&environment=testEnvironment&campaignId=NNNN&start=YYYY-MM-DD&end=YYYY-MM-DD&resolution=hour&segment=name
## Parameter | Value | Required or Optional
--- | --- | ---
Operation | Report | Required
campaignId | The campaign id of the campaign to query | Required
Environment | Any host group defined in Test&Target. Default value: Production | Optional
Resolution | The period of the report, which can be one of the following values: hour, day, week, month. Default value: day | Optional
Segment | The name of the segment. Default value: all segments | Optional
type | The type of metric, which can be one of the following values: visitor, visit, impression. Default value: all types | Optional
Start | From the start of the specified date, formatted as YYYY-MM-DD. The time zone is assumed to be that of the browser. | Optional
End | To the end of the specified date, formatted as YYYY-MM-DD. The time zone is assumed to be that of the browser. | Optional

3. Inspect the results

A sample success response:

```xml
<report campaignId="7" start="2007-08-01" end="2007-08-02" resolution="day" type="visitor">
  <sample start="2007-08-01T00:00" duration="day">
    <recipe name="Recipe A">
      <step name="Entry">3.0</step>
      <step name="Display mboxes">3.0</step>
      <step name="Time on Site">10.0</step>
    </recipe>
    <recipe name="Recipe B">
      <step name="Entry">2.0</step>
      <step name="Display mboxes">2.0</step>
      <step name="Time on Site">12.0</step>
    </recipe>
    <recipe name="Campaign">
      <step name="Entry">5.0</step>
      <step name="Display mboxes">5.0</step>
      <step name="Time on Site">22.0</step>
  </sample>
</report>
```
<recipe>
</recipe>

<sample start="2007-08-02T00:00" duration="day">
  <recipe name="Recipe A">
    <step name="Entry">6.0</step>
    <step name="Display mboxes">4.0</step>
    <step name="Time on Site">15.0</step>
  </recipe>
  <recipe name="Recipe B">
    <step name="Entry">3.0</step>
    <step name="Display mboxes">3.0</step>
    <step name="Time on Site">18.0</step>
  </recipe>
  <recipe name="Campaign">
    <step name="Entry">9.0</step>
    <step name="Display mboxes">7.0</step>
    <step name="Time on Site">33.0</step>
  </recipe>
</sample>

Elements are described below:

<table>
<thead>
<tr>
<th>Element</th>
<th>Attribute</th>
<th>Description</th>
<th>Sample Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report</td>
<td>N/A</td>
<td>Report container. Attributes describe how the campaign report was filtered in the request</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>campaignId</td>
<td>See query parameters</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Start</td>
<td>See query parameters</td>
<td>1975-11-11</td>
</tr>
<tr>
<td></td>
<td>End</td>
<td>See query parameters</td>
<td>2025-11-11</td>
</tr>
<tr>
<td></td>
<td>Resolution</td>
<td>See query parameters</td>
<td>Day</td>
</tr>
<tr>
<td></td>
<td>Type</td>
<td>See query parameters</td>
<td>Visitor</td>
</tr>
<tr>
<td>Sample</td>
<td>N/A</td>
<td>Container for a set of results for the time unit specified in the resolution query parameter</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Start</td>
<td>Start time of the sample</td>
<td>2007-08-01T00:00</td>
</tr>
<tr>
<td></td>
<td>Duration</td>
<td>Duration of the sample</td>
<td>Day</td>
</tr>
<tr>
<td>Recipe (&quot;Experience&quot; in Test&amp;Target)</td>
<td>N/A</td>
<td>Recipe container</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Name</td>
<td>Name of the recipe</td>
<td>Recipe A</td>
</tr>
<tr>
<td></td>
<td>step</td>
<td>Number of visitors,</td>
<td>6.0</td>
</tr>
</tbody>
</table>
visits, or impressions for a step

| name | Name of the type parameter passed as a query parameter | visitor |

Note: The recipe named "Campaign" in each sample shows a summary for that sample.

Error responses:
Error responses will follow the same format as described for the campaign list API. Additional error responses will have the format:

```html
<error>Invalid campaign id (12w) specified</error>
<error>Invalid start date (10 Jun 2007) specified</error>
<error>Invalid segment (from_google) specified</error>
```

4. Parse the results (optional)

Using a scripting language (Perl, Python, Ruby, etc), parse the results and write to output file (like a CSV) that can be displayed in a spreadsheet application like Excel.

20.3 Campaign Audit Report API

20.3.1 Purpose

Extract audit (order) report data for a single campaign.

20.3.2 When to Use

You have campaigns and want to programmatically (without the admin interface) retrieve the audit report data for off-line analysis, permanent storage, or to build your own reporting interface.

Note. Audit data remains stored in Test&Target's databases for only 4 weeks, so if you wish to preserve it, you must retrieve it either with this API or manually with the admin interface.

20.3.3 Implement

1. Create the base URL (follow instructions for campaign list API)

2. Include parameters to filter the query

   For each operation add the parameters and values you need as a filter.
Example URL:
https://testandtarget.omniture.com/api?client=acme&email=john@acme.com&password=mysecret&operation=reports&campaignId=1234&start=2007-08-01T00:00&end=2007-09-01T00:00&step=step1&environment=production&format=csv

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Required or optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>auditReport</td>
<td>Required</td>
</tr>
<tr>
<td>campaignId</td>
<td>The campaign id of the campaign to query</td>
<td>Required</td>
</tr>
<tr>
<td>Format</td>
<td>Csv</td>
<td>Required</td>
</tr>
<tr>
<td>Environment</td>
<td>Any host group defined in Test&amp;Target. It should be URL Encoded. Default value: Production</td>
<td>Optional</td>
</tr>
<tr>
<td>Step (&quot;Success metric&quot; in T&amp;T)</td>
<td>The url encoded step name Default value: the name of the conversion step</td>
<td>Optional</td>
</tr>
<tr>
<td>Start</td>
<td>From the start of the specified date, formatted as yyyy-MM-ddTHH:mm. The time zone is assumed to be that of the browser.</td>
<td>Required</td>
</tr>
<tr>
<td>End</td>
<td>To the end of the specified date, formatted as yyyy-MM-ddTHH:mm. The time zone is assumed to be that of the browser.</td>
<td>Required</td>
</tr>
</tbody>
</table>

3. Inspect the results

The results are in a comma separated format, which can be imported into Excel.

"Campaign: landing campaign RFM"
"Downloaded at: 2007-08-24. Time zone: EST5EDT"
"Step: Conversion"
"Campaign Id","Experience","Order Date","Order Time","Order Id","Amount","Order Flag","Product Ids"
"worst",""
42,81,"2007-08-20","17:25:29","1187645128929",$429.00,---,"9"
42,81,"2007-08-20","17:25:36","1187645135874",$319.00,---,"7"
42,81,"2007-08-20","17:25:43","1187645142834",$325.00,---,"5"
"best",""
42,82,"2007-08-20","16:56:11","1187643370833",$325.00,---,"5"
42,82,"2007-08-20","16:56:16","1187643376266",$479.00,---,"3"
42,82,"2007-08-20","16:58:15","1187643495634",$99.95,---,"42"

4. Parse the results (optional)
Using a scripting language (Perl, Python, Ruby, etc), you can programmatically manipulate the response and construct custom reports.
Appendix A - About the Test&Target Cookie

Test&Target serves a single first party cookie. The cookie has several default settings. These may be changed if needed. Consult your Consultant.

- **cookie name** - mbox.
- **cookie domain** - The second and top levels of the domains *from which you serve the mbox*. Because it is served from your company's domain the cookie is a first party cookie. By default, the domain is set at the main level with no subdomains included. For example: if your website is at "sale.mycompany.com", the mbox cookie will be set on the "mycompany" domain, not "sale.mycompany".

Note. If any of your domain names include a country code, such as *mycompany.co.uk*, work with your Client Services to configure your mbox.js to support this.

- **cookie duration** - The cookie remains on the visitor's browser 2 years from his or her last login.
- **P3P policy** - The cookie is published with your P3P policy, as required by the default setting in most browsers. A P3P policy indicates to a browser who is serving the cookie and how the information will be used.

The cookie keeps a number of values to manage how your visitors experience Test&Target campaigns.

- **session ID** - a unique ID for a user session. By default, this lasts 30 minutes.
- **pc ID** - a semi-permanent ID for a visitor's browser. Lasts until cookies are manually deleted.
- **check** - a simple test value used to determine if a visitor supports cookies. Set each time a visitor requests a page.
- **disable** - set if visitor's load time exceeds the timeout configured in the mbox.js file. By default, this lasts 1 hour.
Appendix B - Mbox.js Advanced Settings

The default settings of the mbox.js function library serve the needs of most Test & Target clients. If needed, consult your Consultant to change or augment the mbox.js settings.

**Traffic level - 100%**
This number sets a limit on the total number of mbox requests to your account, regardless of the campaign. If you set this percentage below 100%, some new (or converted) visitors may be blocked. If someone is already part of a test, she or he will remain in the recipe for every visit during the entire duration of the test or campaign.

**Visitor Exclusion Duration - 120 days.**
Applies only if you set your traffic level to less than 100%. An excluded visitor will be excluded for the visitor exclusion duration. This is called traffic duration in the settings.

**Accepted Browsers**
By default, these browsers are accepted by Test & Target campaigns and tests. You may exclude particular browsers if needed. Please work with your Client Service Manager to do so.

- IE 5.0 or greater (Windows)
- Netscape 5.0 or greater (Mac, Windows, Linux)
- Safari 1.2.4 or greater (Mac)
- Mozilla Firefox 1.0 or greater (Mac, Windows, Linux)

**Mbox.js advanced functions available**
The following functions are available on any page that has mbox.js

mboxGetPageParameter('ParameterName');
* Replace ParameterName with a URL parameter name
* This function will return the parameter value

mboxFactoryDefault.getPCId().getId()
* This function will return the visitors PCID

mboxFactoryDefault.getSessionId().getId()
* This function returns the current sessionID

mboxFactoryDefault.addOnLoad()
* This function allows you to call a function when the page is loaded.
Application Server mbox Integration Examples

The following sample integrations illustrate how to pass server information to Test&Target from a two ecommerce applications - PHP and ATG. These samples are provided for informational purposes only. The code for these sample application server implementations is not included with the Test&Target system, nor are the suggested implementations supported.

**PHP**
The following is a sample using an open source PHP application called osCommerce.

```php
$mbox.js include
<script src="<?php echo (($request_type == 'SSL') ? HTTPS_SERVER : HTTP_SERVER) . DIR_WS_CATALOG; ?>mbox/mbox.js" type="text/javascript"></script>

$orderConfirmPage mbox
<?
    foreach($products_array as $product) $product_ids[] = $product['id'];
    $ids = implode(',, ', $product_ids);
?>
<div class="mboxDefault">
    XX Default Content Here XX
</div>
<script type="text/javascript">
    mboxCreate('orderConfirmPage', 'productPurchasedId=<?=$ids?>');
</script>
```
ATG
The following example uses the ATG JSP tag libraries from the ATG 6.0.0 Pioneer Cycling Demo Store.

```html
orderConfirmPage mbox:
   <div class="mboxDefault">
      XX Default Content Here XX
   </div>
   <script type="text/javascript">
      mboxCreate('orderConfirmPage',
         'productPurchasedId=<dsp:droplet name="ForEach">dsp:param
         bean="ShoppingCart.last.commerceItems"
            name="array"/><dsp:param name="elementName" value="item"/>
         dsp:oparam
            name="output"><dsp:valueof
            param="item.catalogRefId"/></dsp:oparam></dsp:droplet>);
   </script>
```