

OASIS 2.3 User's Manual

Table of Contents

<u>Installation</u>	1
<u>Requirements</u>	1
<u>Setup</u>	1
<u>Post-Install Configuration</u>	7
<u>User Interface</u>	8
<u>Advertisers</u>	8
<u>Sections</u>	11
<u>Campaigns</u>	16
<u>Campaign Insertion</u>	23
<u>Delivery Controls</u>	30
<u>Geographic Targeting</u>	34
<u>Creative Details</u>	35
<u>Rich Media Templates</u>	42
<u>Reports</u>	45
<u>Revenue Report</u>	46
<u>Inventory Report</u>	46
<u>Advertiser Report</u>	46
<u>Campaign Report</u>	46
<u>Creative Report</u>	46
<u>Section Report</u>	46
<u>Report Tables and Formulas</u>	46
<u>Invoicing</u>	48
<u>Admin</u>	51
<u>User Management</u>	52
<u>Publisher Management</u>	55
<u>Preferences</u>	59
<u>Maximum Creative Sizes</u>	59
<u>Traffic Shaping</u>	61
<u>View Admin Log</u>	61
<u>Delivery Engine Internals</u>	62
<u>Reload Delivery Engine</u>	66
<u>Preferences</u>	67
<u>Adding OASIS Ads to Your Site's Pages</u>	71
<u>Method 1: IMG Tagging</u>	71
<u>Cache Busting</u>	71
<u>Instance IDs</u>	72
<u>Specifying the Creative ID</u>	72
<u>Method 2: IFRAMEs and IFRAMEs</u>	72
<u>Method 3: Embedding OASIS Ads</u>	73
<u>Method 4: Remote Javascript Invocation</u>	75
<u>Delivery Engine</u>	76
<u>Impression Script</u>	76
<u>Instance IDs</u>	78
<u>Other uses for oasisi.php</u>	78

Table of Contents

<u>Delivery Engine</u>	
<u>Embedded Impression Code</u>	78
<u>Iframe Impression Code</u>	80
<u>Javascript Impression Code</u>	81
<u>Clickthrough Script</u>	81
<u>Maintenance Scripts</u>	83
<u>Daily Maintenance</u>	83
<u>Check for underdelivery</u>	83
<u>Compute daily targets</u>	83
<u>Load shared memory</u>	83
<u>Hourly Maintenance</u>	84
<u>Step-by-step through hourly_maint.php</u>	84
<u>Adding hourly_maint.php to the crontab</u>	86
<u>Calling hourly_maint.php from the command-line</u>	86
<u>Minutely Maintenance</u>	88
<u>Adding minutely_maint.php to the crontab</u>	88
<u>Log files</u>	89
<u>Delivery Engine Logs</u>	89
<u>Administrative Logs</u>	89
<u>Shared Memory</u>	90
<u>Hourly Assignments</u>	90
<u>Hourly Targets</u>	90
<u>Creative Content</u>	91
<u>Creative Clickthrough</u>	92
<u>Deliveries</u>	92
<u>Overflow flag</u>	92
<u>Database Schema</u>	93
<u>Publishers Table</u>	93
<u>Advertisers Table</u>	94
<u>Campaigns Table</u>	95
<u>Creatives Table</u>	98
<u>Sections Table</u>	100
<u>CampaignAssignments Table</u>	101
<u>CreativesTemplates Table</u>	102
<u>CreativeAssignments Table</u>	103
<u>CompanionGroups Table</u>	104
<u>CompanionSections Table</u>	105
<u>DailyTargets Table</u>	106
<u>DeliveryControls Table</u>	107
<u>RunningCreatives Table</u>	108
<u>Invoices Table</u>	109
<u>PublisherInvoices Table</u>	110
<u>HourlyStats Table</u>	111
<u>CampaignDailyStats Table</u>	112

Table of Contents

Database Schema

<u>Delivery Table</u>	113
<u>SectionDailyTraffic Table</u>	114
<u>Users Table</u>	115
<u>CampaignAccess Table</u>	116
<u>TrafficProfile Table</u>	117
<u>MaxSizes Table</u>	118
<u>OASISPrefs Table</u>	119

Tips.....120

<u>Watch your user</u>	120
<u>Be careful with DaysOfWeek and HoursOfDay</u>	120
<u>Linking directly to creatives</u>	120
<u>About clickthrough errors</u>	121
<u>More about clickthrough errors</u>	121
<u>Last hour of the day</u>	121
<u>Campaign/Creative Weights</u>	122
<u>Importing Campaigns in Progress</u>	123
<u>Be careful with Exclusives</u>	123
<u>Stopping MySQL</u>	124

Troubleshooting.....125

Appendix: Delivery Table Design History.....128

Installation

Requirements

OASIS has been tested only with RHEL 4.

With that said, there are specific software requirements for OASIS. If you can meet these on another platform, you should be able to run OASIS. The installer has been tested with RHEL 4, but it is quite likely that it will work with other Linux and BSD variants, as long as the database requirements are met.

The specific requirements are these:

- Apache 2.0 in prefork mode
- PHP 4.3 (both Apache module and command-line PHP interpreter); make sure your PHP build includes the following modules:
 - ◆ curl
 - ◆ domxml
 - ◆ mysql
 - ◆ sysvshm
 - ◆ sysvsem
- MySQL 4.1 (<http://www.mysql.com/>)
- ImageMagick (<http://www.simplesystems.org/ImageMagick/>) with support for JPEG and PNG (version 6.0 and up)
- curl (<http://curl.haxx.se/>) version 7.12 and up
- htmldoc (<http://www.easysw.com/htmldoc/>) version 1.8.8 and up

Setup

System Configuration

An ad server requires an accurate clock. It is recommended that for all versions of OASIS, you set up NTP on your system to ensure accuracy of the local clock. OASIS Enterprise **requires** that the clocks be synchronized between master and slave.

Make sure that the `ntpd` service is running and is properly configured on your server(s).

SELinux

If your system is using SELinux, you need to modify the local policy. The following instructions are for RHEL 4. You may need to modify the procedure for other Linux variants.

1. Install the policy source

```
up2date selinux-policy-targeted-sources
```

2. Edit the sources in `/etc/selinux/targeted/src/policy/domains/misc/local.te`

```
allow httpd_t tmpfs_t:file { read write };
allow httpd_t httpd_log_t:file { rename unlink };
allow httpd_t var_log_t:file { append getattr read write };
allow httpd_t var_log_t:dir remove_name;
allow httpd_t unconfined_t:shm { associate read unix_read unix_write write };
```

OASIS 2.3 User's Manual

```
allow httpd_t unconfined_t:sem { associate read unix_read unix_write write };
```

3. Load the policy:

```
cd /etc/selinux/targeted/src/policy
make load
```

Apache Account Configuration

Your Apache account will not have privileges to run the cron jobs. You need to give it a real shell. Edit `/etc/passwd` to change the apache user's shell from `/sbin/nologin` to `/bin/bash`. Standard warnings about editing `/etc/passwd` apply.

PHP Configuration

If your command-line PHP interpreter is not at `/usr/bin/php`, make a symbolic link. All of the scripts assume that this is the location of the PHP interpreter.

Make sure that you have a `php.ini` file and that it contains the following directives:

```
error_reporting = E_ALL & ~E_NOTICE
register_globals = On
always_populate_raw_post_data = On
memory_limit = 64M
register_argc_argv = On
track_errors = On
```

Note that some of these configuration directives are already in your `php.ini` file. Don't just blindly add the above lines to your file; change the existing directives to match the ones above.

Special note about `register_globals`: some administrators are uncomfortable with having `register_globals` on. You can use `register_globals` and not compromise your entire site by just enabling it for your OASIS directory. The following lines of `httpd.conf` can serve as an example:

```
<Directory /path/to/oasis>
php_flag register_globals on
</Directory>
```

Apache Configuration

1. Make sure Apache is configured for PHP.
2. Include the `oasis_httpd.conf` file that will be created during the installation process. To do so, add the following to `/usr/local/apache/conf/httpd.conf`:

```
Include conf/oasis_httpd.conf
```

Note that the path to `oasis_httpd.conf` is relative to the Apache `ServerRoot`. Even though this file doesn't exist yet, it's OK to change this; just don't try to restart Apache until you've run the OASIS installer.

3. Make sure that `LimitRequestBody` is set to 0. Under RedHat Linux, this directive is set in `/etc/httpd/conf.d/php.conf`. Change the value and save the file.

OASIS Configuration/Installation

1. Once you have PHP and all the other software above installed, untar the OASIS distribution in a temporary location.

OASIS 2.3 User's Manual

2. Edit the file `oasis.cfg` to set up some initial variables. This file is self-documenting, so you should be able to figure out from the comments how to configure it.
3. Run `install.php` as root. When prompted for a MySQL username and password, give a name and password of a user privileged enough to add a new user to the MySQL privileges tables, create the OASIS database, and reload the MySQL server (probably root, which may have an empty password for MySQL).

If you notice any errors in the last couple steps (where `mysqladmin` is called to reload, and `hourly_maint.php` is called), you'll have to manually run those commands. MySQL *must* be reloaded in order to allow the OASIS scripts to connect to the database. `hourly_maint.php` must be run to load up shared memory for OASIS.

If you see warnings about shared memory or database connections, then you are going to have to run `hourly_maint.php` by hand. If you have to run `hourly_maint.php` by hand, run it like this:

```
su - apache -c '/path/to/oasis/mgmt/hourly_maint.php start'
```

be sure to replace "apache" with the web_user you defined in `oasis.cfg`!

4. Create Directories

1. When you install Oasis, it will create an images directory for you in the mgmt directory, but you need to define the images directory permissions manually.
2. The images directory should have ownership as the web server user normally nobody or apache and permissions `rwxr-xr-x (755)` is preferable.
3. Within the images directory, create a new directory for each existing creative in the Creatives database table with the directory name being equal to the CreativeID number value. For an brand new Oasis installation, you should make a directory called "1" to be used for the out-of-the-box text creative with ID equal to 1. The images directories should have ownership as the web server user normally nobody or apache and permissions `rwxr-xr-x (755)` is preferable.

5. Download and install swfobject.js:

1. Download the latest version of `swfobject.js` and place it in the OASIS images directory.
2. `swfobject.js` can be downloaded from <http://blog.deconcept.com/swfobject/>

6. Now restart Apache, and you can log into the Web interface.

Enterprise Server Notes

Note that OASIS Enterprise Server does not use `hourly_maint.php`. Instead, it uses `master_reload.php` on the master server.

Note that before you run `master_reload.php` on the master server, you must do two things:

1. Install OASIS on the slave servers (follow the exact same procedure used to install the master; use the same `oasis.cfg` on each).
2. Restart Apache.
3. Go to the Web interface on the master server; click on "Admin", then "Preferences", and enter your SlaveServers (as a comma-separated list). Save your settings.

Now you are ready to run `master_reload.php`:

```
su - apache -c '/path/to/oasis/mgmt/enterprise/master_reload.php start'
```

7. Image Sync

Enterprise Server Notes

Image Sync :

If you are using the OASIS Enterprise master/slave configuration, you need to synchronize the images directory between the Master and the Slave Servers. The image synchronization is done in the creative.php file. By default the code for image sync is commented out. If you are not going to use a master & slave cluster then you can skip these steps.

In order to achieve the automatic image synchronization we have to generate SSH keys for password less authentication. Follow these steps to create SSH keys.

STEP 1:

If you are going to run OASIS as the default webuser apache or nobody or daemon (see httpd.conf file to get this information) rather than a real user (virtual hosting) then proceed with STEP 1 else please skip this and proceed with STEP 2.

Login as root in the MASTER server and edit the /etc/passwd file (Standard warnings about editing /etc/passwd apply.) and give the webserver user a real home directory (for example)

```
nobody:x:99:99:Nobody:/home/nobody:/bin/bash
```

STEP 2:

Login to the Master server as OASIS user (webserver user/ real user) and give the following commands..

```
$ ssh-keygen -t rsa Assign the pass phrase (press [enter] key twice if you don't want a passphrase).
It will create 2 files in ~/.ssh directory as follows:
```

```
~/.ssh/id_rsa : identification (private) key
```

```
~/.ssh/id_rsa.pub : public key
```

Use scp to copy the id_rsa.pub (public key) to the slave server (s) as authorized_keys2 file, this is known as Installing the public key to server.

```
$ scp ~/.ssh/id_rsa.pub user@www.slave1.com:~/.ssh/authorized_keys2
```

Where user is the webserver user/real user from which you run the OASIS. **NOTE :** If you have more than one slave copy repeat the above step and copy to all the slaves. iii) If everything went fine then you can login in the slave(s) without password authentication. \$ ssh user@www.slave1.com If not please check the system logs of both master and slave to see what went wrong. **STEP 3:** Open the creative.php file in Master and look for these lines. These lines will be commented out by default. Uncomment it.

```
##### IMAGE SYNC TO SLAVE#####
```

```
$DEST="user@www.slave1.com";
```

```
$DDIR="/home/oasisacc/oasis/images/"; //Path of Images directory in Slave
```

OASIS 2.3 User's Manual

```
$LDIR="/home/oasisacc/oasis/images/"; //Path of Images directory in Master

$SSH="/usr/bin/ssh";

$ret=shell_exec("rsync -avrp --links $LDIR --rsh=$SSH $DEST:$DDIR");

if(!$ret){

$to = "to@domain.com"; //mail-id to whom the error message should be sent

$from = "from@domain.com"; //mail-id from whom the error message should be sent

$subject = "Image copying failed"; //subject of the mail

$message_body = "Image copying process failed, please check."; //Message of the mail

$headers = "MIME-Version: 1.0\r\n";

$headers .= "Content-type: text/html; charset=iso-8859-1\r\n";

$headers = "From: ". $from. "\r\n";

mail($to,$subject,$message_body,$headers); //Here mail sends

and change it to according you needs.
```

NOTE: If you are going to use more then one slave then copy paste this code for each slave and change accordingly.

OASIS Tests

1. Open the OASIS interface, which is found at "http://[oasis_host][oasis_url]mgmt/" (for example, if oasis_host = oasis.yourdomain.com and oasis_url = /oasis/, you'll go to <http://oasis.yourdomain.com/oasis/mgmt/> to open the management interface). Log in with the username and password you specified in `oasis.cfg`.
2. Create a top-level section for your ad server (for example, just call it "Main").
3. Create an advertiser and a campaign (probably easiest to use the Campaign Insertion interface under "New Campaign"). Go into the Campaign interface to make sure the campaign is set to "Active". Also, make sure that it is assigned to the top-level section you created, and make sure that its dates cover the current day.

Enterprise Server Notes

Make sure you have specified one or more slave servers on the Admin->Preferences page. This is a comma-separated list of hostnames or IP addresses. You may include the master server as one of the slaves.

4. On the Admin page, reload the delivery engine.
5. Now go to the Section interface, get the sample URL for your top-level section (just click on the section's name), and then open the URL for the IMG SRC. You should see your first banner served up.

OASIS 2.3 User's Manual

If you don't see a banner, make sure that you see something assigned to the section under Hourly Assignments (under Delivery Engine Internals on the Admin page). Also make sure that the width and height specified in the URL match those for the ad(s) you put into your first campaign. If you don't see any assignments for Section 1, then most likely your campaign is not Active, it wasn't assigned to Section 1, or its dates do not cover the current day.

System Configuration

1. Once you're getting banners, you need to set up cron jobs (`hourly_maint.php` and `minutely_maint.php`) in the crontab belonging to the `web_user` you defined in `oasis.cfg`:

```
0 * * * * /path/to/php /path/to/oasis/mgmt/hourly_maint.php > /dev/null
5,10,15,20,25,30,35,40,45,50,55 * * * * /path/to/php /path/to/oasis/mgmt/minutely_maint.php > /dev/null
```

Enterprise Server Notes

OASIS Enterprise Server does not use `hourly_maint.php`. The crontab on the master server should look like this:

```
0 * * * * /path/to/php /path/to/oasis/mgmt/enterprise/master_reload.php > /dev/null
5,10,15,20,25,30,35,40,45,50,55 * * * * /path/to/php /path/to/oasis/mgmt/minutely_maint.php > /dev/null
```

On the slaves, it should look like this:

```
5,10,15,20,25,30,35,40,45,50,55 * * * * /path/to/php /path/to/oasis/mgmt/minutely_maint.php > /dev/null
```

2. Set up OASIS to start upon a system boot. Add these lines to `/etc/rc.d/rc.local`:

```
su - apache -c '/path/to/php /path/to/oasis/mgmt/hourly_maint.php start'
```

Of course you need to substitute your own paths and username in these lines! Ideally, you will start OASIS before you start Apache (otherwise, some users may hit the server before you have loaded shared memory).

Enterprise Server Notes

OASIS Enterprise Server does not use `hourly_maint.php`. Put this in your `rc.local` instead:

```
su - apache -c '/path/to/php /path/to/oasis/mgmt/enterprise/master_reload.php start'
```

3. Finally, make sure that MySQL and Apache are starting at system boot. Under RedHat Linux, you can use `chkconfig` to control what starts up at the various runlevels.

Post-Install Configuration

Many parameters that control the behavior of OASIS are stored in the MySQL database and are accessible via the [Admin interface](#).

However, parameters that affect real-time delivery are not stored in the database, as retrieval of these values would be too expensive. Instead, they are stored in server variables in an Apache configuration file, `oasis_httpd.conf`.

The values in this file are created when the software is installed. They come from the `oasis.cfg` file which you edited before installing the software (you did edit it before you installed, right?) To understand the meaning of the values, see the comments in `oasis.cfg`.

If you need to change any of these values, edit `/etc/httpd/conf/oasis_httpd.conf`, and then restart Apache!

User Interface

OASIS is fully manageable via a PHP web interface. All aspects of a publisher's ad operations can be managed using the OASIS interface: campaign and creative creation, advertiser management, reporting, inventory (section) management, and invoicing, using any web browser.

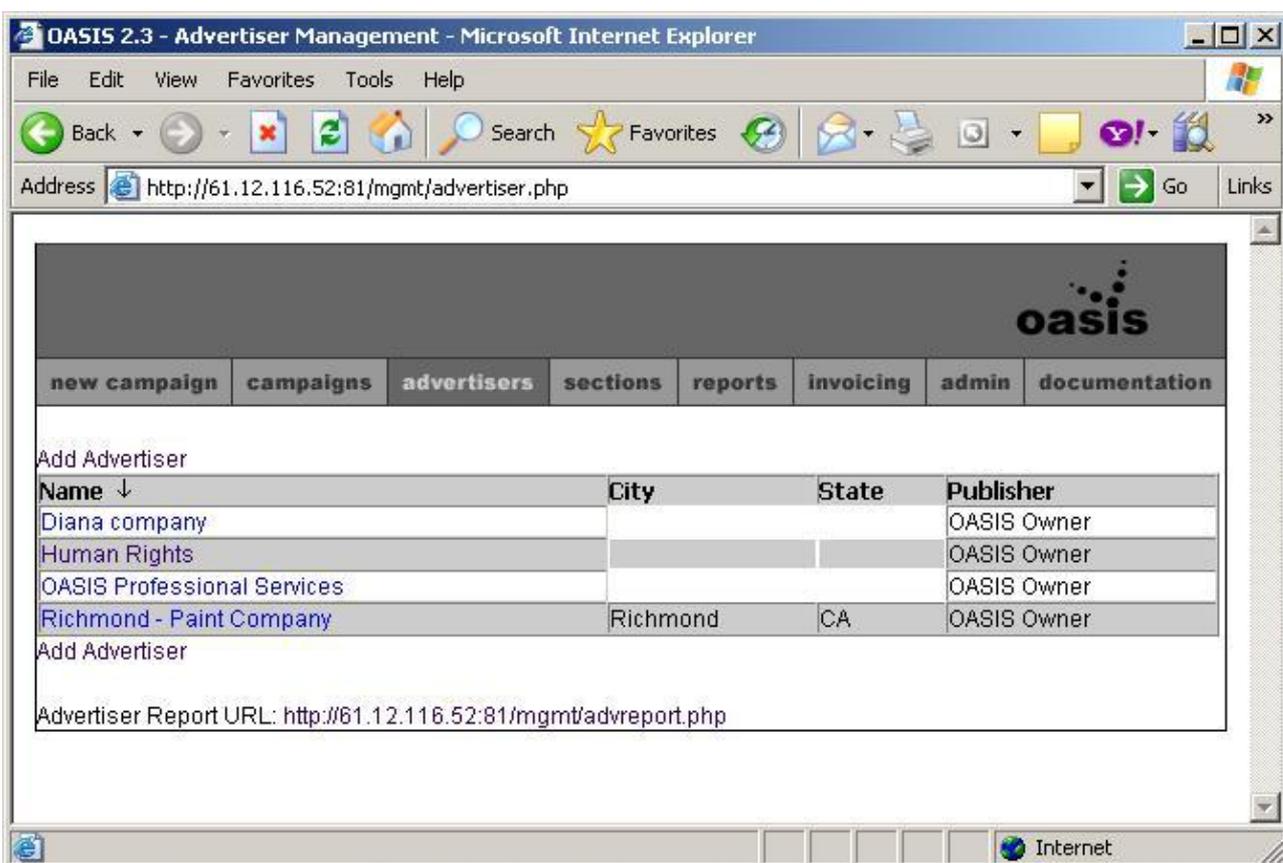
The OASIS user interface consists of eight main areas, each of which is accessed using the tab links at the top of the page. When the publisher advertising administrator creates users for the OASIS system, he can also assign privileges to those users which define which of the following tabs a user can access:

- New Campaign - This is a tool or "wizard" which walks a user through the steps in setting up a campaign. Typically used for sales purposes, the administrator can give a salesperson privileges to access the New Campaign and Reports tab only, so that sales can only create new campaigns and run reports but cannot edit campaigns once they have been created.
- Campaigns - The campaigns tab lists all of the campaigns that belong to a publisher and any of that publisher's sub-publishers. Campaigns control how an advertising order will be delivered and which creatives will be associated with an ad order. A new campaign can be created from the campaigns tab without requiring the use of the New Campaign wizard.
- Advertisers - The advertisers tab lists all of a publisher's advertisers and advertiser details.
- Sections - The sections tab is used to organize a publisher's "inventory", which describes the layout and structure of the publisher's sellable categories, sites, pages and positions.
- Reports - The reports tab allows a publisher to create various reports including revenue, inventory forecast, campaign and section reports.
- Invoicing - The invoicing tab lets a publisher create, manage and print invoices for both advertiser and sub-publisher clients.
- Admin - The admin tab allows a publisher to create and manage users and sub-publishers.
- Documentation

[Documentation Home](#)

Advertisers

The advertiser interface lets you enter information about the companies that are buying advertising on your site. This information is needed for assigning ad campaigns, giving out advertiser reporting logins and generating invoices.



In the main advertiser list screen as shown above, you see a list of all of the advertisers in the database. Click on the Name arrow to reverse the sort order. Click on any column header to sort by that column.

The "Advertiser Report URL" is the URL you provide to advertisers who wish to have direct access to advertiser reports. Advertisers with access to reporting will only be able to generate campaign reports for campaigns which belong to their advertiser. Providing advertisers with access to reporting is optional. In order to give an advertiser reporting access, you must define a Login and Password on the Advertiser Details page, as shown below.

Click "Add Advertiser" to create a new advertiser. To edit an advertiser, click on the advertiser's name. Shown below is a sample advertiser details page.

OASIS 2.3 - Advertiser Management - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites Print Mail Stop

Address <http://61.12.116.52:81/mgmt/advertiser.php?AdvertiserID=3> Go Links

oasis

new campaign campaigns **advertisers** sections reports invoicing admin documentation

Advertiser Details

Name

URL

Contact

Street Address

City

State

Postal Code

Country

Phone

Fax

Contact E-mail

Login

Password

Note

Done Internet

Except for the advertiser Name, none of the information here is mandatory, but you will not be able to generate useful invoices if you do not enter the complete address information.

[Documentation Home](#)

Sections

OASIS uses a hierarchical tree of "Sections" to manage which ad banners get served on which pages. Each campaign in the system is assigned to one or more sections. The campaign will then run on that section and all sections beneath it in the tree.

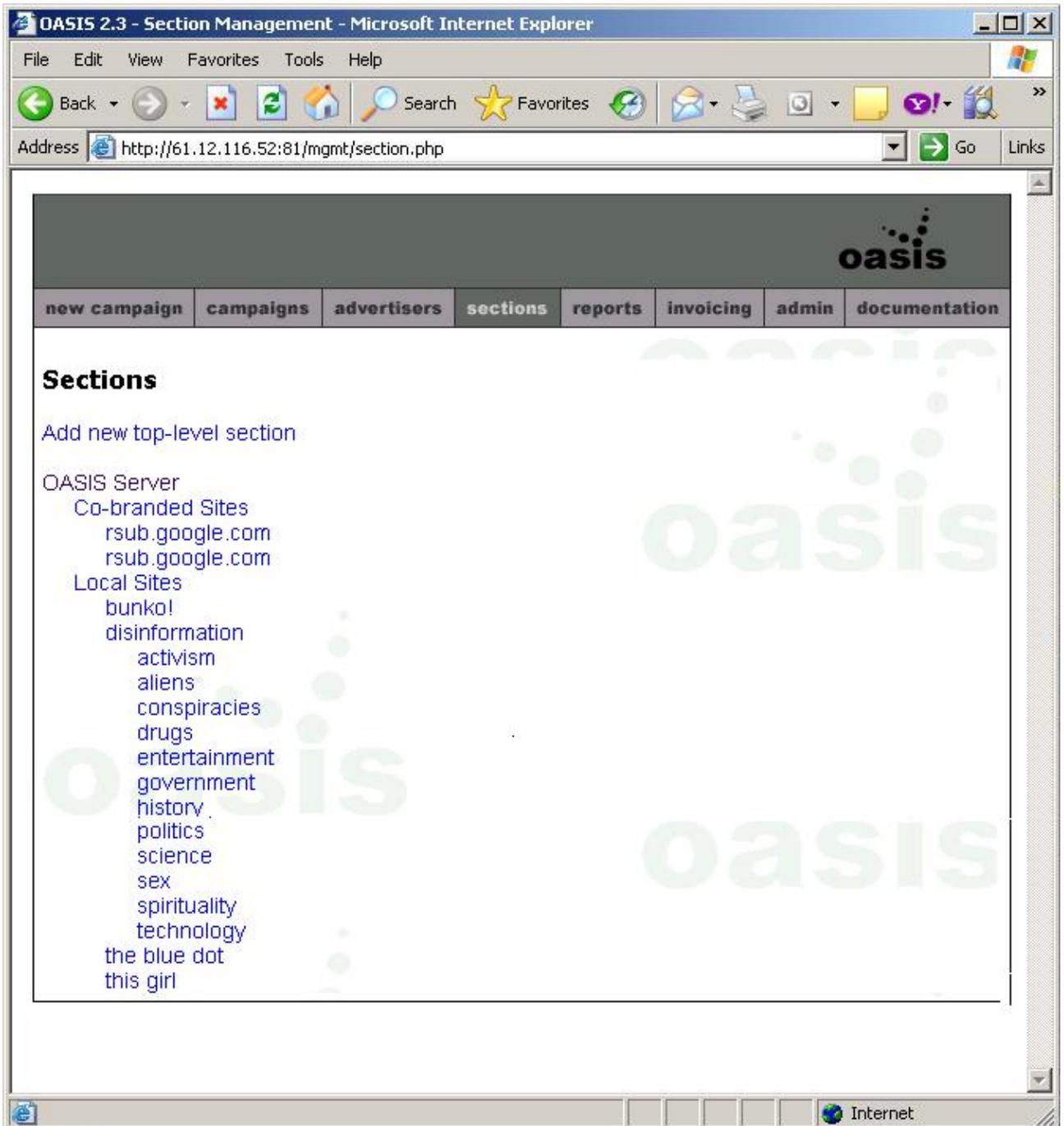
Additional power and flexibility comes from the ability to *exclude* a campaign from a section. So with the tree shown below, you could easily run a campaign on all the subsections of `disinformation` except `activism` by including the campaign on `disinformation` and excluding it from `activism`.

In addition, you can assign a campaign to a section as an *exclusive*, so that no other campaigns can run on that section. Care must be taken when using these exclusive assignments. Imagine if you tried to assign campaign B to `activism` exclusively, but you had already assigned campaign A to `disinformation` exclusively!

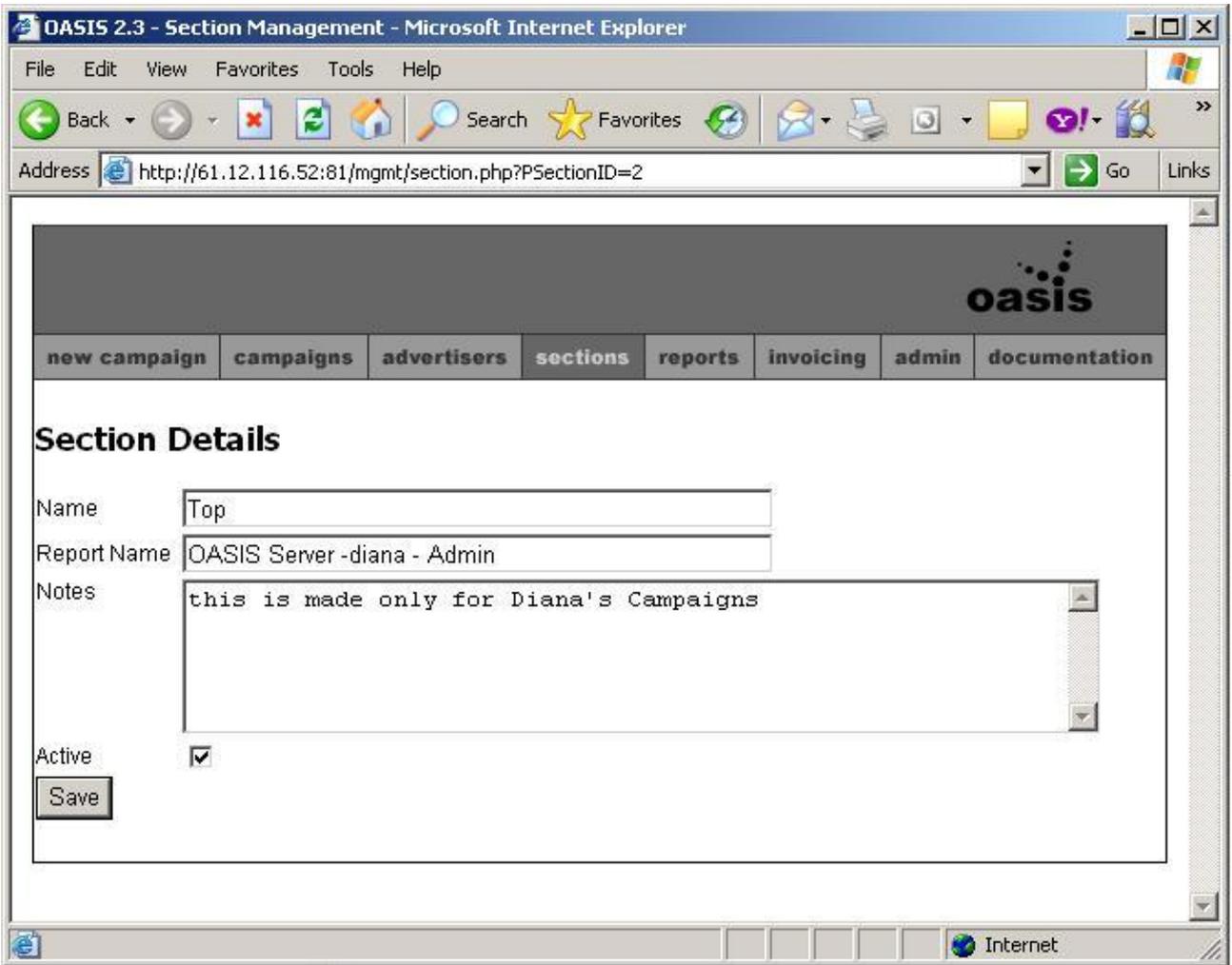
To further complicate matters, creatives can be assigned to sections independently of their parent campaigns. All conflicting creative assignments override those of the parent campaign.

Each hour, the maintenance script assigns creatives to sections, propagating these assignments down the section tree. They are processed in this order:

1. Campaign Includes
2. Creative Includes
3. Campaign Excludes
4. Creative Excludes
5. Campaign Exclusives
6. Creative Exclusives



The main Section management screen displays the full section tree. Click "Add a new top-level section" to add a new section with no parents (ie., start a new tree). Click on any of the section names to edit or delete them or to add subsections.



On the section edit form, you see the two names given to the section (one is a shorter name, used in hierarchical displays of the section tree), and the other is used on reports, where you don't have the context of knowing what the parent section is.

You can also mark a section as active or inactive by checking or unchecking the checkbox next to "Active". If a section is marked inactive, no creatives will be assigned to it or any of its children. Also, you won't get error reports if the inventory checker finds that no creatives are assigned to inactive sections. If all your campaigns for a particular section complete, and you don't intend to run any more on it for a while, you might want to mark it as inactive.

Use the Notes field to enter notes about the section. Information in this field is purely for your convenience; it has no effect on the operation of the ad server.

Hit "Save" to save changes you make to the names. Hit "Delete" to delete the section (and all subsections). You will be prompted to confirm this decision.

The entire tree below the current section is displayed. To add a new subsection to the current section, click "Add new subsection". To edit or delete a subsection, click on its name

You can generate sample ad tags. Select the tag type (based on what you're using on your site). Select the dimension of the creative you want to display. Then press "View tag", and a new window should pop up with

the code ready to copy and paste into your pages.

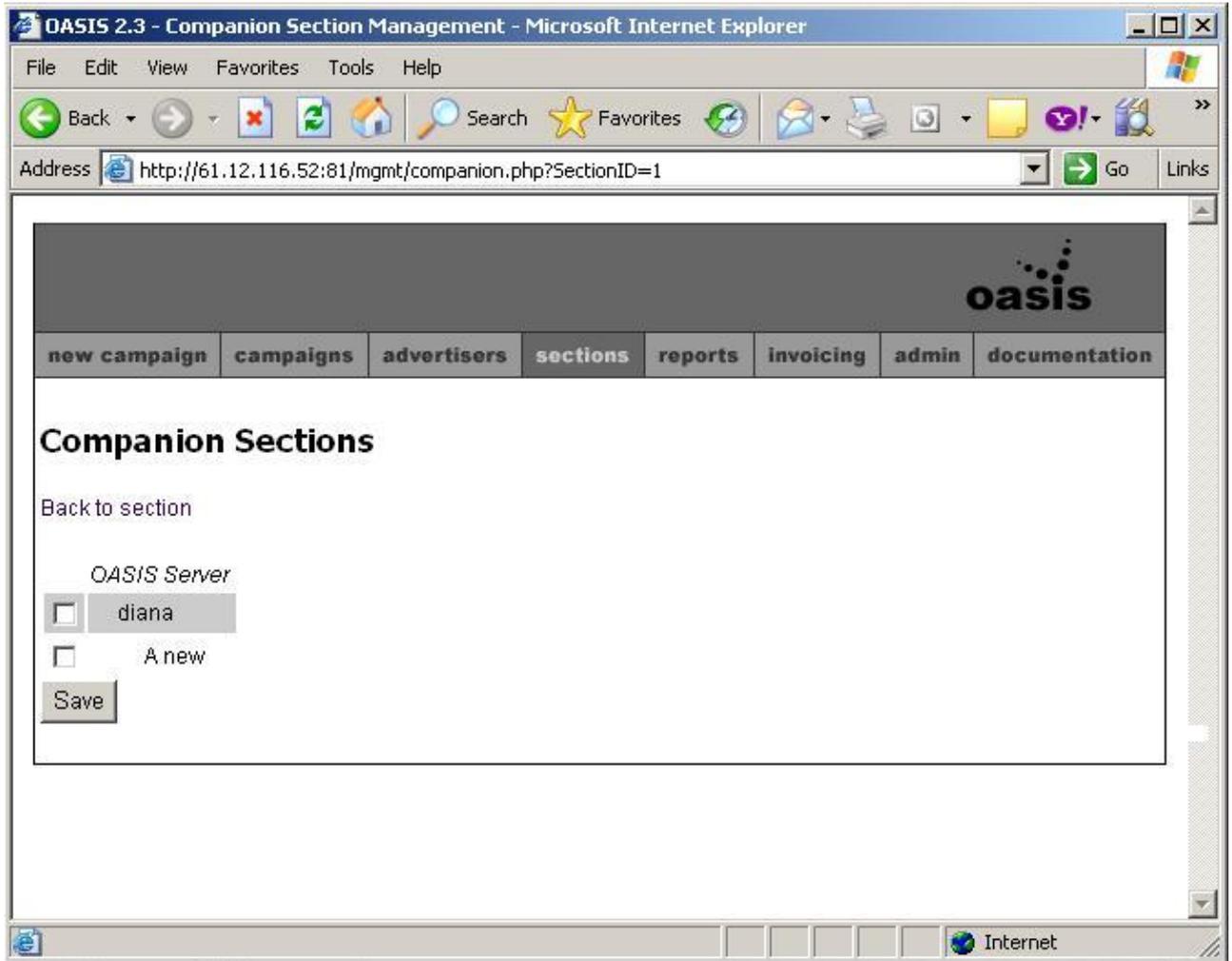
Finally, if there are campaigns or creatives assigned to this section, they appear in tables on this page. These lists are only those campaigns and creatives assigned explicitly to the current section, not those assigned to parent sections.

Companion Sections

If you run two creatives on a single page (for example, top and bottom positions), you may want to give each its own section so that you can track the effectiveness of the two positions. You may want to insure that you don't get the same creative in both positions at the same time. Companion sections provide the mechanism for this.

If two sections are designated companion sections, they will not ever run the same creative at the same time. In fact, they generally will not run two creatives from the same campaign at the same time, unless the campaign has the "Companion sharing" option set.

You can designate any arbitrary number of sections as companions with the companion section interface:



Simply check the boxes next to the sections in the companion group and click "Save". To remove the companion group, you can uncheck these boxes and click "Save".

[Documentation Home](#)

Campaigns

The main campaign page lists all of the campaigns that belong to a publisher and any of that publisher's sub-publishers. Campaigns control how an advertising order will be delivered and which creatives will be associated with an ad order. In the image below, you see a list of all campaigns in the database. Click on the arrow to reverse the sort order. Click on any column header to sort by that column.

new campaign campaigns advertisers sections reports invoicing admin documentation

Add Campaign

Campaign ↓	Advertiser	Start Date	End Date	Scheduled Impressions	Scheduled Clicks	Status
Diana campaign for cooldrinks	Diana company	2007-07-24	2007-08-24	1500	0	Active
Human rights commission	Human Rights	2007-06-10	2008-06-03	1000000	0	Active
OASIS Default	OASIS Professional Services	0000-00-00	0000-00-00	0	0	Active

Add Campaign

Active · Inactive · Suspended · Cancelled · Completed · Subpublishers

To only show campaigns which have specific statuses, you can check and uncheck the boxes at the bottom of the form and hit "Filter". Click on "Add Campaign" to add a new campaign. To edit a campaign, click on the name of the campaign.

In the image below, you see the details of a campaign.



oasis

new campaign | **campaigns** | advertisers | sections | reports | invoicing | admin | documentation

Campaign Details

Publisher: OASIS Owner

Name: Status:

Advertiser:

Start Date:

End Date:

Impressions:

Clicks:

Deliver:

Weight:

Impression Cap: per w d h m s

Companion Sharing:

Overflow OK:

Sunday	12 a.m.
Monday	1 a.m.
Tuesday	2 a.m.
Wednesday	3 a.m.
Thursday	4 a.m.
Friday	5 a.m.
Saturday	6 a.m.

CPM: \$ Agency Commission: %

CPC: \$ Purchase Order:

Fixed: \$ Force Invoice:

E-mail Reports:

Note:

Creatives

Name	Status	Wt	Type/Dim	Preview
diana cooldrink creative	Active	10	Image 250 x 250	

[Add Creative](#)

Section Assignments

Section	Type
OASIS Server -diana	Include

[Add / Remove Section Assignments](#)

OASIS 2.3 User's Manual

Use "Save" to save changes. Use "Delete" to delete the campaign and all associated creatives (you cannot delete a campaign while its status is set to "Active"). You will be prompted to confirm your decision if you opt to delete a campaign.

Use "Copy Campaign" to copy the campaign (and all its creatives and section assignments). A new campaign will be created with identical parameters, and with "Copy of" prepended to the name.

Name

Enter any name you like for the campaign. This name will show up on reports and invoices. It is good practice to use a naming convention for your campaigns which could include abbreviations of the advertiser, date range, section assignments, etc. You want to be able to quickly identify a campaign by its name in a list of many campaigns.

Advertiser

Select an advertiser from the drop-down list. Each campaign must belong to one and only one advertiser.

Status

There are a number of different status settings for campaigns:

- **Inactive** - the default status; inactive campaigns are not scheduled for delivery
- **Active** - only active campaigns are scheduled; you cannot delete a campaign while it is active. Note that setting a campaign to Active does not make delivery begin immediately (even if the campaign is scheduled to deliver today, you'll have to wait until the Hourly Assignments shared memory segment is rebuilt at the top of each hour.
- **Suspended** - the campaign will not be scheduled for delivery; functionally, there is no difference between "Suspended" and "Inactive", but you can use this status designation as a sort of reminder to yourself that the campaign is only temporarily halted.
- **Completed** - the maintenance scripts mark a campaign as "Completed" when either its impression target is met or its end date is reached (campaigns end at the *end* of the end date). You cannot do this manually. Obviously, they are not scheduled for delivery, either.
- **Cancelled** - you can manually mark a campaign as cancelled. It will not be scheduled for delivery, and it will not show up in the campaign list. Also, campaigns which are cancelled do not show up in revenue reports, nor are invoices generated for them. If you are halfway through an invoicing period, you may want to suspend the campaign and then cancel it after the invoicing period is over.

Start Date

You do not *have* to enter a start date unless you have an impressions target specified or a fixed cost for the campaign, in which case you must have a start and an end date.

End Date

You do not *have* to enter an end date unless you have an impressions target specified or a fixed cost for the campaign, in which case you must have a start and an end date.

When an end date is set, the campaign will stop delivering at midnight of the day *following* the end date (ie., the campaign will run *through* the entire end date.

Days of Week / Hours of Day

You can specify that a campaign only run on certain days of the week or certain hours of the day. By default, campaigns will run 24/7. In general, it is best to use this option only with campaigns that have no impressions target set, as you may end up making it impossible to meet the impressions target.

OASIS 2.3 User's Manual

Please note that if you select an hour, the campaign will run from the beginning of that hour until the beginning of the next hour. So if you select 8 a.m. through 5 p.m., the campaign will start running at 8 a.m. and stop running at 6 p.m.

If you have selected specific days of the week or hours of the day for the campaign and then specify different days/hours for the creatives, the creatives will only run on the days or hours where the campaign's selections and the creatives' selections overlap. So for instance, if the campaign is set to run Monday through Friday and the creative is set to run Friday and Saturday, the creative will only run on Friday. If a campaign is set to run from 5 a.m. to noon, and a creative in that campaign is set to run from 2 p.m. to 6 p.m., the creative will not run at all.

Impressions

Enter the number of impressions to be delivered for the campaign. If you enter a number here, you *must* have start and end dates specified.

You can leave this blank (or set to 0) and the campaign will run without a specific impression goal, delivering as many as possible. Campaigns without impression goals always have a lower priority than those campaigns with impression goals. All campaigns with impression targets will run first each hour, with the "remnant" campaigns with no impression goals picking up whatever is leftover after the guaranteed campaigns have met their hourly goals.

Clicks

Enter the number of clicks to be recorded for the campaign. The campaign will be marked as completed when the specified click goal has been reached.

Deliver

This setting determines how the campaign will be scheduled to meet its impression target. You can have the delivery done evenly by day, week, or month.

- Evenly by Day - an equal number of impressions will be delivered each day during the campaign.
- Evenly by Week - an equal number of impressions will be delivered each week (or week fragment) during the campaign. Weeks begin on Sundays, and each week fragment counts as a whole week. For example, if a campaign begins this Thursday and runs through next Monday, it counts as two weeks. Exactly half the impressions will be scheduled for Thursday through Saturday, and the other half will be scheduled for Sunday through Monday.
- Evenly by Month - an equal number of impressions will be delivered each month (or month fragment) during the campaign. For example, if a campaign begins on June 15 and ends August 15, it counts as three months (part of June, all of July, and part of August). One-third of the impressions will be scheduled for June 15-30, one-third for July 1-31, and one-third for August 1-15.

Weight

This only matters if you have Impressions set to 0. If so, this represents the relative weight between the campaigns. For example, if you have one campaign with a weight of 5 and another with a weight of 20, the second will run 4 times as often as the first. Keep in mind that campaigns with Impressions=0 run only after those with non-zero Impressions values have gotten their hourly targets. The default weight for a campaign is 10.

Impression Cap

This option allows you to set the number of times you wish each creative in the campaign to be seen by visitors for a specific time period. Enter a number and designate an interval. If you do not designate an interval, the impression cap is deemed absolute, that is, if you enter "5" with no interval, each user will see the

creatives 5 times over.

Impression caps are enforced at the individual creative level, so if you have 4 creatives in a campaign and set a campaign-level impression cap of 2 impressions per day, each of the 4 creatives may be displayed 2 times per day to each user, for a total of 8 impressions. If you have previously set individual impression caps for your creatives, setting it at the campaign level will overwrite the creative-level caps.

Impression caps are enforced using cookies, so users who have turned off cookies in their browsers may see more than the designated number of impressions.

Companion Sharing

Companion sections are sections which normally do not display the creatives from the same campaign at the same time. The campaign Companion Sharing setting will override the any companion sections for a specific campaign. If you want to have more than one creative from a campaign showing up in companion sections, check this box.

Note that even with companion sharing selected for a campaign, the delivery engine will not run the same creative on companion sections; it will only allow different creatives from the same campaign to share companion sections.

Overflow OK

For campaigns with specific impression targets, this value is irrelevant. Such campaigns will always deliver the exact number of impressions specified (assuming there is adequate inventory).

However, for campaigns that do not have specific impression targets, this box indicates whether it is OK to let their hourly delivery "overflow" into the next hour. At the top of each hour, the delivery engine is reloaded. During this reload time, OASIS continues to deliver ads from the overflow delivery tables. What this means is that if you scheduled a campaign with no specific impression target to run from June 1 to June 30, a few impressions may be delivered in the first few seconds of July 1. This may have undesirable results when advertisers' reports are generated, if they expect all impressions to be delivered precisely on the dates scheduled.

It is wise to leave this option checked for at least one site-wide "house" campaign so that you have impressions to deliver during the engine reload period.

By default, all campaigns have this option checked. If you change it at the campaign level, the changes propagate down to the creatives. Also, any new creatives inherit the setting from their parent campaign. Note, however, that individual creatives can have a different setting from the campaign.

CPM

Enter the CPM (cost per thousand) here. You can leave it empty or set it to zero.

CPC

Enter the CPC (cost per clickthrough) here. You can leave it empty or set it to zero.

Fixed

Enter the fixed cost of the campaign here. You can leave it empty or set it to zero. If you set it to non-zero, you have to specify a start and end date for the campaign (for invoicing purposes). Also, select how the fixed cost is to be paid:

- End of Campaign - the full fixed cost is billed in the last invoice period of the campaign

OASIS 2.3 User's Manual

- CIA (Cash in advance) - the full fixed cost is billed in the first invoice period of the campaign.
- By Period - the fixed cost is prorated across all invoice periods by the number of days in each invoice period.

Agency Commission

If there is an agency commission involved, enter the percentage here. This will be applied to the total charge of each invoice and will be deducted from the net amount due from the advertiser.

Purchase Order

If the advertiser is using a purchase order to place the buy, enter the number here so that it will appear on the invoice.

Force Invoice

If this box is checked, invoices will be generated for this campaign even if CPM, CPC, and Fixed are all 0.

E-mail Reports

A list of e-mail addresses to which should be sent weekly reports of the campaign's performance. Separate multiple addresses using commas. By default, the e-mail address supplied for the advertiser should appear here. You can remove it if you like or add more to it. E-mail reports are sent at the beginning of each Sunday.

Note

Use this space to enter notes about the campaign. Information in this field is purely for your convenience; it has no effect on the operation of the ad server.

Delivery Controls

Delivery controls allow you to specify special delivery targeting parameters. See the [Delivery Controls](#) section.

Creatives

If the campaign has creatives, they appear on this screen. To edit a creative, click on its name. To add a new creative, click "Add Creative". For information about managing individual Creatives, see the [Creative Details](#) documentation page.

Section Assignments

If the campaign has been assigned to sections, they appear on this screen. To change the sections to which this campaign is assigned, click on "Add/Remove Section Assignments".

[Documentation Home](#)

Campaign Insertion

The Campaign Insertion interface is designed to work like a "wizard", a popular interface design for tools for novices. It is intended to step the user through the creation of a new campaign, validating each piece of information as it is entered, leaving no room for error.

OASIS 2.3 - Campaign Insertion - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites

Address <http://61.12.116.52:81/mgmt/campaigninsertion.php> Go Links

new campaign campaigns advertisers sections reports invoicing admin documentation

New Campaign

Choose an advertiser:

-- OR --

Enter info for a new advertiser:

Name *

Street Address *

City *

State *

Postal Code *

Country

Phone

Fax

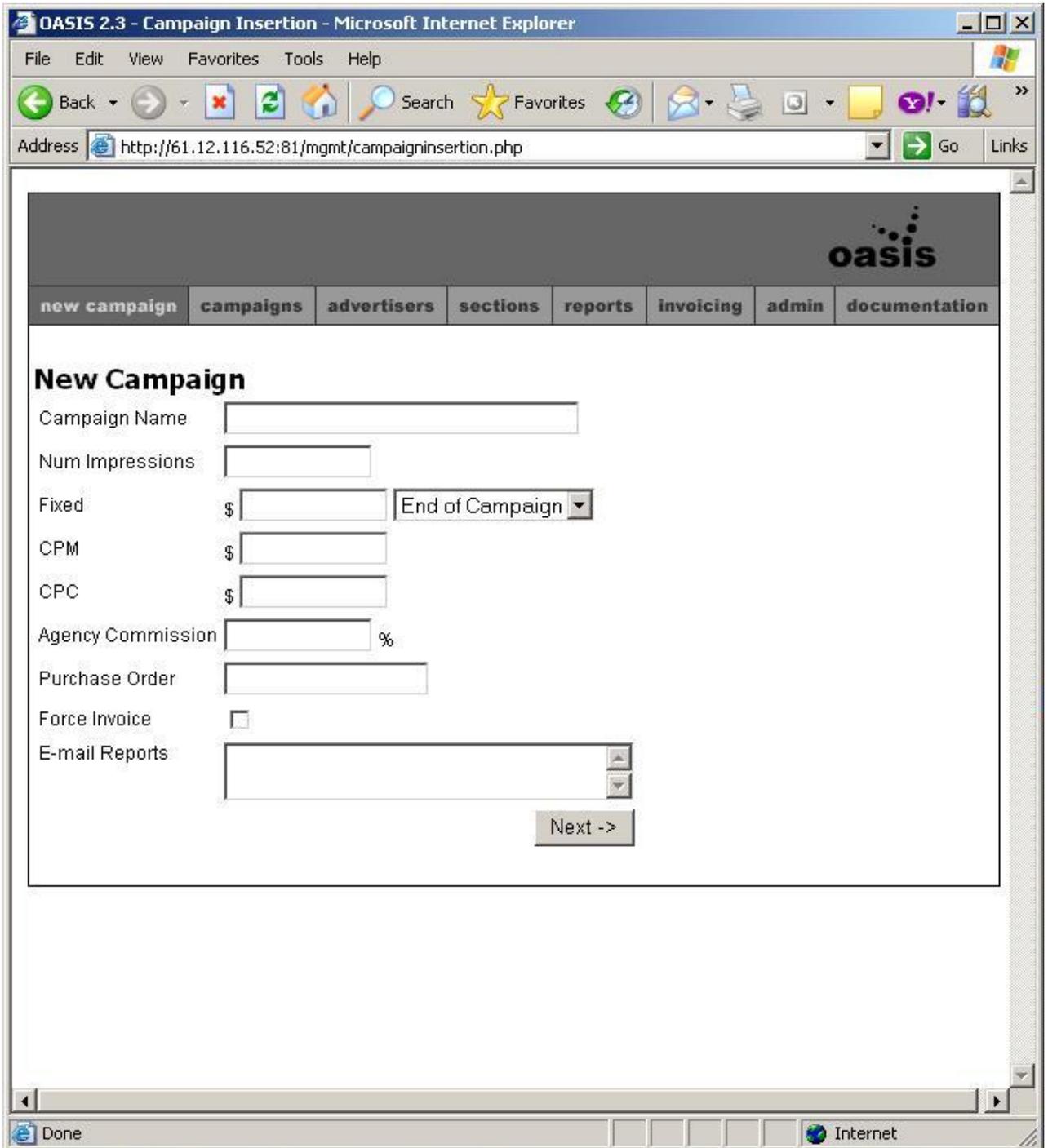
Contact E-mail

* Required fields if entering a new advertiser

Next ->

The initial screen prompts the user for advertiser information. If the advertiser is already in the database, you can just select it from the dropdown box. Otherwise, you must enter all the fields that have asterisks by them. These fields are required because of their importance in invoicing.

Hit "Next" to proceed to the next screen.



On this screen, you must enter a Campaign Name and a value in Fixed, CPM, or CPC. You may enter 0 for Fixed if this campaign is not a revenue-generating campaign. By default, invoices are not generated for non-revenue campaigns. If you wish an invoice to be sent to a client (of course, it will show \$0 due, but it will provide a report of the campaign's performance), check the box labeled "Force Invoice".

CPM

Enter the CPM (cost per thousand) here. You can leave it empty or set it to zero.

CPC

Enter the CPC (cost per clickthrough) here. You can leave it empty or set it to zero.

Fixed

Enter the fixed cost of the campaign here. You can leave it empty or set it to zero. If you set it to non-zero, you have to specify a start and end date for the campaign (for invoicing purposes) Also, select how the fixed cost is to be paid:

- **End of Campaign** - the full fixed cost is billed in the last invoice period of the campaign
- **CIA** (Cash in advance) - the full fixed cost is billed in the first invoice period of the campaign. The campaign insertion interface will automatically generate an invoice for the campaign if you choose CIA, marking it as paid, but not sent.
- **By Period** - the fixed cost is prorated across all invoice periods by the number of days in each invoice period.

Agency Commission

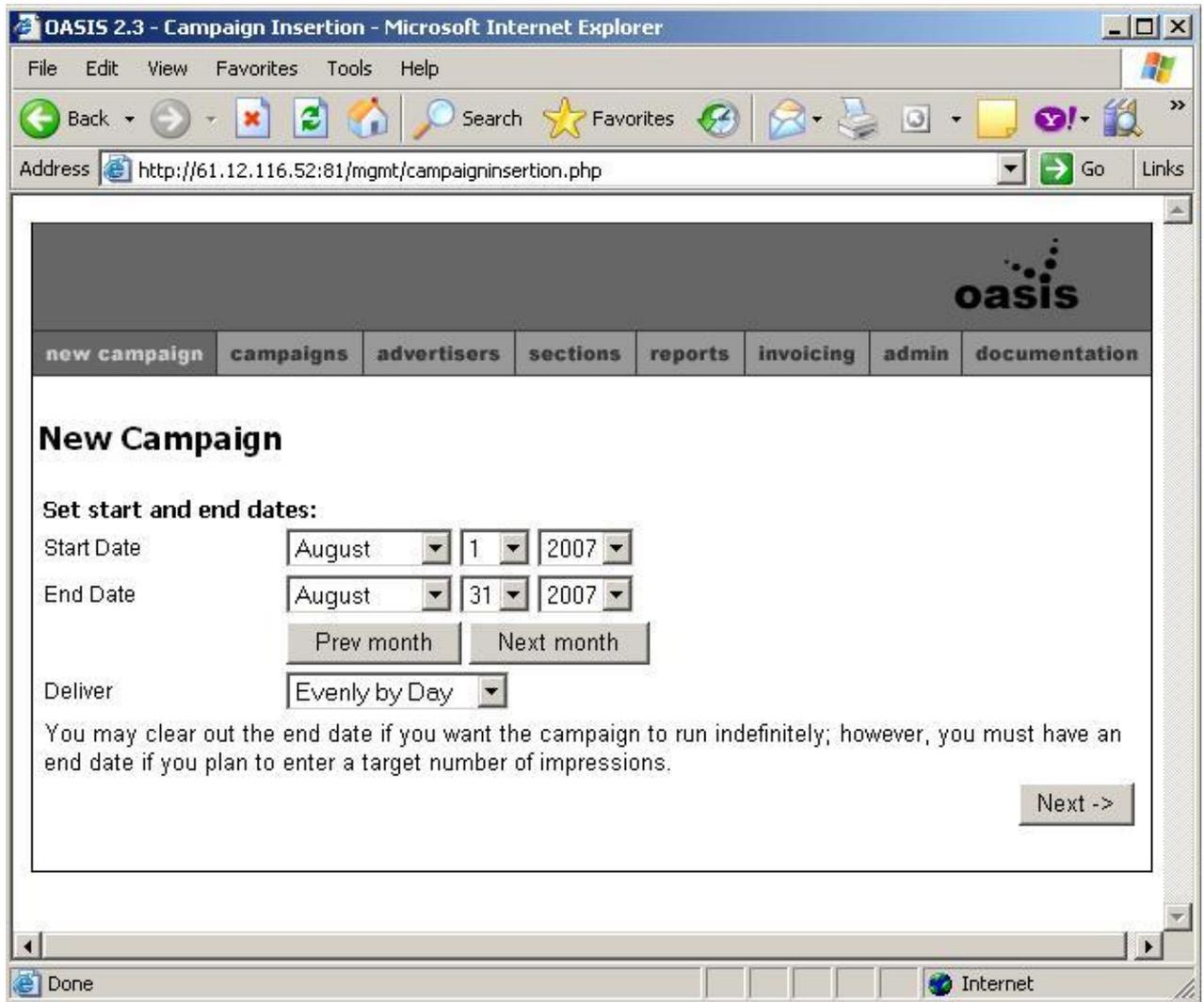
If there is an agency commission involved, enter the percentage here. This will be applied to the total charge of each invoice and will be deducted from the net amount due from the advertiser.

Purchase Order

If the advertiser is using a purchase order to place the buy, enter the number here so that it will appear on the invoice.

E-mail Reports

A list of e-mail addresses to which should be sent weekly reports of the campaign's performance. Separate multiple addresses using commas. By default, the e-mail address supplied for the advertiser should appear here. You can remove it if you like or add more to it.

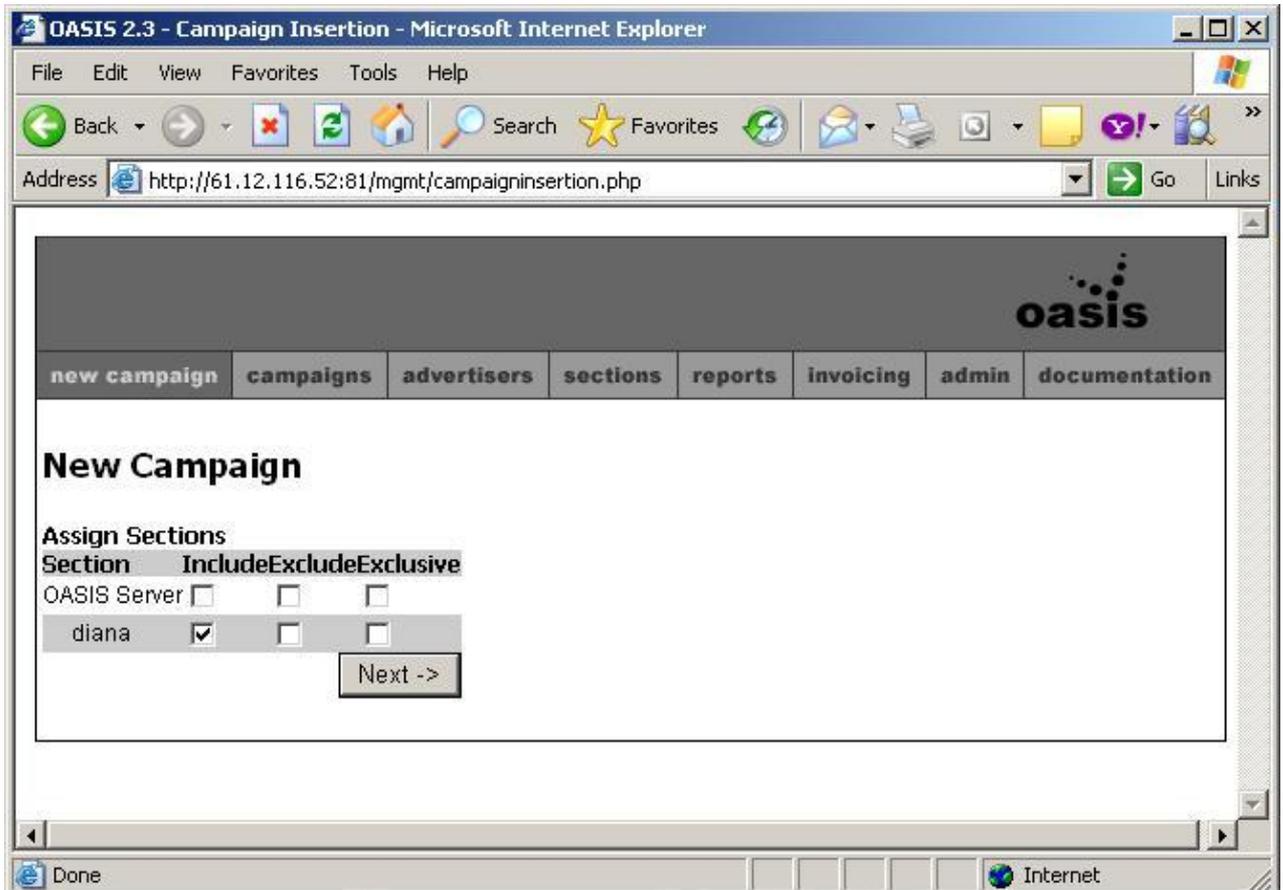


On this screen, you must enter a start and end date. You can use the "Prev month" and "Next month" buttons to jump forward and back in increments of one month (these buttons will always default to the first of the month for the start date and the last of the month for the end date).

Deliver

This setting determines how the campaign will be scheduled to meet its impression target. You can have the delivery done evenly by day, week, or month.

- **Evenly by Day** - an equal number of impressions will be delivered each day during the campaign.
- **Evenly by Week** - an equal number of impressions will be delivered each week (or week fragment) during the campaign. Weeks begin on Sundays, and each week fragment counts as a whole week. For example, if a campaign begins this Thursday and runs through next Monday, it counts as two weeks. Exactly half the impressions will be scheduled for Thursday through Saturday, and the other half will be scheduled for Sunday through Monday.
- **Evenly by Month** - an equal number of impressions will be delivered each month (or month fragment) during the campaign. For example, if a campaign begins on June 15 and ends August 15, it counts as three months (part of June, all of July, and part of August). One-third of the impressions will be scheduled for June 15-30, one-third for July 1-31, and one-third for August 1-15.



On this screen, you assign the campaign to sections. You see the entire section tree, with the option to "Include", "Exclude", or make "Exclusive" the campaign/creative on a particular section.

Each hour, the maintenance script assigns creatives to sections, propogating these assignments down the section tree. They are processed in this order:

1. Campaign Includes
2. Creative Includes
3. Campaign Excludes
4. Creative Excludes
5. Campaign Exclusives
6. Creative Exclusives

In this way, creative assignments always override those of their parent campaigns. Also, exclusive assignments override all others.

Now it is time to add the creatives to the campaign. For each creative, enter a name and a clickthrough URL. You'll also need to either browse for a local file or enter a URL for a third party ad server.

If this ad is a third-party redirection, enter the other ad server's banner URL here. Use the special tags:

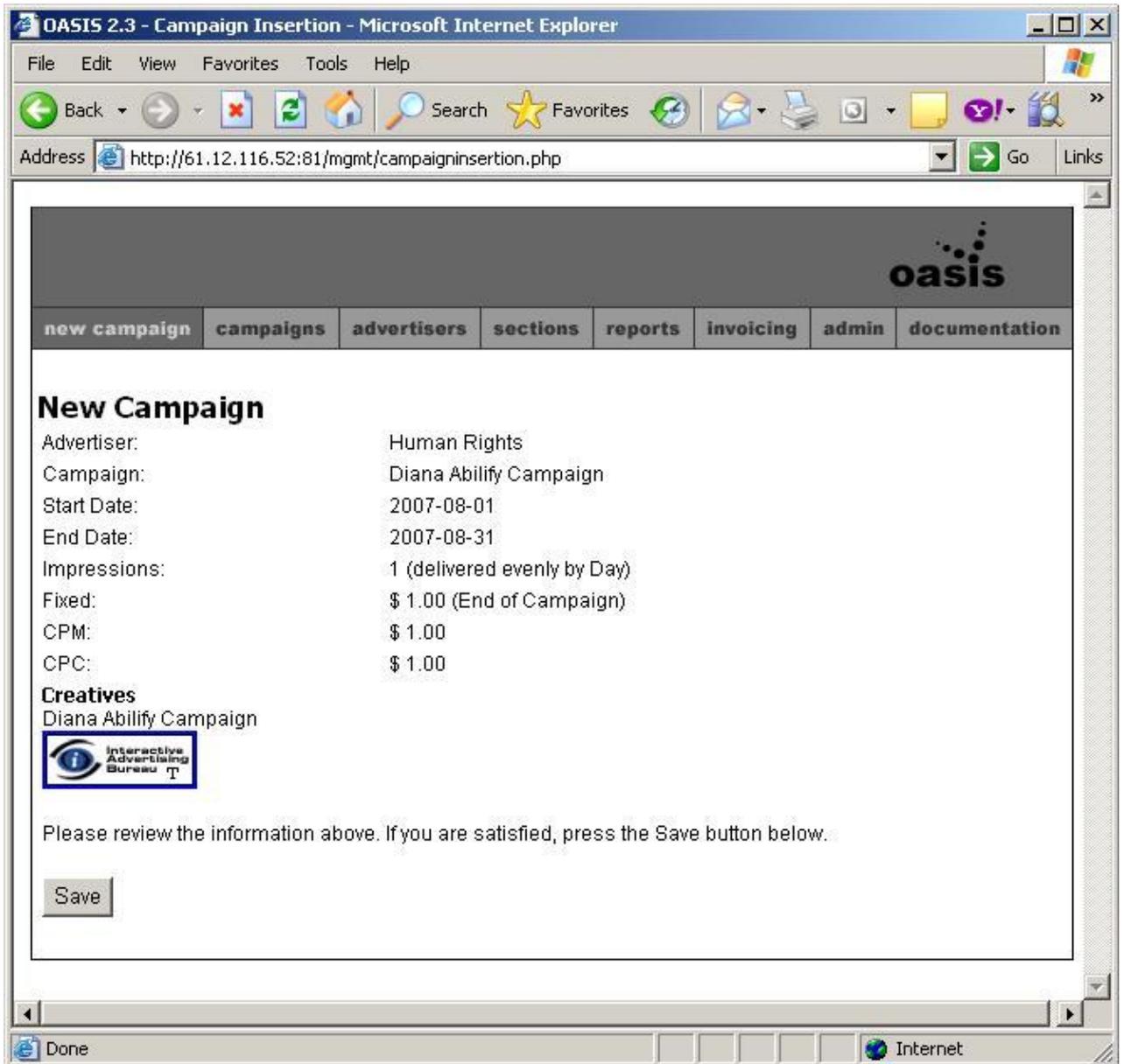
- [OASISCB] for a random cache-busting number:
- [OASISTIMESTAMP] for a UNIX-style timestamp (seconds since Jan 1, 1970)
- [OASISREFERER] for the URL of the referring page

For example:

```
http://ad-adex3.flycast.com/server/ad/WRALOnLine/FlycastROS/?cb=[OASISCB]
```

At delivery time, [OASISCB] will be replaced with a random number to prevent caching.

If you want to set up another creative, hit "More". Otherwise, hit "Finish".



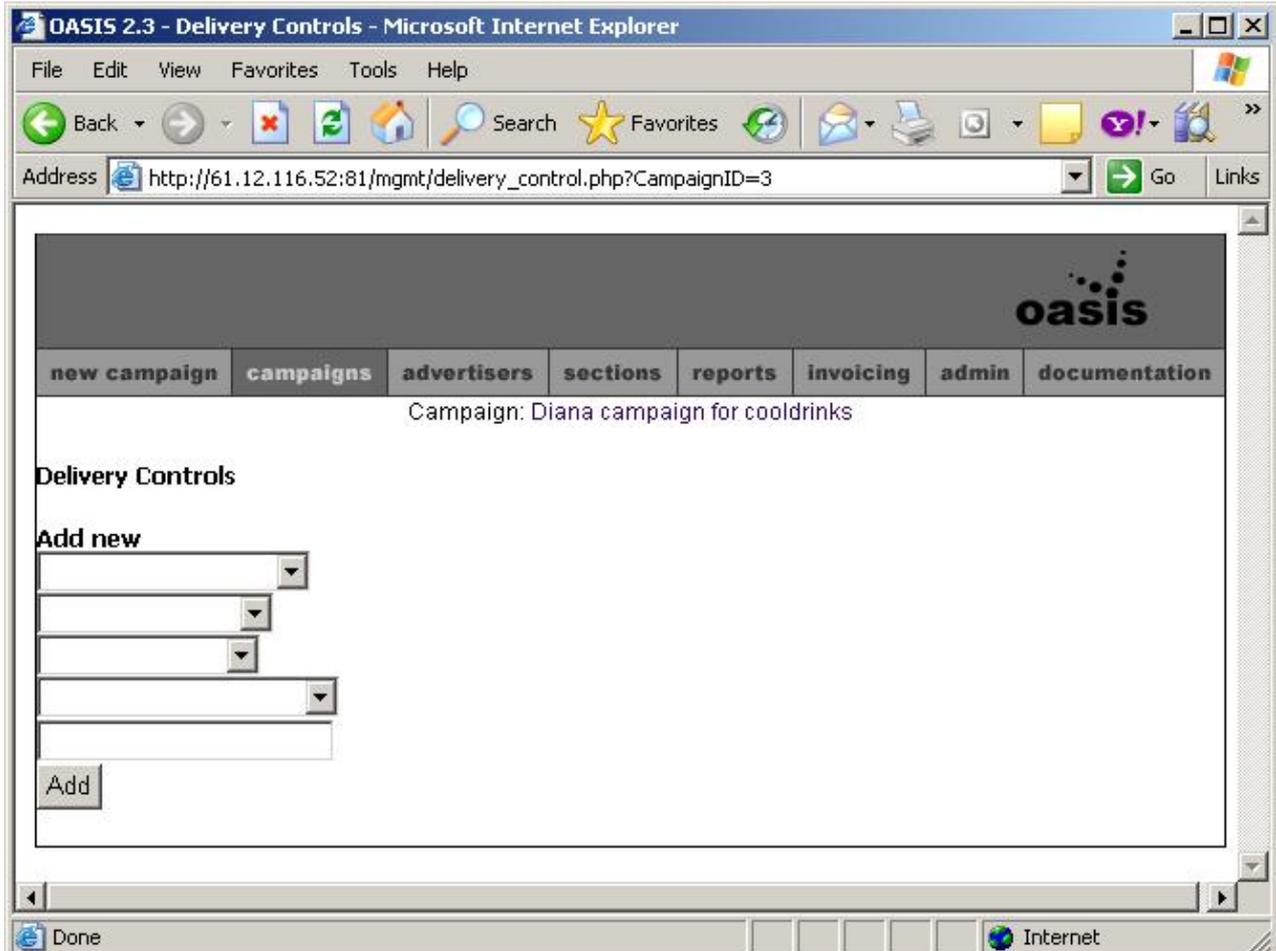
The last step is to look over the summary of the campaign. If all looks well, hit "Save". Depending on how you have configured the admin preferences using the InsertActive preference, your campaign may go live immediately, or it may be entered as an "Inactive" campaign, with e-mail going to the administrator, who can make it live.

[Documentation Home](#)

Delivery Controls

Delivery controls allow you to fine-tune the audience for a creative or an entire campaign through targeting. Click on "Delivery Controls" from the campaign interface or the creative interface to define the controls for your ads.

You will see the following interface:



The dropdowns allow you to specify the following:

- **Delivery**
 - ◆ Deliver - provide preferential delivery to impressions that satisfy this delivery control
 - ◆ Deliver only - deliver only when the impression satisfies this delivery control
 - ◆ Do not deliver - do not deliver to impressions that satisfy this delivery control
- **Filter Type**
 - ◆ IP address - compare the IP address of the visitor to the specified pattern
 - ◆ Domain - compare the hostname of the visitor to the specified pattern (this is slower than using the IP address, as it requires a name server lookup)
 - ◆ Language - compare the accepted language (as stated by the visitor's browser) to the specified pattern
 - ◆ User agent - compare the browser's User Agent string to the specified pattern

- ◆ Keyword - compare the value of the "k" CGI variable to the specified pattern; note that keyword matches take priority over other delivery control matches
- ◆ Country - using geotargeting information, deliver based on the visitor's country; format of the codes depends on your geodata provider's database. See [Geographic Targeting](#) .
- ◆ State - using geotargeting information, deliver based on the visitor's region within his country; format of the region names depends on your geodata provider's database (but they are returned as "country_code:region_name", regardless of how the country code and region name themselves are formatted). See [Geographic Targeting](#) .
- ◆ Metro Area - using geotargeting information, deliver based on the visitor's metro area (if available); returned as "country_code:region_name:metro_area".
- ◆ City - using geotargeting information, deliver based on the visitor's city; returned as "country_code:region_name:metro_area:city_name".
- ◆ PHP Expression - you can enter an arbitrary PHP expression that will be evaluated at delivery time; note that "Pattern Type" has no effect on PHP Expressions
- **Sense**
 - ◆ Matches - the delivery control is satisfied if the designated filter matches the specified pattern
 - ◆ Does not match - the delivery control is satisfied if the designated filter does not match the specified pattern
- **Pattern type**
 - ◆ String - the pattern is a simple string; any part of the test value can match the pattern

Examples:

Pattern	Test value	Result
199.72	199.72.1.1	matches
	251.8.127.9	does not match
foo	somehost.foo.com	matches
	somehost.bar.com	does not match

- ◆ Wildcard - the pattern is a wildcard pattern, with the special characters '?' and '*' representing any single character or any single group of 0 or more characters, respectively.

Examples:

Pattern	Test value	Result
foo	somehost.foo.com	matches
*foo	somehost.foo.com	does not match
foo*	somehost.foo.com	does not match
somehost?foo.com	somehost.foo.com	matches
somehost.?foo.com	somehost.foo.com	matches

- ◆ Regular expression - the pattern is a perl-compatible regular expression; for power users only.

- **Pattern**

The pattern which is to be compared. The syntax of this pattern depends on the pattern type.

You can specify multiple patterns by separating them with commas. Note that because of this, you cannot use commas in your patterns (this should not be a problem, as a comma would not be very useful in defining the various types of patterns).

OASIS 2.3 User's Manual

Delivery controls can be considered as a group of rules for delivering a creative. When the delivery engine is evaluating each creative to determine whether or not to deliver it, it evaluates all the delivery controls for that creative. If a delivery control is satisfied, action is taken based on how you have specified the delivery option for each rule.

A delivery control is satisfied when the filter variable matches one of your patterns (if you have selected "do not match", the control is satisfied when the filter variable does not match).

If a creative's delivery control specifies "Deliver", and it is satisfied, the creative will get preferential weighting. The first creative encountered by the delivery engine with a satisfied "Deliver" control will be delivered to the end user.

If a creative's delivery control specifies "do not deliver", and it is satisfied, the creative will not be delivered at all.

If a creative's delivery control specifies "deliver only", and it is satisfied, the creative will be delivered; if the control is not satisfied, the creative will not be delivered.

Note that a satisfied "do not deliver" control or an unsatisfied "deliver only" control will override a satisfied "deliver" or a satisfied "deliver only" control. In such a situation, the creative will not be delivered.

Let's consider the following example:

Creative 1	DC1-1 Deliver when	Language matches	String	en
	DC1-2 Deliver when	Language matches	String	fr
	DC1-3 Do not deliver when	Domain matches	Wildcard	*.yahoo.*
Creative 2	DC2-1 Deliver when	User agent matches	Wildcard	Gecko*
	DC2-2 Deliver only when	IP address matches	String	199.72
Creative 3	DC3-1 Deliver only when	Keyword matches	String	beer
Creative 4	DC4-1 Deliver when	Language does not match	String	fr
Creative 5	DC5-1 Deliver when	Keyword matches	Wildcard	win*2000

Now consider the following series of requests:

IP address 204.66.101.92

Domain www.foo.com

Language en-us

User agent Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)

Keyword -

The delivery controls for Creative 1 are checked; DC1-1 is satisfied while DC1-2 and DC1-3 are not. Creative 1 will get preferential delivery. Creative 2's delivery controls are checked; DC2-1 is not satisfied, and neither is DC2-2. Since DC2-2 stipulates "deliver only when", Creative 2 will not be considered for delivery. Creative 3's delivery controls are checked; DC3-1 is not satisfied; because it stipulates "deliver only when", it will not be considered for delivery. Creative 4's delivery controls are checked; DC4-1 is satisfied, so Creative 4 will get preferential delivery. Finally, Creative 5's delivery controls are checked; DC5-1 is not satisfied, so Creative 5 gets normal priority for delivery.

Creatives 1 and 4 get preferential delivery, so one of the two will be selected arbitrarily for delivery. As soon as one of them has reached its impression target, the other will be selected for all such future impressions.

IP address 199.72.8.96

Domain www.bar.com

Language en-us

User agent Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)

Keyword windows 2000

DC1-1 is satisfied, DC1-2 and DC1-3 are not. Creative 1 will get preferential delivery. DC2-1 is not satisfied; DC2-2 is. Creative 2 will also get preferential delivery. DC3-1 is not satisfied, so Creative 3 will not be considered. DC4-1 is satisfied, so Creative 4 will get preferential delivery. DC5-1 is satisfied, so Creative 5 will get preferential delivery; since this is a keyword delivery control, it will take priority over the other delivery controls, and Creative 5 will be selected.

[Documentation Home](#)

Geographic Targeting

Geographic targeting is part of OASIS's "Delivery Controls". Delivery controls can be applied on either the campaign-level or the creative level. Geographic targeting is the delivery of a campaign or creative based on the location of the visitor. In order to determine in which country or state the visitor is located, the ad server translates the visitor's IP address into a physical location using a commercial geo-database of IP-to-location mappings. OASIS uses the MaxMind "Geo IP Region Database".

Geo-targeting can be done either by country or by U.S. state:

- by *Country* - Country delivery targeting can be done for any international country. [Click here to see the 245 available country codes](#) that should be used for country targeting.
- by *State* (or Province)- U.S. State and Canadian Province delivery targeting uses the Country:State format. [Click here to see the 75 available state/province codes](#) that should be used for U.S. state or Canadian province targeting.

Targeting by Metro Area and City is currently not available with OASIS.

Geographic targeting is done by creating a "country" or "state" delivery control with the delivery control's pattern matching the country + state codes listed in the links above. If you are targeting by country, you simply use the country code by itself. If you are targeting to a state, you must use the country code and the state code separated by a colon.

EXAMPLES

1) To deliver ads only to visitors from California, create a delivery control that looks like this:

Deliver only when State matches the string US:CA

2) To delivery ads only to visitors from Germany, create a delivery control that looks like this:

Deliver only when Country matches the string DE

3) To deliver only to non-U.S. visitors, create a delivery control like this:

Do not deliver when Country matches the string US

4) To target multiple countries or U.S. states, create only one delivery control but enter all the codes, separated by commas:

Deliver only when State matches the string US:WI,US:MN,US:MI

The above example will delivery only to visitors from Wisconsin, Minnesota or Michigan.

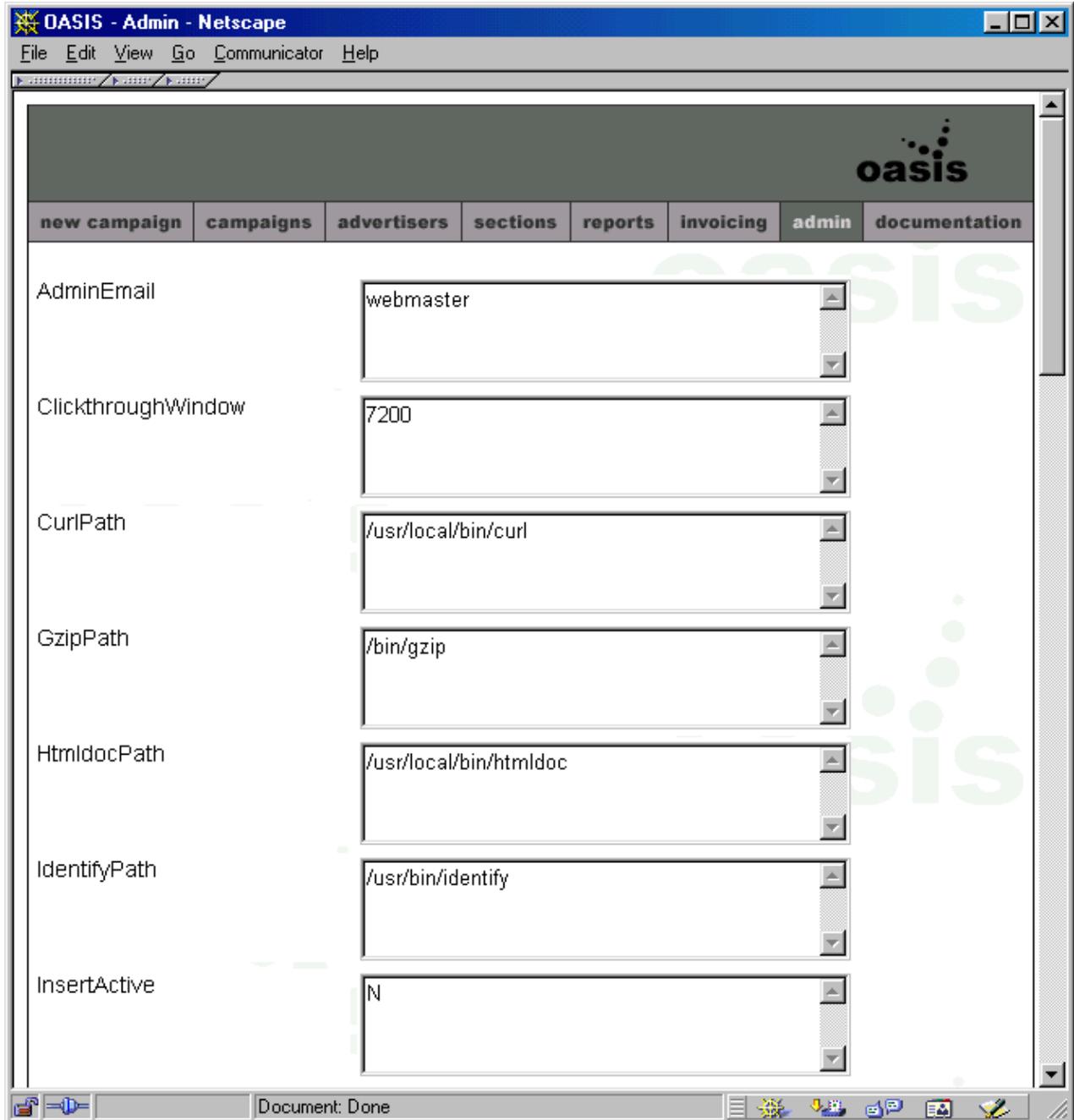
[Documentation Home](#)

Creative Details

Use Creative Details interface to manage your individual creatives.

Below you see the details of a creative. Use "Save" to save changes. Use "Delete" to delete the creative (you cannot delete a creative while its status is set to "Active"). You will be prompted to confirm your decision if you opt to delete a creative.

Use "Copy Creative" to copy the creative (and all its section assignments). A new creative will be created with identical parameters, and with "Copy of" prepended to the name.



Name

Enter any name you like for the creative. This name will appear on reports, so you might want something fairly descriptive.

Status

There are a number of different status settings for creatives:

- **Active** - the default status. Only active creatives in active campaigns are scheduled; you cannot delete a creative while it is active.
- **Suspended** - the creative will not be scheduled for delivery.
- **Completed** - the maintenance scripts mark a creative as "Completed" when either its impression target is met or its end date is reached (creatives end at the *end* of the end date). You cannot do this manually. Obviously, completed creatives are not scheduled for delivery.
- **Cancelled** - you can manually mark a creative as cancelled. It will not be scheduled for delivery.

By default, a creative is set to "Active". If it has an end date or impression target set, when it hits either, it will be set to "Completed". You can set it manually to "Cancelled" if you need to stop it from running.

Dimensions

These are automatically filled in when you upload an image file or enter a ThirdParty URL, but you can override them. Do so at your own risk.

MIME Type

This is automatically set when an image is uploaded or a ThirdParty URL is entered. OASIS currently recognizes GIFs, JPEGs, and PNGs. Rich media creatives are given a MIME type of "text/html"

Clickthrough URL

The URL to which a user will be redirected if he clicks on the banner. You can use the following special tags in the ClickURL which will be replaced when the click occurs:

- [OASISCB] for a random cache-busting number:
- [OASISTIMESTAMP] for a UNIX-style timestamp (seconds since Jan 1, 1970)
- [OASISREFERER] for the URL of the referring page - Only works with IMG or Javascript ad tags! Will not work with IFRAME tags.
- [OASISHOSTNAME] for the Hostname of the referring page to enable relative click URLs - Only works with IMG or Javascript ad tags! Will not work with IFRAME tags.

For example:

```
http://[OASISHOSTNAME]/directory/file.html
```

Upon click , [OASISHOSTNAME] will be replaced with the hostname of the page on which the ad appears.

Weight

This is the weight of this creative relative to the other creatives in the campaign. This only matters if the creative does not have an impression target set. The default weight for a creative is 10.

Impressions

You can set an impression target for a creative. Suppose the campaign is scheduled for 100,000 impressions. It has two creatives, and you want to make sure that one gets exactly 200,000 impressions. You can set this here. (Note that setting the weight of one creative to 1 and the other to 4 would have *almost* the same effect,

but there would be no guarantee that the first would get *exactly* 200,000 impressions).

Of course, you cannot set this value greater than the impression total for the campaign.

Also note that if you set this value to a number less than the impression total for the campaign, you must have another creative to "pick up the slack", that is, the sum of Impressions for all creatives must equal the campaign total, or there must be at least one creative that has Impressions=0.

Clicks

Enter the number of clicks to be recorded for the creative. The creative will be marked as completed when the specified click goal has been reached.

Third Party

If this ad is a third-party redirection, enter the other ad server's banner URL here. You can use the following special tags in the Third Party field which will be replaced when the click occurs:

- [OASISCB] for a random cache-busting number:
- [OASISTIMESTAMP] for a UNIX-style timestamp (seconds since Jan 1, 1970)
- [OASISREFERER] for the URL of the referring page

For example:

```
http://ad-adex3.flycast.com/server/ad/WRALOnLine/FlycastROS/?cb=[OASISCB]
```

At delivery time, [OASISCB] will be replaced with a random number to prevent caching.

Start Date

The start date for the creative. By default, it will just use the start date of the campaign. If you set this to something earlier than the campaign's start date, the campaign's start date will override.

End Date

The end date for the creative. By default, it will just use the end date of the campaign. If you set this to something later than the campaign's end date, the campaign's end date will override.

When an end date is set, the creative will stop delivering at midnight of the day *following* the end date (ie., the creative will run *through* the entire end date).

Days of Week / Hours of Day

You can specify that a creative only run on certain days of the week or certain hours of the day. By default, creatives will run 24/7. In general, it is best to use this option only with campaigns/creatives that have no impressions target set, as you may end up making it impossible to meet the impressions target.

Please note that if you select an hour, the creative will run from the beginning of that hour until the beginning of the next hour. So if you select 8 a.m. through 5 p.m., the creative will start running at 8 a.m. and stop running at 6 p.m.

If you have selected specific days of the week or hours of the day for the campaign and then specify different days/hours for the creatives, the creatives will only run on the days or hours where the campaign's selections and the creatives' selections overlap. So for instance, if the campaign is set to run Monday through Friday and the creative is set to run Friday and Saturday, the creative will only run on Friday. If a campaign is set to run from 5 a.m. to noon, and a creative in that campaign is set to run from 2 p.m. to 6 p.m., the creative will not

run at all.

Impression Cap

This allows you to set an impression cap for individual creatives. See the description of Impression Cap under Campaigns.

Overflow OK

See the description of Overflow OK under the Campaign documentation.

Section Assignments

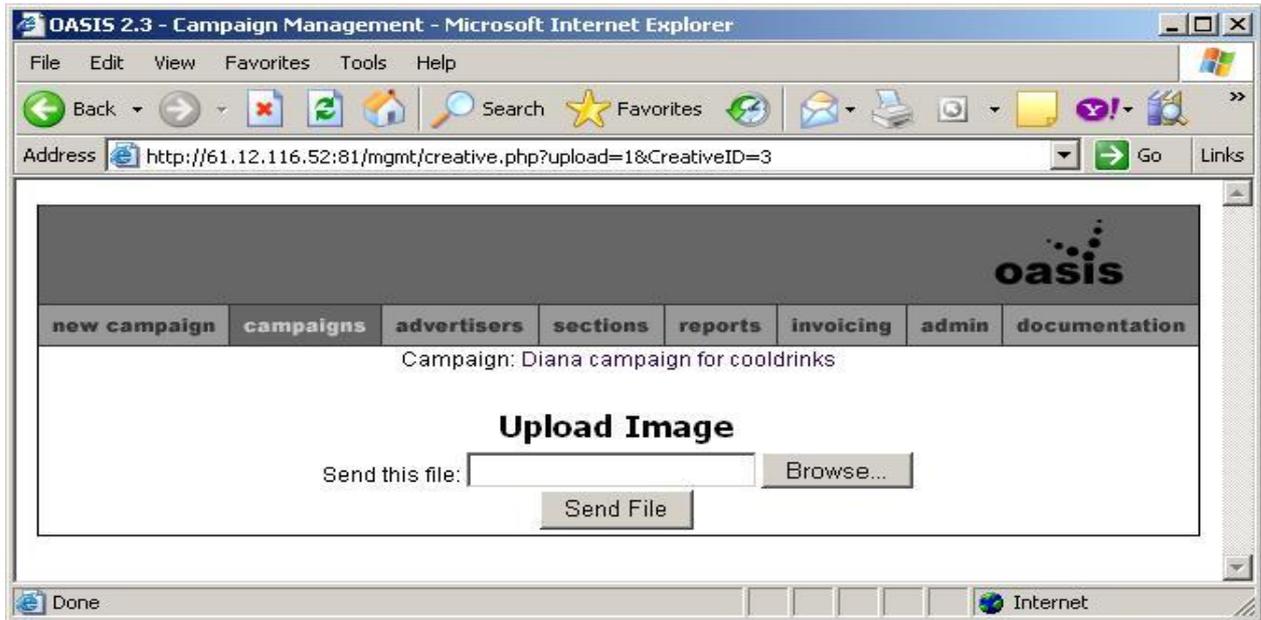
If the creative has been assigned to sections, they appear on this screen. To change the sections to which this campaign is assigned, click on "Add/Remove Section Assignments".

The section assignment screen is identical for both campaigns and creatives. On this screen, you see the entire section tree, with the option to "Include", "Exclude", or make "Exclusive" the creative on a particular section.

Each hour, the maintenance script assigns creatives to sections, propogating these assignments down the section tree. They are processed in this order:

- 1. Campaign Includes
- 2. Creative Includes
- 3. Campaign Excludes
- 4. Creative Excludes
- 5. Campaign Exclusives
- 6. Creative Exclusives

In this way, creative assignments always override those of their parent campaigns. Also, exclusive assignments override all others.



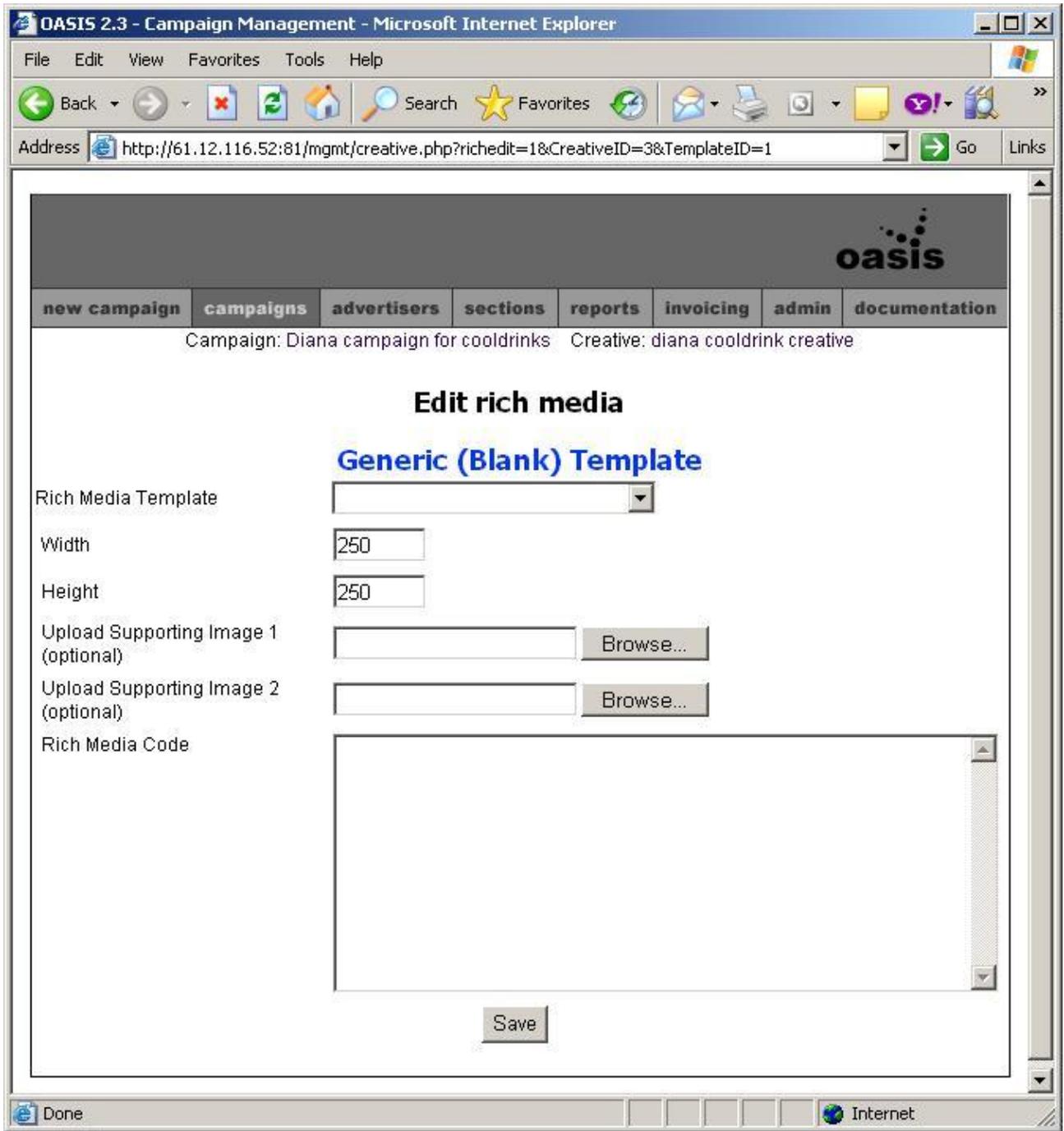
Uploading a new banner is simple. From the creative editing form, click on "Upload new image". Hit "Browse..." to look for a file on your computer. Once you find it, click "Send File" to upload it. Assuming that it is small enough for its dimensions (the administrator defines a maximum size for each dimension of

creative), you will be taken back to the Creative Details screen where you can edit the creative properties further.

Edit Rich Media

Clicking on the Edit Rich Media link allows you to enter any rich media code or third-party ad tags to be delivered to the page in place of a creative image. The Edit Rich Media page also allows you to use pre-defined rich media templates. See the [Rich Media Templates](#) documentation page for more information on using rich media templates.

The "Blank" template is used to enter and modify generic rich media which does not have any specific format. Shown below is the Blank template page.



You can edit any type of generic rich media ads using this form. From the creative editing form, click on "Edit rich media". Next, choose the rich media template called "Blank". Enter the width and height of your ad. You can enter any HTML (or for that matter, any markup language) you wish into the Rich Media Code field. This code may reference files on other Web servers, like images, Shockwave files, or Java applets. Alternatively, you can specify and upload up to two different "Supporting Images" that are used in your rich media code. The use of Supporting Images is optional. When a Supporting Image is uploaded, it will be saved into a directory on the OASIS file system corresponding to the creative's CreativeID number.

For example, if the image called sample.gif is uploaded as a Supporting Image for a creative with CreativeID=321, it will be saved in the directory "321" and publicly accessible with the URL:

OASIS 2.3 User's Manual

<http://ads.adhostingsolutions.com/images/321/sample.gif>

To see what your rich media ad will look like in a browser, hit "Save". On the creative editing form, click "Preview rich media". A new window should pop up with your ad in it.

Note that rich media ads cannot be served if you're using IMG-based tagging (`oasisi.php`). You need to use either the embedded method, the Javascript method, or the IFRAME method. See the [Tagging](#) section of the manual for more information.

To track clickthroughs with Rich Media ads, you can use special tags in the creative's source. The code:

```
[OASISCLICK:http://actual.url.here/]
```

will be replaced at delivery time with the appropriate clickthrough URL, and the resulting URL will record the click and redirect the user to `http://actual.url.here/`. The section, size, and creative will be recorded in the database when a user clicks on the link. You may include multiple clickthrough links in a single rich media creative, each with a different URL if you like.

Note that you can use the OASISCLICK tag in a standard HTML hyperlink:

```
<A HREF="[OASISCLICK:http://actual.url.here/]">
```

or you can do other things like pass them to Java applets:

```
<PARAM NAME="URL1" VALUE="[OASISCLICK:http://actual.url.here/]">
```

OASIS will tie the clicks back to the right creative. Unfortunately, if you have different links in the same creative, you won't be able to tell them apart in the reports at this point.

[Documentation Home](#)

Rich Media Templates

OASIS has a group of out-of-the-box rich media templates to simplify the creation of different types of rich media. Rich media templates allow us to easily and quickly make rich media creatives without having to know the special code that is required and without having to manually build the rich media code. This saves time and avoids mistakes when making creatives.

To use a rich media template, first access the creative's Creative Details page and click on the "Edit Rich Media" link. Choose the rich media template from the drop down list. The rich media template's fields will be displayed on the Edit Rich Media page. Note that the basic generic rich media template is accessed using the "Blank" template.

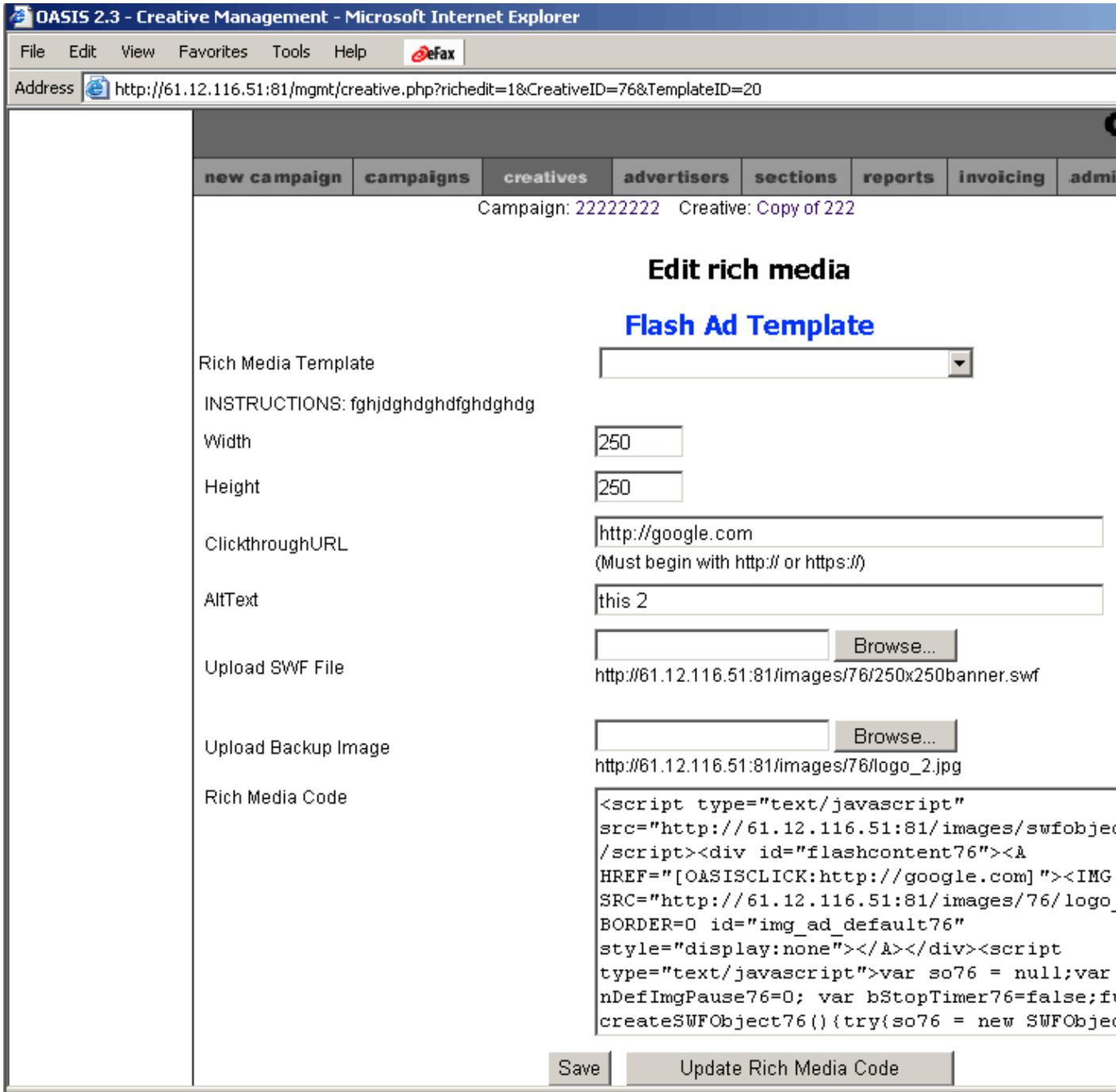
Enter the values for each field displayed on the template's Edit Rich Media page and upload any supporting images required to build the rich media creative.

After you have filled out the rich media template fields, click on the "Update Rich Media Code" button which will automatically populate the Rich Media Code field. Finally, click on the Save button to save the rich media creative.

When you are on the Creative Details page, you can preview the rich media by clicking on the "Preview Rich Media" link.

Links to any supporting files will be displayed on the Creative Details page. Make sure that the creative's MIME type is set to "text/html".

Shown below is an example of the Flash Ad template:



If you are the OASIS administrator, you can also create your own rich media templates by making new entries in the CreativeTemplates table in the OASIS database.

When you make a new template table entry, follow these rules:

- The CreativeTemplateContents field holds the rich media code. It must be written on one single line. It cannot be written with line breaks on several lines.
- The contents of the CreativeTemplateCode can be HTML, Javascript, XML or other types of code. It is also useful to call supporting files such as swfobject.js, which live in other locations which help enable the rich media code when it delivers.

OASIS 2.3 User's Manual

- We can choose which fields we want to display and use in the Edit Rich Media page of OASIS by enabling the needed fields in the CreativeTemplates table entry. For each field we choose to display, we can control the text title for the field in the table entry.
- The `quote` field in the database is used to insert double quotes into the rich media template. This is done because the raw CreativeTemplateContents code cannot contain quotes or it will not work. We must tell the database to substitute the `"` macro with `”` by inserting the `quote` into the CreativeTemplates quote field.

We use special macros in the CreativeTemplateContents field which will get replaced with the creative's database values when the user clicks the Update Rich Media button in the Edit Rich Media page. The macros are as follows:

- `%%AD_SERVER_URL%%` - This gets replaced with the ad server hostname.
- `"` - This gets replaced with the contents of the quote field.
- `%%Width%%` - Gets replaced with the creative's width setting.
- `%%Height%%` - Gets replaced with the creative's height setting.
- `%%CreativeID%%` - Gets replaced with the creative ID.
- `%%ClickthroughURL%%` - Gets replaced with the creative's ClickURL setting.
- `%%SupportingFile1%%` - Gets replaced with the name of the supporting file 1 that the user uploads.
- `%%SupportingFile2%%` - Gets replaced with the name of the supporting file 2 that the user uploads.
- `%%Value1%%` - Gets replaced with the user-entered value for Value1
- `%%Value2%%` - Gets replaced with the user-entered value for Value2
- `%%Value3%%` - Gets replaced with the user-entered value for Value3
- `%%Value4%%` - Gets replaced with the user-entered value for Value4
- `%%Value5%%` - Gets replaced with the user-entered value for Value5

[Documentation Home](#)

Reports

There are six types of reports in OASIS:

- revenue
- inventory
- advertiser
- campaign
- creative
- section

Select the type of report and set up any necessary parameters on the Reports page. The revenue report requires a start and end date. The inventory report requires a section and depth (the start and end dates are defined a priori in the Preferences under the Admin interface). The advertiser/campaign/creative reports require selecting a advertiser/campaign/creative and a start and end date. The Section report requires a section, a depth, and start and end dates.

For all report types, if you wish to see advertisers, campaigns, creatives and sections which belong to sub-publishers, check Show Subpublishers checkbox and click on Show.

Revenue Report

The revenue report summarizes all invoiceable revenue accrued during the specified period. It does not project revenue into the future, so CPM and CPC charges are only computed for impressions and clickthroughs that have actually been delivered.

The average CPM is equal to the total revenue from CPM charges divided by all impressions (whether sold by impression or not).

The average CPC is equal to the total revenue from CPC charges divided by all clickthroughs (whether sold by clickthrough or not).

Inventory Report

This report shows you projected inventory for the specified section (down to the depth you specified). It also shows you how much of that inventory is allocated (based on nightly simulations). By clicking on a date, you can get a breakdown of the sections' inventory for that day.

To change the number of days forward the projections are made or the number of days' worth of back data used, change the InvDaysForward or InvDaysBack preferences with the Admin interface.

Advertiser Report

An advertiser report summarizes all activity for a given advertiser for the specified period.

Campaign Report

A campaign report summarizes all activity for a given campaign for the specified period. This is the same exact report that is sent to an advertiser each week, summarizing the performance of the campaign for the week.

Creative Report

A creative report summarizes all activity for a given creative for the specified period.

Section Report

A section report summarizes all activity for a given section for the specified period. You can use this report to get an idea of how many impressions you have to give in a particular section of your site. When you select a section for the report, you must select a depth. Choose "All" to see numbers for the entire tree beneath your selected section. Choose "1" to only see the numbers for the selected section itself. Choose 2 - 5 to see more or less of the tree.

Report Tables and Formulas

For advertiser, campaign, creative and section reports, the results are broken out by dimension, by advertiser, by campaign, by creative, by month, by date, by hour, by publisher and by section.

Formulas:

OASIS 2.3 User's Manual

- Clickthrough rate = Clickthroughs / Total Impressions * 100
- Impression error rate = Impression Errors / Total Impressions * 100
- Click error rate = Click Errors / (Total Clicks + Click Errors) * 100

[Documentation Home](#)

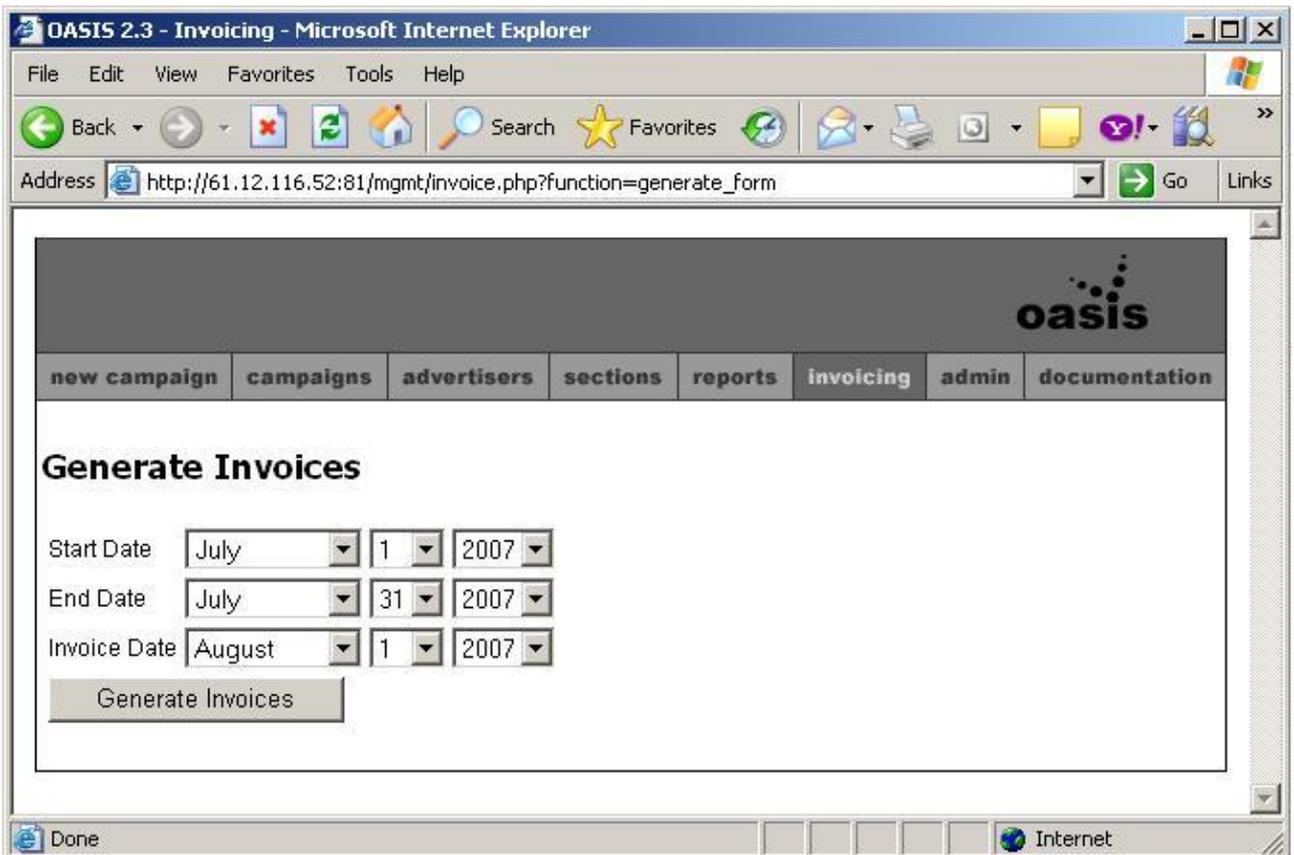
Invoicing

With OASIS, you can invoice your advertisers for their campaigns and generate invoices for your subpublishers. A campaign invoice is a sort of snapshot of campaign activity for a specified period (OASIS sort of assumes that you will invoice advertisers on a monthly basis, but you don't have to do it this way). The invoice records the number of impressions and clickthroughs delivered during the invoice period. It also stores the charges associated with those deliveries as well as the fixed-cost charges.

A subpublisher invoice contains any charges incurred by your subpublisher for his campaigns and those of his own subpublishers. In addition, it includes any credits due your subpublisher for running your campaigns (or those of your parent publishers). It also includes any monthly hosting charges.

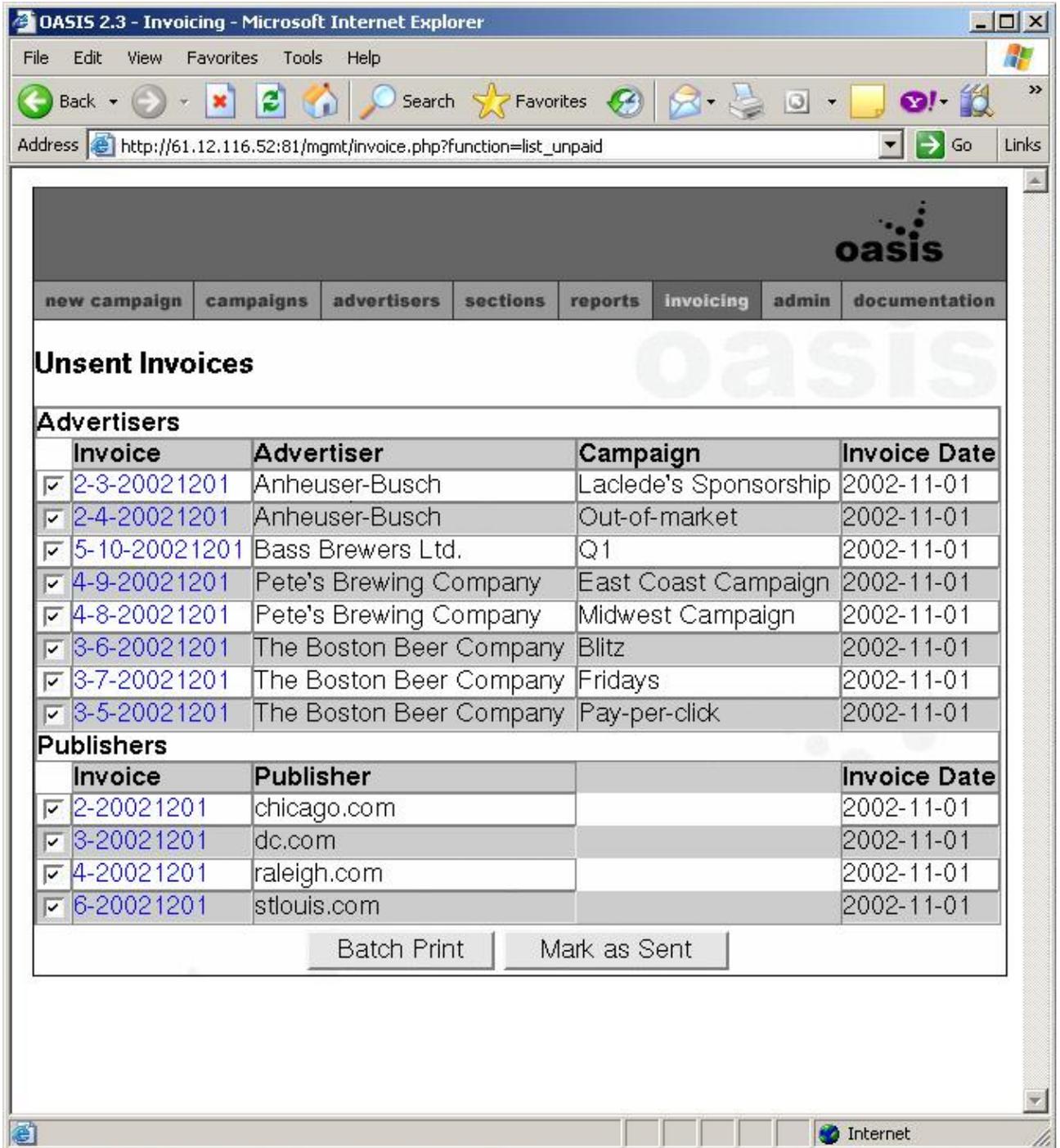
Invoices can be marked as "sent" or "unsent", and they can be marked as "paid" and "unpaid" to facilitate tracking payments.

For convenience, you can easily batch print groups of invoices. OASIS will use `htmldoc` (your administrator *did* install it, right?) to convert the invoices from HTML to PDF for one-step printing.



The first step is to generate invoices. Enter the beginning and end of the invoicing period and click "Generate Invoices". Any campaigns or subpublishers which have already been invoiced for *any portion* of this billing period will not get new invoices. It is imperative that your billing periods are consistent and do not overlap at all, or you may have a disaster on your hands as invoices are not generated properly for all campaigns.

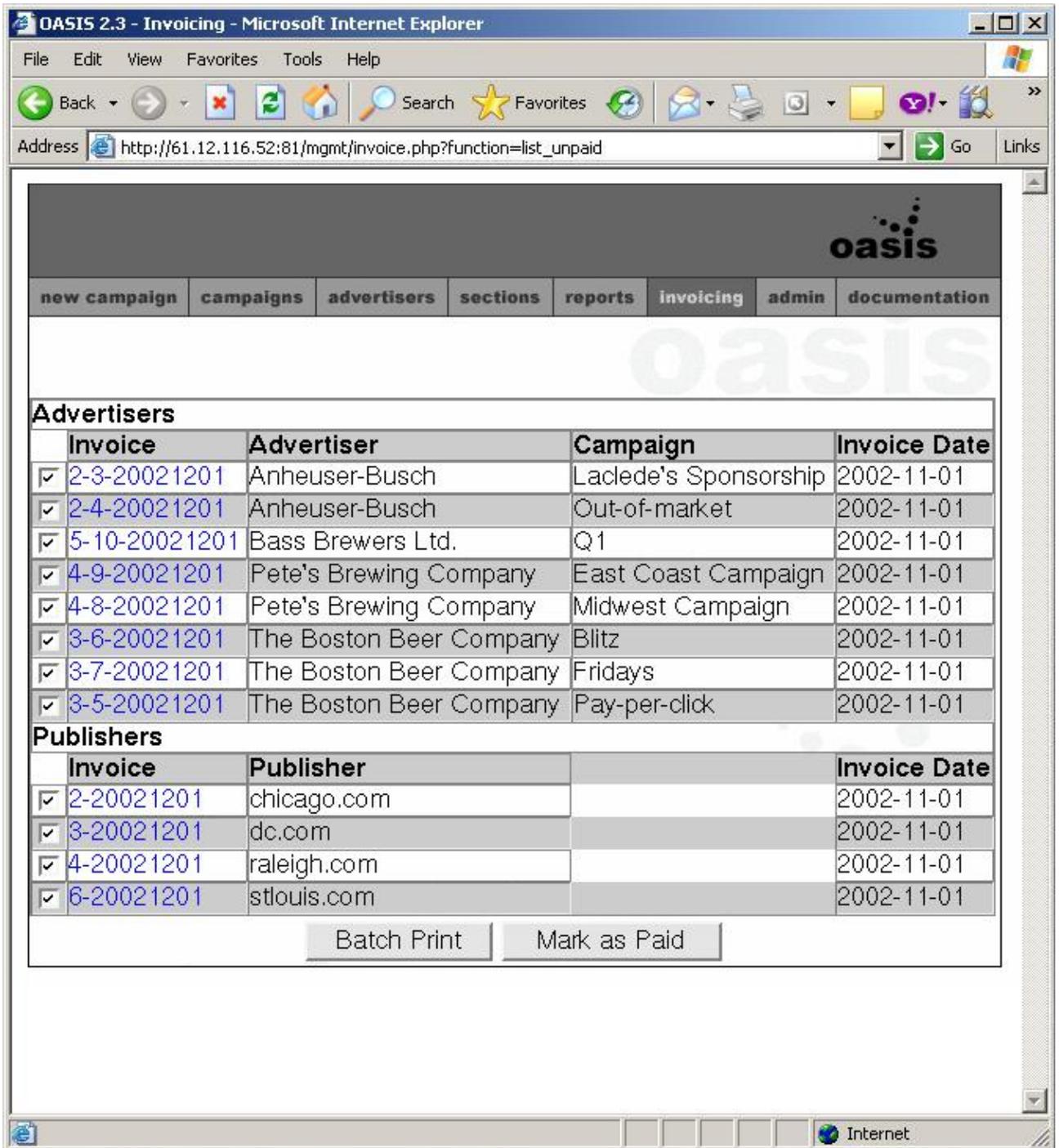
You will see a list of all of the invoices that are generated. To print these out, or to batch print them, click on "Back to the main invoicing menu" to go back to the main invoicing menu.



You can view all unsent invoices. Click on an invoice number to view it and/or print it.

Clicking "Batch Print" will generate a PDF of all checked invoices.

Clicking "Mark as Sent" will mark all checked invoices as sent. At this point, you'd have to go to the "View unpaid invoices" to see them again.



You can view all unpaid invoices. Click on an invoice number to view it and/or print it.

Clicking "Batch Print" will generate a PDF of all checked invoices.

Clicking "Mark as Paid" will mark all checked invoices as paid. At this point, only an administrator would be able to access the invoices (and then only by digging through the database).

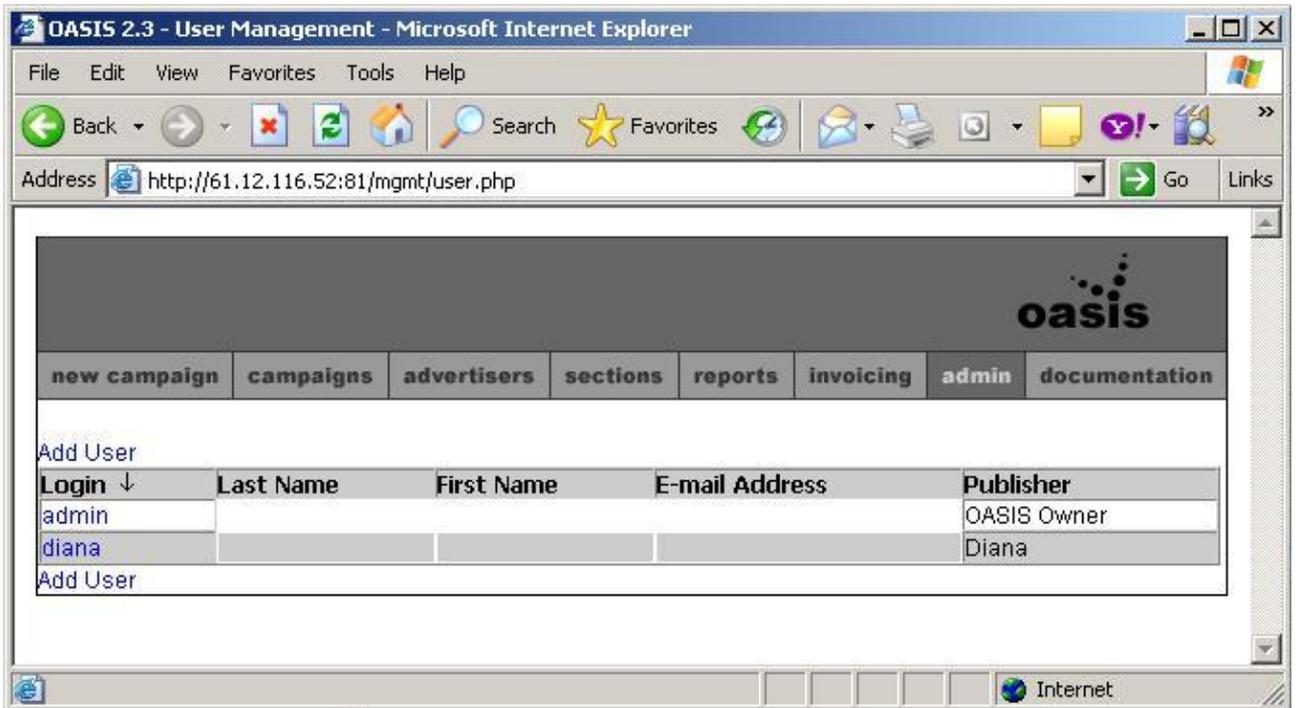
[Documentation Home](#)

Admin

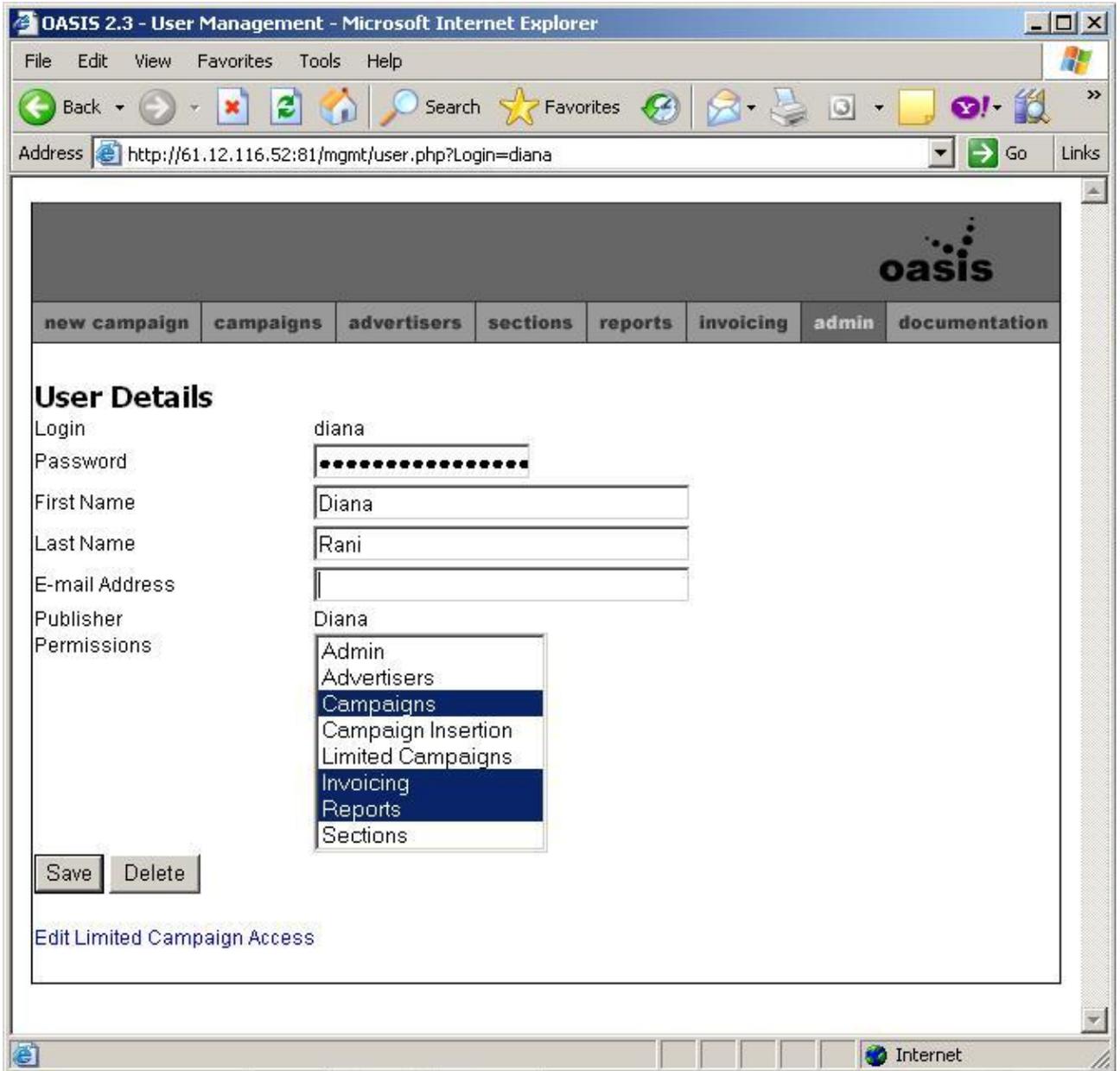
From the admin page, you can perform a lot of high-level actions. For this reason, it is best if access to this section is restricted to experienced users only.

User Management

Use this interface to control who can access the OASIS interface and which parts of the interface the users are able to use.



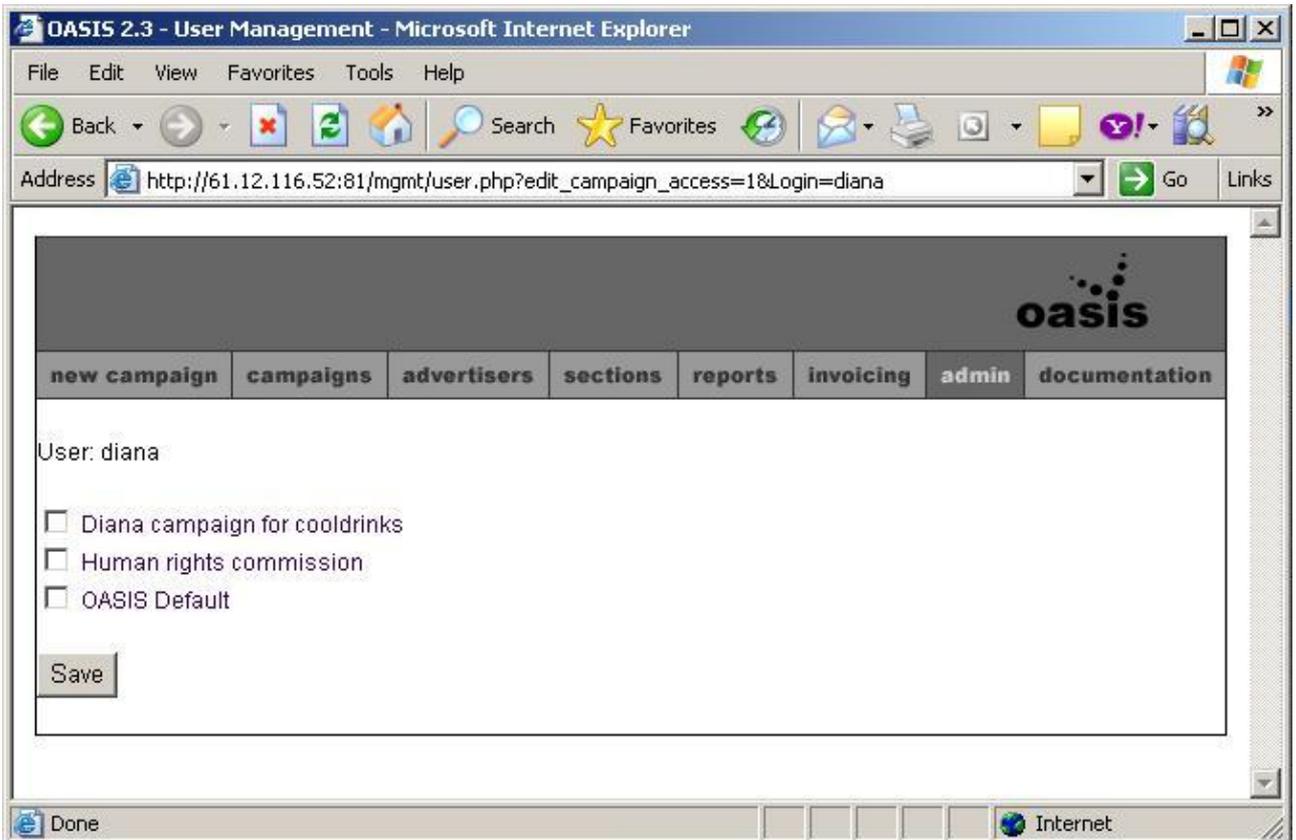
Here you see a listing of all active users in the system. To add a new user, click "Add User". To edit an existing user, click on the user's login.



The user editing form is fairly simple to use. Login and password are required, as is the Publisher (you can create and edit users for any publisher in your subpublisher tree). The other fields are optional. Select the permissions that you want to grant to the user. Hit "Save" to save your changes.

If you grant a user "LimitedCampaigns" access, you can indicate which campaigns the user is allowed to access with the "Edit Limited Campaign Access" link:

OASIS 2.3 User's Manual



Users who have LimitedCampaigns access will see the "Campaigns" link at the top of their interface, but when they click on it, they will only see the campaigns to which you have specifically granted access. Also, the interface will be somewhat limited (they will not be able to modify section assignments for campaigns and creatives).

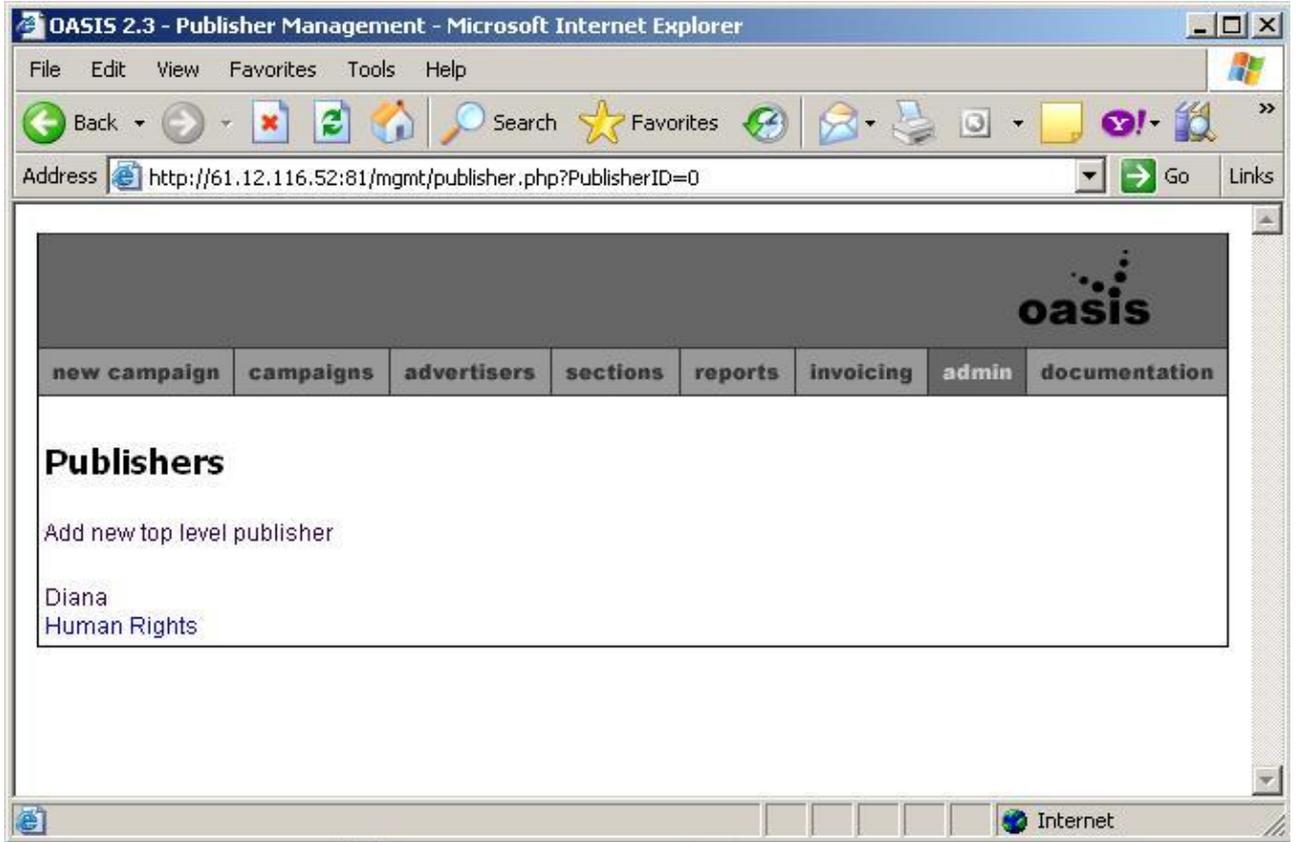
Note that granting "Campaigns" access overrides "LimitedCampaigns" access.

If you change your own password, the next link you follow within the OASIS interface will require you to enter the new password.

If you hit "Delete" to delete a user, you'll be asked to confirm this action. Follow the onscreen instructions to delete the user.

Publisher Management

Here you can create subpublishers. Each publisher on an OASIS server controls his own users, advertisers, sections, and campaigns. The OASIS publisher mechanism allows for arbitrarily deep hierarchies of publishers, so your subpublishers can have their own subpublishers and so forth.



Using the publisher interface, you can do a lot of things with your OASIS server. You can run an ad network where subpublishers share in revenue of campaigns placed by the top-level publishers. You can also run an ASP where subpublishers pay their parent publisher to run their own campaigns. You can even run a hybrid network where subpublishers get revenue for network campaigns, but pay to run their own campaigns.

Each publisher controls his own tree of subpublishers. If the list is empty, you can click on "add new top level publisher" to start a publisher tree. You can click on one of the publishers to edit that publisher's information or add a subpublisher to it.

OASIS 2.3 - Publisher Management - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address <http://61.12.116.52:81/mgmt/publisher.php?PublisherID=2> Go Links SnagIt

oasis

new campaign campaigns advertisers sections reports invoicing admin documentation

Publisher Details

Name

Street Address

City

State

Postal Code

Country

Contact

Phone

Contact E-mail

Publisher Status

Hosting Charges

Monthly Charge \$

CPM \$ or Percentage CPM %

CPC \$ or Percentage CPC %

Fixed \$ or Percentage Fixed %

Network Payments

CPM \$ or Percentage CPM %

CPC \$ or Percentage CPC %

Fixed \$ or Percentage Fixed %

[Add new subpublisher](#)

Enter the address information; it will appear on invoices you generate for your subpublishers. When you create a new publisher, you need to enter an administrative login and password, noting that you cannot reuse usernames and passwords on an OASIS server (so if your login is "admin", you'll have to give your subpublishers unique logins, like "admin-subpub1" and so on). You also need to enter a top-level section

OASIS 2.3 User's Manual

name and a parent section; this is the root of the subpublisher's section tree.

You may wish to go edit the new administrative user to control privileges. If you only want the new publisher to be able to log in to view reports and generate invoices (in other words, you don't want the new publisher to run its own campaigns), you'll need to edit privileges for the admin user.

It might be wise to create a special section in your own tree where you can put your subpublishers' trees, like this:

```
OASIS Server
|
+-- My Sections
|   |
|   +-- My subsection 1
|   +-- My subsection 2
|
+-- Subpub Sections
    |
    +-- Subpub section 1
    +-- Subpub section 2
```

Such an arrangement will allow you to easily place ad campaigns on all your subpublishers' sites, while still allowing you to easily place campaigns on just your sections. If you're using the OASIS server as an ASP (where subpublishers pay you to run their ads, and you do not share revenue on your campaigns), you definitely want such an arrangement so that your campaigns do not spill over onto their sections.

You can Suspend a publisher, and his campaigns will stop running on the server. Use this option if you have difficulty with a publisher but don't want to destroy all of his campaigns, sections, and advertisers on the off chance that the difficulties may be resolved. Also, you may still wish to invoice the publisher one more time. If you delete the publisher account, you will not be able to invoice.

After you suspend a publisher, you have the option to delete the publisher account, which will permanently remove the publisher, his sections, advertisers, and campaigns. This is irreversible, so exercise caution with this option.

Hosting Charges

You can charge your subpublishers a flat monthly fee for using your server. You can also charge based on impressions, clicks, and campaigns. These charges can be made on a flat rate basis or a percentage basis. If you set a flat rate, for example, \$1 CPM, your subpublisher will owe you \$1 for every 1000 impressions run by him or his subpublishers. If you set a percentage rate, for example, 50% of CPM, your subpublisher will owe you 50% of his CPM revenue (and any CPM-based revenue he gets from his subpublishers). You can also charge a combination of flat and percentage.

Network Payments

If you intend to use your server in an ad network where you sell campaigns to advertisers and run the campaigns on your subpublishers' sites, you can define how much they will be paid for running the campaigns here. These payments can be set as flat rates or percentages (or combinations of both). Payments "trickle down" the publisher hierarchy, so that if you pay your subpublisher P1 50% of CPM revenue, and P1 pays his subpublisher P2 50% of his CPM revenue, P2 will get 25% of the CPM revenue for any of your creatives that he runs.

Be sure you understand how the invoicing system works. Here are a few pointers:

OASIS 2.3 User's Manual

- Make sure it is cost-prohibitive for your subpublishers to abuse your server. If you are going to allow your subpublishers to serve their own campaigns through your server, you need to make sure that you set up charges that reflect your costs (and of course, you can make a profit if you like). If you do not charge something to your subpublishers, they can run your server into the ground with no consequences.
- Make sure you understand the implications of a "straight CPM" network payment. If you give a subpublisher \$0.50 CPM and run a house campaign that happens to run on his sections, you will end up paying him for the impressions. On the other hand, a percentage CPM would cost you nothing, assuming the campaign was set up with a \$0 CPM. Use the straight CPM very carefully.
- If you're running your server on an ASP model (ie., charging your publishers to run their own campaigns), be aware that an unscrupulous publisher may try to shield revenue from you. Make sure that you charge the publisher CPM, CPC, and Fixed. For example, if you only charge CPM, the publisher could sell his campaigns on a Fixed charge basis, and then you'll get no payments from the publisher.
- Another nasty trick is for a subpublisher to create a subpublisher of his own with no hosting charges applied. He can then run all campaigns through that bogus subpublisher, and since his "real" publisher will not get any hosting charges, you will not see any percentage-based payments. You can do a couple of things to prevent this sort of scenario:
 - ◆ disallow the creation of subpublishers by not granting the admin privilege to the publisher's users
 - ◆ charge your subpublisher on a "straight CPM" basis (CPC and Fixed, too, of course) instead of a percentage basis

Preferences

Each publisher is allowed to set up a group of preferences, which are intended to allow the publisher to brand his own look-and-feel and functionality of the publisher account and user interface, so that a seamless "private label" ad server can be created for the publisher's clients.

Click [HERE](#) to view detailed information about publisher preference settings.

Maximum Creative Sizes

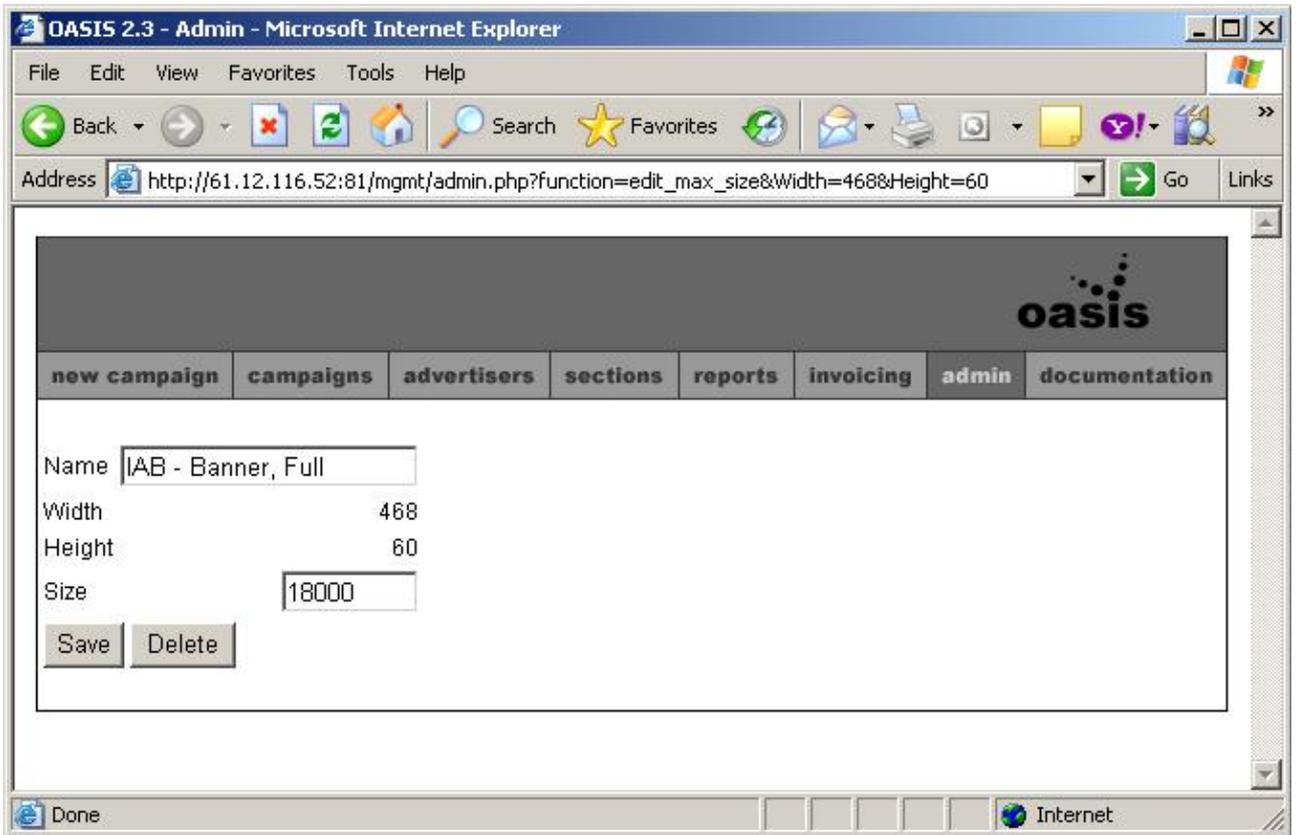
With this interface, you can control the size of uploaded creatives. For each dimension of creatives your site will serve (for example, 468 x 60), you define the maximum size in bytes for creatives of that dimension. Any time somebody tries to upload a creative larger than this maximum, it will be rejected. There is no override, short of temporarily changing the maximum using this interface.

Name	Width x Height	Size
IAB - Bar, Micro	88 x 31	2000
IAB - Button 2	120 x 60	4600
IAB - Button 1	120 x 90	7000
IAB - Banner, Vertical	120 x 240	18500
IAB - Skyscraper	120 x 600	27500
IAB - Button, Square	125 x 125	10000
IAB - Skyscraper, Wide	160 x 600	35000
IAB - Rectangle	180 x 150	18000
IAB - Banner, Half	234 x 60	9000
IAB - Rectangle, Vertical	240 x 400	50000
IAB - Pop-up, Square	250 x 250	37500
IAB - Rectangle, Medium	300 x 250	45000
IAB - Rectangle, Large	336 x 280	50000
IAB - Banner, Full	468 x 60	18000

Add new

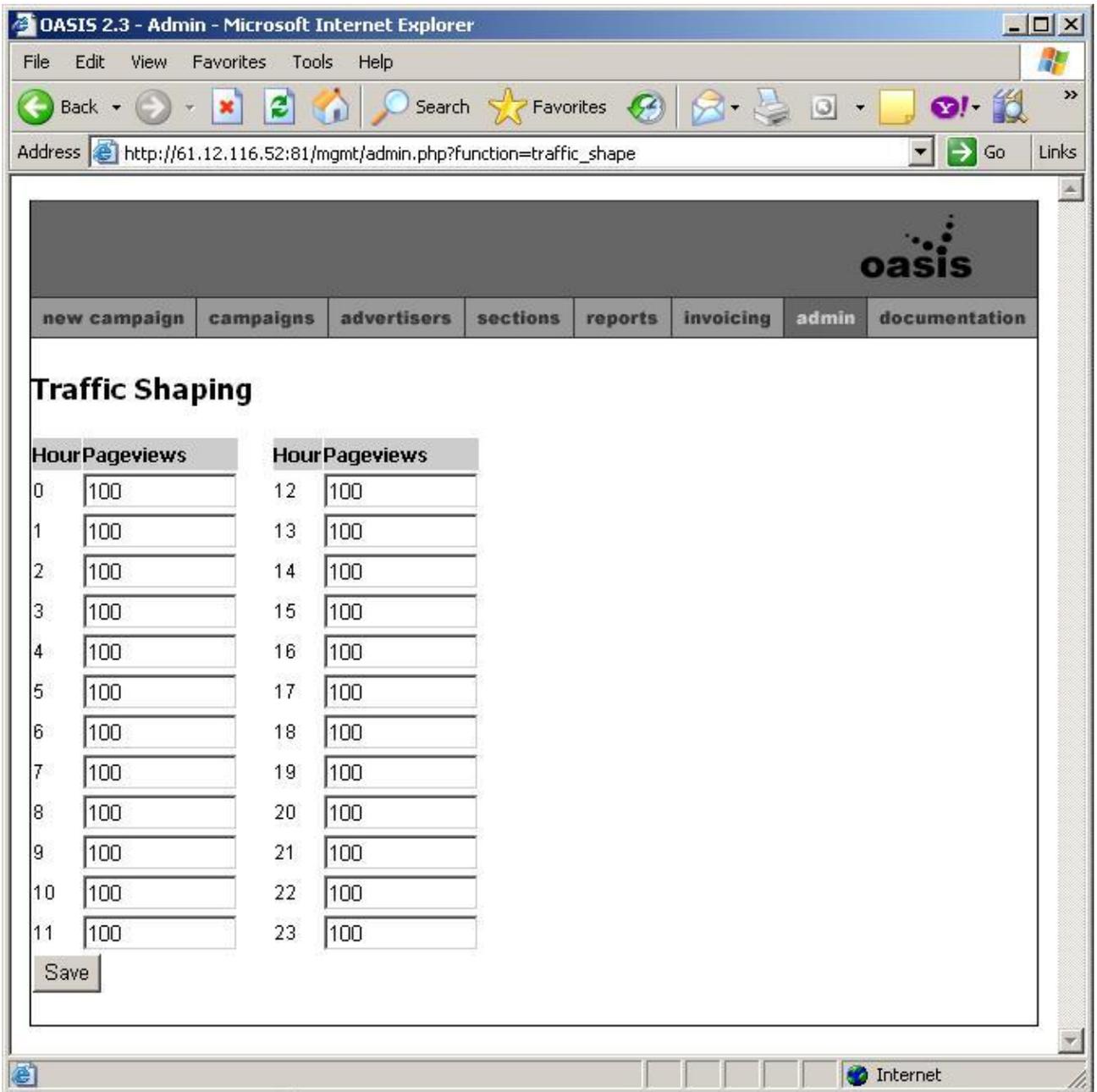
Here you see a list of all existing maximums. Click on "Add new" to set a new maximum. Click on the dimensions to redefine a maximum.

OASIS 2.3 User's Manual



You cannot edit the width and height once you've defined a maximum. You can, however, delete a maximum and define a new one. To change the maximum, enter the new size, and hit "Save".

Traffic Shaping



For each hour of the day, enter a value indicating the approximate number of pageviews your site does during that hour. These numbers are not used in an absolute fashion; rather, they are used simply to compute the percentage of traffic at each hour of the day for optimal scheduling.

Hit "Save" to save your changes.

View Admin Log

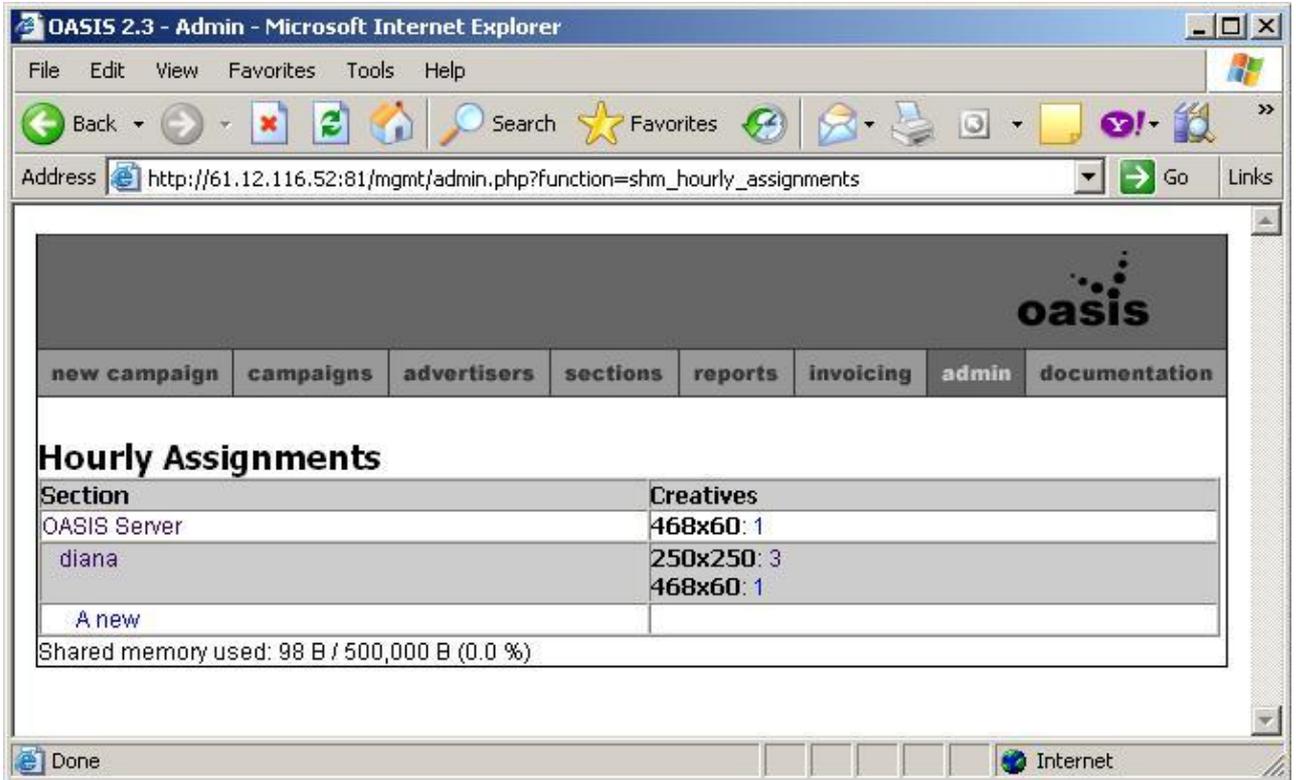
This provides a convenient way to view today's administrative log, which contains a lot of valuable information about how OASIS is scheduling your campaigns and creatives.

Delivery Engine Internals

Hourly Assignments

With this page, you can look at exactly which creatives are assigned to which sections on your site. This is the Hourly Assignments shared memory segment that is loaded by the hourly maintenance script.

The left-hand column displays the section names (along with links to the Section management interface for each section). The right hand column shows, for each creative dimension, an array of the creatives assigned to the section.



Daily/Hourly Targets

This table combines data from two sources, the DailyTarget table and the Hourly Targets shared memory segment. For each creative which is being scheduled today, you'll see the Target, Weight, and Remaining counts.

On the right-hand side of the table, you see the corresponding entries in the Hourly Targets shared memory. Notice that if you reload this page, you may see changes in the right-hand side. This is a live look into the delivery engine's internals.

Targets

Campaign/ Creative	Daily			Hourly			Clicks	Type
	Target	Weight	Remaining	Target	Weight	Remaining		
Diana campaign for cooldrinks / diana cooldrink creative	47	10	47	4	10	2	1	Image
OASIS Default / OASIS Default 1	0	1000	0	0	1000	0	0	Image

Impressions today: 2
 Impressions this hour: 2
 Clicks this hour: 1 (50.00 %)

Shared memory used: 136 B / 50,000 B (0.3 %)

Creative Content

This table displays the Creative Content shared memory segment. For each creative active this hour, you see the creative's name, the content (using `creative_preview.php` to display the contents of locally-served creatives), whether the ad is a third-party redirection, the MIME type, and whether the creative is animated.

You can click on any of the banners to get the full-size version.

Creative Clickthrough

This table displays the Creative Clickthrough shared memory segment. For each creative active this hour, you see the creative's name, and the clickthrough URL for that creative.

Note that for display purposes, the clickthrough URLs have been broken up to wrap better, so don't try to copy and paste them. However, the URLs are hyperlinked with the unmodified URL, so you can get the URL that way if you need it.

Delivery Table

In a live server, the Delivery table may be too big for practical viewing. In general, you can get a lot of valuable information just from the stats. The stats link will tell you how big the Delivery table is (how many bytes it occupies in RAM), the number of rows, the number of unique IDs, and the number of IP addresses in the table.

If you opt to view the entire table, then you will see, for each user ID (or IP), the most recently viewed creative for each section/dimension combination. You will also see when the creative was viewed.

Delivery Table
 Table Size: 129,060 (data) + 129,032 (index) = 258,092
 Rows: 4
 Unique IDs: 1
 Unique IPs: 1

Section	ID Count	IP Count
disinformation	1	1
disinformation: activism	1	1

Unique ID	Section	WxH	CreativeID	Last Seen
3ab943b95b99d	6	260x65	5	04/06/2001 17:50:05
	10	260x65	5	04/06/2001 17:50:05

IP Address	Section	WxH	CreativeID	Last Seen
216.77.243.168	6	260x65	5	04/06/2001 17:50:05
	10	260x65	5	04/06/2001 17:50:05

Reload Delivery Engine

This is an option that you will hopefully never have to use. It will force OASIS to recalculate all the section assignments and the hourly targets. Normally, if you add a new campaign (or change a campaign's status, impression goals, weights, etc.), the changes will show up in the Hourly Targets shared memory at the top of the next hour. If you absolutely can't wait until then, you can force OASIS to rebuild these shared memory segments immediately.

There are a couple of disadvantages to doing this. One is that the delivery of ads will pause while these shared memory segments are reloaded (usually not more than a second or two, though). To understand the second drawback, you have to understand how OASIS schedules campaigns. At the top of every hour, all campaigns with non-zero impression targets are given top priority until they meet their hourly targets. Then they stop delivering, and all campaigns with impression targets values of zero will run (in accordance to their relative weights). If you force a reload of the delivery engine, you will reset this process, and the non-zero campaigns will run again. They still will not exceed their daily targets, but if smooth delivery throughout the day is critical, you will not want to reload the delivery engine.

There is no confirmation required for this link -- if you click it, the delivery engine is immediately reloaded.

[Documentation Home](#)

Preferences

There are two sets of preferences that can be set in OASIS:

- **Top-level administrator user preferences** - The admin preferences apply to all publishers but can only be set by the OASIS system administrator. The admin also has access to all of the publisher preferences.
- **Publisher preferences** - Each publisher is allowed to set up a group of preferences, which are intended to allow the publisher to brand his own look-and-feel and functionality of the publisher account and user interface, so that a seamless "private label" ad server can be created for the publisher's clients. Whenever a new publisher is made, it will inherit its parent publisher's preference values.

The following is a list of the preferences available to publishers and their meanings. A full text box is provided for every preference, regardless of what types of values the software expects to find for these preferences. Be careful -- for fields that expect numbers, be sure you enter a number; for fields that expect "Y" or "N", be sure to enter one of these letters.

Admin Preferences

The following is a list of the Admin preferences and their meanings. A full text box is provided for every preference, regardless of what types of values the software expects to find for these preferences. Be careful -- for fields that expect numbers, be sure you enter a number; for fields that expect "Y" or "N", be sure to enter one of these letters.

AdminEmail

The e-mail address of the administrator. Used by `daily_maint.php` to send campaign underdelivery warnings. Default: `webmaster`.

ClickthroughWindow

The amount of time, in seconds, that entries can remain in the Delivery table. If an entry is deleted between the time an ad is shown to a user and the time he clicks on it, the clickthrough will result in an error. You don't want this value too small for that reason. However, you might not want it too big, either, as your Delivery table's memory consumption will grow. Default: `7200`.

CurlPath

The path to `curl` (used to retrieve remote URLs). PHP has some built-in mechanisms for retrieving URLs, but they don't work with redirection, which of course, is a common technique used in ad serving. If you install `curl`, enter the path to it here. If not, OASIS will use PHP's built-in HTTP code, but be warned that it may not work. Default: `/usr/bin/curl`.

GzipPath

The path to `gzip` (used to compress the archived log files). Default: `/bin/gzip`.

HtmldocPath

The path to `htmldoc`, the program that converts HTML invoices into PDF for batch printing. Default: `/usr/bin/htmldoc`.

IdentifyPath

The full path to `identify`, the ImageMagick tool used to determine image dimensions, type, animation, etc. Default: `/usr/X11R6/bin/identify`.

InvCycle

What kind of cycle does the daily traffic on your site follow? Your choices are "Day", "Week", "Month", and "Year". Note that your site's traffic may follow a daily cycle in its hourly traffic, but it may have a weekly cycle in its daily traffic. We're only interested in how the daily numbers vary.

If you set this to "Week", your predicted Monday traffic will be calculated based on historical data from Mondays. Tuesday traffic will be based on Tuesdays, and so on. If you set this to "Month", the 1st of the month will be calculated based on past 1sts. The same idea applies to "Year". If you set it to "Day", all days are treated the same.

Note that if you set this to "Week", it will take 7 days for you to have enough data to project future inventory. If you set it to "Month", it will take 30 days to project future inventory, and if you set it to "Year" -- you guessed it -- it will take 365 days to project future inventory. Default: `Week`.

InvDaysBack

How many days back should OASIS go to project future inventory? Default: `90`.

InvDaysForward

How many days out should OASIS project future inventory? Default: `60`.

InvMethod

What method should be used to project traffic? Currently, you can only set this to "Average". Default: `Average`.

InvScaleFactor

The delivery simulation used to predict inventory availability is very time-consuming. To speed it up, you can use this scale factor to simulate a fraction of the traffic. For example, if the traffic projection says you have 10,000 available impressions, and you have two campaigns each scheduled for 5,000 impressions, rather than simulating all 10,000 impressions, you could simulate 1,000 impressions with two campaigns of 500 impressions each.

The smaller you set this value, the faster your simulation will run, but the less accurate it will be. Default: `0.5`.

InvWarningThreshold

As the delivery simulation runs, it looks for section/day combinations where the allocated impressions seem high. This value determines what is considered "high". It represents a percentage of allocated impressions. For any day in the simulation, if any section's allocated impressions exceeds this value, a warning will be e-mailed to the administrator. Default: `80`.

KeepLogsFor

For how many days should we keep OASIS delivery logs? These files are quite large, and they're really not important unless you're trying to diagnose a problem with the server. Default: `14`

KeepStatsFor

For how many days should we keep OASIS stats? You want to keep these around for a pretty long time so that you can run reports, but you want to keep an eye on the size of the `HourlyStats` table. It gets pretty big. Default: `800`

LogDir

The directory in which log files are written by the delivery engine. This should be on as fast a disk as possible,

but one with a decent amount of storage, as log files are archived under this directory. Default:
`/usr/local/oasis/logs.`

OASISRoot

The URL to the OASIS directory. Used to construct sample tags in the section management interface, and used to e-mail the URL to the address specified in NewCampaignEmail. No default.

ShmSizeCreativeContent

Size (in bytes) of the shared memory segment containing the creative content for the server. See the [shared memory](#) section for information on how to tune this value. Default: 5,000,000.

ShmSizeCreativeClickthroughs

Size (in bytes) of the shared memory segment containing the creative clickthroughs for the server. See the [shared memory](#) section for information on how to tune this value. Default: 500,000.

ShmSizeHourlyAssignments

Size (in bytes) of the shared memory segment containing the hourly creative assignments for each section. See the [shared memory](#) section for information on how to tune this value. Default: 5,000,000.

ShmSizeHourlyTargets

Size (in bytes) of the shared memory segment containing the hourly targets for each creative. See the [shared memory](#) section for information on how to tune this value. Default: 50,000.

Publisher Preferences

The following is a list of the Publisher preferences and their meanings. A full text box is provided for every preference, regardless of what types of values the software expects to find for these preferences. Be careful -- for fields that expect numbers, be sure you enter a number; for fields that expect "Y" or "N", be sure to enter one of these letters.

InsertActive

Should campaigns added with the New Campaign interface be active? If this preference is set to "Y", all new campaigns made with the New Campaign wizard will be active, and no auto e-mail will be sent. If it is set to "N", campaigns will be added with a status of "Inactive", and an e-mail will be sent to the address specified in the NewCampaignEmail preference. This is intended to allow for the separation of sales and trafficking user roles. For example, a sales user can be given access to the New Campaign tab and the Reports tab; the salesperson can analyze reports and make new campaigns, and a trafficker will receive the new campaign email, edit the campaign and make it live.

InvoiceFooterString

Enter a block of text which will appear at the bottom of every invoice.

Invoice Due

Enter the number of days after which invoices must be paid. This is added to the InvoiceDate.

InvoiceHeaderGraphic

Enter the URL of the graphic file to be inserted at the top of every invoice.

InvoiceHeaderString

Enter a block of text which will appear at the top of every invoice.

NewCampaignEmail

Enter the e-mail address to which the system sends notifications of campaigns being added via the "New Campaign" interface. This is a feature to allow the separation of sales from trafficking. Presumably the salesperson will use the New Campaign wizard, an email will be sent to the trafficker, and then the trafficker can validate, edit and make the campaign active.

ReportFromAddr

Enter the e-mail "From:" address to be used when sending weekly reports to advertisers. It may include a "friendly form", for example: "OASIS Ad Operations ". This value *must* be specified, or reports may not get mailed at all.

RevenueReportClicks

Should clicks and CPCs be included on the revenue report? If you don't sell campaigns based on CPC, then these columns will be superfluous on the revenue reports. Set this to "N" to suppress the columns, and to "Y" to display them.

ValidateClickthrough

Should clickthrough URLs be validated when inserting creatives? If set to "N", the system will not check the clickthrough URLs. If set to "Y", the system will validate creative clickthroughs.

ValidateThirdParty

Should third party creative URLs be validated when inserting creatives? If set to "N", the system will not check the URLs. If set to "Y", third party creative URLs will be validated.

MainHeaderGraphic

Enter the URL of an image that you would like to be displayed in the ad server user interface header.

MainHeaderGraphicClickUrl

Enter the click-through URL of the ad server user interface header image.

CampaignReportInterval

Enter the number of days that a campaign will be available in the campaign report pick list after the campaign end date has been reached. This helps keep the list of campaigns available in the selection list for reporting from becoming too cluttered.

DefaultCampaignWeight

When a new campaign is created using the Add Campaign link or the New Campaign wizard, the campaign will have this campaign weight by default if the user leaves the campaign weight field blank during creation of the campaign.

[Documentation Home](#)

Adding OASIS Ads to Your Site's Pages

Method 1: IMG Tagging

The simplest way to incorporate OASIS banners on your site is to use IMG tagging. In this scenario, your OASIS server can be a standalone server separate from your Web server. The drawback is that you don't get to use ALT tags, and you can't serve rich media. In addition, you have to use cookies to record what banner a user last saw. This can be somewhat unreliable and can lead to erroneous clickthroughs.

In the pages of your Web site, you add tags like this:

```
<A HREF="http://oasis.yourdomain.com/oasisc.php?s=35&w=468&h=60"><IMG  
SRC="http://oasis.yourdomain.com/oasisi.php?s=35&w=468&h=60" WIDTH=468  
HEIGHT=60 BORDER=0></A>
```

There are two components, the impression URL and the clickthrough URL. The impression URL is `"/oasisi.php"` (on your OASIS server), and the clickthrough URL is `"/oasisc.php"` (also on your OASIS server).

Each takes three CGI arguments:

- `s`: the section number to which this page belongs
- `w`: the width of the banner to be displayed here
- `h`: the height of the banner to be displayed here

Note that it is *critical* that these three arguments match up in the impression URL and the clickthrough URL.

Cache Busting

If you're going to serve ads from OASIS using IMG tags, you really should consider adding a random number to the URLs you're using. OASIS does a lot of things to prevent caching of banners, but in order for those to have an effect, you've got to get your visitors to contact the server. Let's look at an example:

```
<A HREF="http://oasis.yourdomain.com/oasisc.php?s=35&w=468&h=60"><IMG  
SRC="http://oasis.yourdomain.com/oasisi.php?s=35&w=468&h=60" WIDTH=468  
HEIGHT=60 BORDER=0></A>
```

The first time a visitor hits this link, his browser will contact the OASIS server and get a banner ad. If the ad is animated (as so many banners tend to be), OASIS will not be able to send cache-busting HTTP headers (doing so causes bizarre looping of banners in many browsers). The second time the visitor's browser sees these tags, it may decide, "hey, I've got this in cache, so there's no need to contact this OASIS server". The visitor sees the same tired banner ad.

A solution is to put a random number into the URL for OASIS. You have to do this in a safe manner (you don't want the random number to interfere with the normal behavior of OASIS). The best way to do this is to create a new CGI variable (let's use `"cb"`) and tack it on to the OASIS URL:

```
<A  
HREF="http://oasis.yourdomain.com/oasisc.php?s=35&w=468&h=60&cb=123456"><IMG
```

```
SRC="http://oasis.yourdomain.com/oasisi.php?s=35&w=468&h=60&cb=123456"  
WIDTH=468 HEIGHT=60 BORDER=0></A>
```

There are different ways to add this random number. If you can afford the CPU cycles, you can use a program to generate a new random number each time a visitor loads a page (and using SSI to include it on the page). If your pages are dynamically generated, you can use your programming language's features to include the random number. You can have static SSI files that are rewritten periodically by a process running on your Web server. Or you can have a program rewrite the pages themselves. Use whatever method suits your site best.

Instance IDs

An instance ID helps to alleviate problems from the unreliability of cookies (remember that some users will have cookies disabled). It is a unique number that is added to both the `oasisi.php` and `oasisc.php` URLs in an impression/clickthrough pair.

Generally, the value is added dynamically via Javascript or server-side scripting. It should be unique (or at least picked randomly from a large enough range that collisions are unlikely).

Another benefit of IIDs is that you can use two banners of the same section and dimension on a single page as long as their IIDs are different. Without IIDs, OASIS would not be able to accurately deliver clickthroughs in such a scenario.

As a final advantage, the IID will help with cache busting (see above).

```
<A  
HREF="http://oasis.yourdomain.com/oasisc.php?s=35&w=468&h=60&iid=6879543"><IMG  
SRC="http://oasis.yourdomain.com/oasisi.php?s=35&w=468&h=60&iid=6879543"  
WIDTH=468 HEIGHT=60 BORDER=0></A>
```

Specifying the Creative ID

You can also use the `c` CGI parameter to specify the exact creative to run. If you do this, you must pass the `l` (lowercase "L") parameter as well to force the script to log the impression.

```
<A HREF="http://your.server.name/oasisc.php?c=27&s=52"><IMG  
SRC="http://your.server.name/oasisi.php?c=27&s=52&l=1" BORDER=0 ALT="My  
Creative"></A>
```

Note that if you link directly to creatives, you should not schedule the creatives for specific impression numbers, since the delivery engine will not have good control over how many times the creative gets delivered.

Method 2: IFRAMEs and ILAYERs

The IFRAME is an MSIE-only tag. It allows you to pull HTML content from another URL into your document. It doesn't actually incorporate the content directly into the document, but it does make it appear to be on the same page as the rest of the document.

OASIS 2.3 User's Manual

Since Netscape Navigator will not honor the IFRAME tag, you have to provide a backup tag. MSIE will ignore the backup tag and will only display the IFRAME content.

Here's an example:

```
<IFRAME SRC="http://oasis.yourdomain.com/oasisi-i.php?s=35&w=468&h=60"
MARGINWIDTH=0 MARGINHEIGHT=0 HSPACE=0 VSPACE=0 FRAMEBORDER=0 SCROLLING=NO
WIDTH=468 HEIGHT=60> <A
HREF="http://oasis.yourdomain.com/oasisc.php?s=35&w=468&h=60"><IMG
SRC="http://oasis.yourdomain.com/oasisi.php?s=35&w=468&h=60" WIDTH=468
HEIGHT=60 BORDER=0></A> </IFRAME>
```

MSIE will create a 468x60 IFRAME and fill it with content from

<http://oasis.yourdomain.com/oasisi-i.php>.

Navigator will ignore the IFRAME tag and display what comes between the <IFRAME> and the </IFRAME> tags. This is a nice hybrid approach between method 1 and method 3. You get most of the advantages of embedded ad serving (for users of MSIE), with the convenience of a standalone ad server.

You can expand upon this method with the ILAYER tag, a Netscape-only tag. In the following example, Netscape will use the ILAYER section and ignore the NOLAYER section. IE will use the IFRAME section inside the NOLAYER section. Other browsers will fall through to the IMG tag between the IFRAME tags.

```
<NOLAYER>
<IFRAME SRC="http://oasis.yourdomain.com/oasisi-i.php?s=8&w=468&h=60"
WIDTH=468 HEIGHT=60 FRAMEBORDER="no" BORDER=0 MARGINWIDTH=0
MARGINHEIGHT=0 SCROLLING="no"><A
HREF="http://oasis.yourdomain.com/oasisc.php?s=8&w=468&h=60"><IMG
SRC="http://oasis.yourdomain.com/oasisi.php?s=8&w=468&h=60"
BORDER=0 WIDTH=468 HEIGHT=60></A>
</IFRAME>
</NOLAYER>
<ILAYER ID="layer1" VISIBILITY="hidden" WIDTH=468 HEIGHT=60></ILAYER><P>
<LAYER
SRC="http://oasis.yourdomain.com/oasisi-i.php?s=8&w=468&h=60"
WIDTH=468 HEIGHT=60 VISIBILITY="hidden"
onLoad="moveToAbsolute(layer1.pageX,layer1.pageY);clip.height=60;clip.width=468
```

When using IFRAMEs with rich media, you must be sure that the code for your rich media includes TARGET="_top" (or TARGET="_new") in every hyperlink. Otherwise, the clickthrough URL will open inside the IFRAME itself.

Some users have reported problems with the ILAYER tag in some versions of Netscape Navigator 4.7x. For more discussion on the topic, see

https://sourceforge.net/forum/forum.php?thread_id=646644&forum_id=29988.

Method 3: Embedding OASIS Ads

OASIS 2.3 User's Manual

If you want to place OASIS ads on PHP-generated pages, and if OASIS is running on the same server you use to serve your site, you can use embedded PHP calls. This method has some real advantages:

- you can serve ALT tags in your image-based banners
- you can serve rich media in addition to image-based banners
- there is a greatly reduced chance of clickthrough errors, since the HTML generated by OASIS will encode the ID of the creative being served

To do this, you must put the following tags at the very top of your document (don't even put in any whitespace before this code):

```
<?php
require("/home/webdocs/oasis/oasisi-e.php");
$ad = get_creative(5, 468, 60, '');
?>
```

In the call to `get_creative()`, you pass the section number, the width and height of the desired creative, and an optional frame target.

Note: if you are using keyword targeting, you should put the keywords into the variable `$_REQUEST['k']` before calling `get_creative()`.

It's critical that the code be the very first thing in the document, since the code may need to spit out a cookie for the user. With PHP, any whitespace preceding this block of code will be output to the user *before* the code is executed, thereby preventing the cookie from being set.

Once you've made the call to put the creative into a variable, you can place it on the page in this way:

```
<?php echo $ad; ?>
```

If you have multiple sections on a single page (for example, "homepage left" and "homepage right"), you just need to get both ads into different variables in the first block of code:

```
<?php
require("/home/webdocs/oasis/oasisi-e.php");
$ad1 = get_creative(5, 120, 90, '');
$ad2 = get_creative(6, 120, 90, '');
?>
```

Then you can place the ads wherever you want with this:

```
<?php echo $ad1; ?>
```

and this:

```
<?php echo $ad2; ?>
```

Note that if you are using impression caps, you'll want to add the following code to hold back the delivery of the OASISCAP cookie until the last creative is selected.

Before making any calls to `get_creative()`, set `$OG_suppress_cap_cookie` to `true`. After the last call to `get_creative()`, set `$OG_suppress_cap_cookie` to `false`, then make a call to `set_cap_cookie()` to send the OASISCAP cookie to the client.

```
<?php
require("/home/webdocs/oasis/oasisi-e.php");
$OG_suppress_cap_cookie = true;
$ad1 = get_creative(5, 120, 90, '');
$ad2 = get_creative(6, 120, 90, '');
$OG_suppress_cap_cookie = false;
set_cap_cookie ();
?>
```

Note that even with this mechanism, OASIS will only set the cookie if changes were made to the impression counts of impression capped creatives. This saves a lot of unnecessary updates to the OASISCAP cookie.

Specifying a frame target will generate an <A HREF> tag with a "TARGET" modifier. This works best for Image banners, but it can also work for rich media -- when OASIS serves rich media, it modifies any "A HREF" in the rich media to insert a "TARGET" modifier. Here's a sample call to `get_creative()` that specifies a frame target:

```
<?php
require("/home/webdocs/oasis/oasisi-e.php");
$ad = get_creative(5, 120, 90, '_TOP');
?>
```

`get_creative()` can also accept an optional 5th argument that specifies a particular creative (useful if you want to explicitly specify the ad to run).

Method 4: Remote Javascript Invocation

Most modern browsers will allow your Web page to invoke a Javascript source file from another URL. You can use this to insert an OASIS banner into your page.

```
<SCRIPT LANGUAGE="JavaScript"
SRC=http://oasis.yourdomain.com/oasisi-j.php?s=1&w=468&h=60">
</SCRIPT>
<NOSCRIPT>
<A HREF="http://oasis.yourdomain.com/oasisc.php?s=1&w=468&h=60"><IMG
SRC="http://oasis.yourdomain.com/oasisi.php?s=1&w=468&h=60"
WIDTH=468 HEIGHT=60 BORDER=0></A>
</NOSCRIPT>
```

Note the NOSCRIPT section for browsers not supporting Javascript.

[Documentation Home](#)

Delivery Engine

Impression Script

`oasisi.php` is the script used to deliver an impression.

`oasisi.php` is responsible for delivering image banner impressions when you are using the IMG tagging delivery method. In this case, the URLs take the form

```
<IMG SRC="http://oasis.yourdomain.com/oasisi.php?s=5&w=468&h=60" BORDER=0  
HEIGHT=468 WIDTH=60>
```

You will note that in the tag, we specifying three things:

- section (CGI variable "s")
- width (CGI variable "w")
- height (CGI variable "h")

OASIS uses these three pieces of information to determine which creative to display to the user. It will search the list of creatives of the appropriate height and width which assigned to the section, and pick one based on a number of rules.

Before looking for candidate creatives, OASIS acquires a semaphore. Independent copies of the delivery engine must be able to scan the available creatives, select one, and update the "Remaining" field without trampling one another.

OASIS will first poll the Hourly Assignments shared memory segment for the list of all $w \times h$ creatives (of MediaType "Image") assigned to section s . It will then go through each of these and pull from the Hourly Targets shared memory segment the hourly impression target, weight, and number of remaining impressions needed to meet the hourly target.

Creatives with non-zero impression targets and zero remaining are ignored. These have already reached their hourly targets and should not run any more this hour.

If there are any creatives with non-zero impression targets where the number remaining is greater than 0, these get top priority. They are weighted according to the number of remaining impressions, and then one is randomly selected.

If there are no creatives with non-zero impression targets and more than zero remaining impressions, one of the creatives with a zero impression target is selected, according to the weights assigned to these creatives.

When OASIS randomly selects a creative, it basically adds up the weights of all candidate creatives (or in the case of non-zero-impression-target creatives, it uses the remaining impressions as "weights"); then selects a random number from 0 to the sum of the weights. It then loops through the candidates, decrementing the random number by each candidate's weight until the random number is less than zero. Whichever candidate causes the number to be less than zero is the creative served.

If for some reason, there is nothing of $w \times h$ assigned to section s (or if all have impression targets that have been met), the engine will not be able to select a creative to run. In such cases (and in any cases where the delivery engine encounters an error with shared memory or a semaphore), it will deliver a 1x1 invisible GIF

OASIS 2.3 User's Manual

(the actual GIF data is hard-coded into `oasisi.php`).

At this point, we have a creative ID. OASIS will now decrement the Remaining field for the creative (it is unfortunate that we can't do this later, after we've actually delivered the content to the user, but if we waited that long, we'd be holding the semaphore for too long, which would have a huge impact on performance). OASIS immediately records the delivery in `oasis.log` (in the directory 'LogDir', specified in the Preferences interface). It records 'imp' for a successful impression and 'ierr' when it has to deliver the 1x1 GIF. The shared memory and log are updated within the protection of the semaphore, and are done one after the other to reduce opportunities for an impression to get recorded in one but not the other. OASIS will now retrieve the actual content of the creative from the Creative Content shared memory segment. Here is pseudocode for the retrieval process:

```
if(creative is a third-party creative)
{
  ##### content is a text string representing the URL for the
  ##### third-party ad server

  replace "[CB]" strings in the URL with random number
  save the Location header (along with some cache control headers)
}
else
{
  ##### content is binary data for a GIF image

  ##### we can only print cache control headers for non-animated
  ##### GIFs (doing so with animated GIFs results in browsers looping
  ##### through a series of GIFs -- it's really wacky)
  if(creative is not animated)
  {
    save some cache control headers
  }

  save MIME type header
  save content
}
```

The Delivery table is now updated to record which creative was seen by the user on section s, so that if the user clicks through, he will be directed to the proper clickthrough URL. The delivery is recorded twice; once using the OASISID cookie number (either presented by the client or just now assigned to him) and once using the IP address. Although we hope that the user is accepting cookies, and we will be able to use the cookie to deliver the right clickthrough, we can't make that assumption, so we also have to record the IP address as a sort of backup. If we have to rely on the IP address, the clickthrough accuracy cannot be guaranteed, but it's better than nothing.

Finally, the headers and the content saved during the retrieval step are sent to the user, and the delivery engine is done.

Log Codes Here are the meanings of the various codes logged by the delivery engine.

Event	Code	Description
imp		successfully served impression
ierr	no_sem_1	couldn't acquire the semaphore used to determine whether the engine is in overflow mode
	no_sem_2	

	couldn't acquire the semaphore used to control access to the delivery tables
no_sec_cr	no active creatives for the specified section
no_secdim_cr	no active creatives of the right dimension for the specified section
no_ht_shm	unable to connect to the HourlyTargets shared memory segment (diagnostics may be available in parentheses)
secdim_cr_maxed_out	the creatives assigned to this section are maxed out (all have targets, and all have remaining = 0)
no_ht_shm_2	unable to connect to the HourlyTargets shared memory segment the second time, when we attempt to decrement the "remaining" count (diagnostics may be available in parentheses)
no_target_info	unable to retrieve target info from the HourlyTargets shared memory segment for the selected creative (diagnostics may be available in parentheses)
no_target_info_put	unable to put target info into the HourlyTargets shared memory segment for the selected creative (diagnostics may be available in parentheses)

These log files are picked up hourly by `hourly_maint.php`, which distills it down into a number of entries in the HourlyStats table.

Instance IDs

In addition to using cookies and IP addresses to link impressions to clickthroughs, `oasisi.php` will take a value passed in the `iid` CGI variable and store it in the delivery table, concatenated with the user's IP address. This can be referenced by `oasisc.php` as a unique identifier to determine the correct clickthrough.

Other uses for oasisi.php

If you are using the embedded tag method to display the ads on your site, `oasisi-e.php` is responsible for the bulk of the work described in this section. But to actually deliver an Image banner, it will call `oasisi.php`. The link generated by `oasisi-e.php` will take the form

```
<IMG SRC="http://oasis.yourdomain.com/oasisi.php?c=45" BORDER=0
HEIGHT=468 WIDTH=60>
```

This form specifies exactly which creative is to be delivered. `oasisi.php` doesn't have to do much work in this case; it simply has to deliver the contents of the designated creative.

It is not advisable to use links of this form directly in your site's pages. If you're not using `oasisi-e.php` to deliver the ads, specify the section, width, and height to `oasisi.php` and let OASIS make the decision of which creative to run.

[Documentation Home](#)

Embedded Impression Code

`oasisi-e.php` contains the code used to embed OASIS calls directly into your site's pages. To use this method, you must run OASIS on your Web server, and you can only embed the ads in PHP pages. On the

page where you want to embed an ad, you would have this:

```
<?php
require("/home/webdocs/oasis/oasisi-e.php");
get_creative(5, 468, 60, '');
?>
```

You will note that in the `get_creative()` function call, we specifying four things:

- section
- width
- height
- frame target (optional)

OASIS uses these three pieces of information to determine which creative to display to the user. It will search the list of creatives of the appropriate height and width which assigned to the section, and pick one based on a number of rules.

Before looking for candidate creatives, OASIS acquires a semaphore. Independent copies of the delivery engine must be able to scan the available creatives, select one, and update the "Remaining" field without trampling one another.

`get_creative()` uses the exact same algorithm for selecting a creative as is used in `oasisi.php`. The only difference is that `get_creative()` also considers creatives with `MediaType` of "RichMedia" (`oasisi.php` only considers those with `MediaType` of "Image").

Another difference between `get_creative()` and `oasisi.php` is that `get_creative()` does not use the Delivery table to record which creative was seen by each user. In the case of Image ads, the creative number is embedded right into the URL that `get_creative.php` generates for `oasisc.php`.

The following routines are shared between `oasisi.php` and `oasisi-e.php`: `get_client_id()`, `select_creative()` (with a minor change: it saves the `MediaType` of the selected creative to a global variable for later use), `record_delivery()`, and `log_msg()`.

If for some reason, there is nothing of `w x h` assigned to section `s` (or if all have impression targets that have been met), the engine will not be able to select a creative to run. In such cases, it will deliver an empty string.

At this point, we have a creative ID. If the selected creative has a `MediaType` of "Image", `get_creative()` will return the HTML to display that specific creative on the page (via a URL to `oasisi.php`), along with a cache-busting random number. If your call to `get_creative()` specified a frame target, the `<A>` tag will include a "TARGET=" modifier. If the creative has a `MediaType` of "RichMedia", `get_creative()` will return the actual content of the ad (which presumably is of an appropriate format to embed directly in the page). The frame target does not affect RichMedia creatives.

OASIS records the delivery in `oasis.log` (in the directory 'LogDir', specified in the Preferences interface). Note that locally-served Image ads are not recorded, since the subsequent request to `oasisi.php` will record the impression. It records 'imp' for a successful impression and 'ierr' when it has to deliver the 1x1 GIF. There are also a number of codes we can record:

Event	Code	Description
imp		successfully served impression

ierr	no_sem_1	couldn't acquire the semaphore used to determine whether the engine is in overflow mode
	no_sem_2	couldn't acquire the semaphore used to control access to the delivery tables
	no_sec_cr	no active creatives for the specified section
	no_secdim_cr	no active creatives of the right dimension for the specified section
	no_ht_shm	unable to connect to the HourlyTargets shared memory segment (diagnostics may be available in parentheses)
	secdim_cr_maxed_out	the creatives assigned to this section are maxed out (all have targets, and all have remaining = 0)
	no_ht_shm_2	unable to connect to the HourlyTargets shared memory segment the second time, when we attempt to decrement the "remaining" count (diagnostics may be available in parentheses)
	no_target_info	unable to retrieve target info from the HourlyTargets shared memory segment for the selected creative (diagnostics may be available in parentheses)
	no_target_info_put	unable to put target info into the HourlyTargets shared memory segment for the selected creative (diagnostics may be available in parentheses)

Note that each embedded ad makes two calls to the OASIS delivery engine, the first when your PHP calls `get_creative()` and the second when your browser requests `oasisi.php`. You don't have to worry about these requests getting logged twice, though, because when `oasisi.php` is called with a specific creative number, it does not log the request, knowing that `get_creative()` already did the logging.

The log files are picked up hourly by `hourly_maint.php`, which distills it down into a number of entries in the HourlyStats table.

[Documentation Home](#)

Iframe Impression Code

`oasisi-i.php` contains the code used to deliver content to IFRAMES (an MSIE-only feature that allows you to embed the content from another URL more or less directly in your page).

This code is *exactly* like `oasisi-e.php`, except for a single line of code:

```
print get_creative($s, $w, $h, '_top');
```

A link to `oasisi-i.php` will spit out HTML suitable for use in an IFRAME.

We could have certainly implemented this by require-ing `oasisi-e.php`, but for performance reasons, it is better just to replicate the code in both files.

[Documentation Home](#)

Javascript Impression Code

`oasisi-j.php` contains the code used to deliver content as Javascript source, feature that allows you to embed the banner directly into the HTML of your page (for browsers that support remote Javascript invocation).

This code is *exactly* like `oasisi-i.php`, except for a call to `javafy()`, which turns the HTML for the creative into Javascript.

A link to `oasisi-j.php` will spit out Javascript which writes HTML to the current document.

Some shared code between the delivery scripts would be advantageous from a code maintenance standpoint, but for performance reasons, it is better just to replicate code in each script.

[Documentation Home](#)

Clickthrough Script

`oasisc.php` is the script used to deliver a clickthrough. It is called in an A HREF tag. For example:

```
<A HREF="http://oasis.yourdomain.com/oasisc.php?s=5&w=468&h=60">
```

You will note that in the tag, you are specifying three things:

- section (CGI variable "s")
- width (CGI variable "w")
- height (CGI variable "h")

OASIS uses these three pieces of information to look up in the Delivery table which creative of dimension w x h was last seen by the user on section s.

OASIS looks for a url (which can be passed in the `url` CGI variable, usually as part of a rich media creative). If found, it redirects to that URL.

Next it looks for a creative ID. Embedded deliveries generate a link to `oasisc.php` that includes the creative ID, eliminating any ambiguity about what creative was last seen. If found, the clickthrough for this creative is delivered.

If neither a URL or creative ID is found, OASIS will try to match the clickthrough to a record in the Delivery table.

If provided, OASIS uses the `iid` CGI variable as a unique identifier for the impression. If sites use this variable, they must take care to use the same value for `iid` in each `oasisi.php/oasisc.php` pair. If no `iid` is found, OASIS tries to obtain the user's unique identifier from the OASISID cookie.

If a unique identifier is obtained, OASIS will attempt to look up in the Delivery table the w x h creative which was last seen by the user on section s. If it finds it, it will deliver the clickthrough.

If no unique ID was obtained (or if no cookie was obtained), OASIS will look in the Delivery table for the w x h creative which was last seen by somebody at the user's IP address (hopefully this is the user himself, but

OASIS 2.3 User's Manual

note that it is extremely unreliable in this day and age of NAT-ed firewalls) on section s.

If we did not get a creative ID, we deliver an empty Location string, which generally reloads the user's current page. This behavior might be something worth changing in the future so that the user gets some sort of error message.

If we `_did_` get a creative ID, we have to note it in the Hourly Targets shared memory segment (which of course, we do within the protection of a semaphore so that we don't step on any other copies of `oasisi.php` or `oasisc.php`).

Now we grab the clickthrough URL from the Creative Clickthrough shared memory segment and print out a Location header (along with some cache control headers) to redirect the user to the clickthrough destination.

Finally, we log the clickthrough. We can record either "click" or "cerr" as the event. There are also a number of codes we can record:

Event	Code	Description
click	success	normal delivery
	fallthru	got a cookie, but had to fall through to the IP address
	nocookie	didn't get a cookie; had to use IP address
cerr	no_cr_found	couldn't find a CreativeID in the Delivery table
	no_sem_1	couldn't acquire the semaphore used to determine whether the engine is in overflow mode
	no_sem_2	couldn't acquire the semaphore used to control access to the delivery tables
	no_ht_array	couldn't find the hourly targets in shared memory (diagnostics may be available in parentheses)
	no_ct_array	couldn't find the clickthrough URL in shared memory

[Documentation Home](#)

Maintenance Scripts

Daily Maintenance

Once per day (at hour 0), `hourly_maint.php` runs daily maintenance. This used to be implemented in a separate script; it is now implemented within `hourly_maint.php`

Check for underdelivery

The script looks through all the creatives in the `DailyTargets` table (remember that this script only runs at midnight, and thus it is looking at the daily targets for yesterday, *after* the hour 23 results have been tabulated by `hourly_maint.php`). For any creatives which have specific impression targets, it checks the number remaining. If that number is greater than 0, mail will be sent to the administrator.

The administrator's email is in the `OASISPrefs` table as "AdminEmail".

Compute daily targets

Under the protection of a lock, the `DailyTargets` table is emptied and reloaded. The creatives' targets are calculated in a specific order. First, all campaigns with `ImpressionsGuaranteed - ImpressionsDelivered > 0` are scanned. A daily target for the campaign is generated, and then this target is split among the campaign's creatives. Creatives with non-zero `ImpressionsGuaranteed` get an appropriate number of impressions to meet their targets (unless that number would exceed those allotted to the campaign, in which case, the creative only gets the number allotted to the campaign). Creatives which have `ImpressionsGuaranteed = 0` get a share of the Campaign's impressions proportional to their `Weight`.

Finally, campaigns with no impression targets are scanned. Their creatives are inserted into the `DailyTargets` table with `Target = 0`, and

$$\text{Weight} = \text{int}(\text{Campaigns.Weight} * (\text{Creatives.Weight} / \text{Sum of all campaign creatives' weights}) * 100)$$

This multiplication preserves the relative weights of creatives within a campaign and the relative weights of campaigns to each other.

This might be a no-brainer, but it is important to note that as campaigns and creatives are scheduled, only those scheduled to run on today's day of the week will be scheduled. For example, if the script is running on a Monday, only campaigns and creatives slated to run on Mondays will be examined.

Load shared memory

Finally, the creatives which are scheduled to run today are loaded into shared memory. Here is where the semaphore acquired by `hourly_maint.php` is really important. The Creative Content and Creative Clickthrough shared memory segments are wiped clean (thus, if the delivery engine were running, it would experience severe trauma). For each creative in the `HourlyTargets` table, the `Contents`, `ClickthroughURL`, and other relevant fields (`Redirect`, `MIMETYPE`, `Animated`) are loaded into the two shared memory segments.

[Documentation Home](#)

Hourly Maintenance

`hourly_maint.php` is run at the top of each hour. It reloads the shared memory segments that determine which creatives are assigned to which sections and how many of each to run.

Step-by-step through `hourly_maint.php`

This section will illustrate the various tasks performed by this script.

Record the last hour's traffic

Immediately upon launching, a semaphore is obtained (0x4f415300, or "OAS0" in ASCII) to prevent new data from being written to or read from the shared memory segments while we're performing our calculations. We want to hold this semaphore for as short a period as possible, since all ad delivery is paused while we're holding it.

Record the last hour's traffic

All of the impressions and clickthroughs are committed back to the database. The ImpressionsDelivered and ClickthroughsDelivered fields are updated in the Creatives table. Later, this information will be propagated up to the level of the campaigns, but for now, since we want to operate as fast as possible while holding the semaphore, we just record things at the creative level.

Call `daily_maint.php` if appropriate

If we're called during hour 0 (or if the `force_daily` command-line option is specified), we'll call the daily maintenance script, which will rebuild the DailyTargets table.

Build the assignment table

The script goes through all the campaign and creative section assignments and propagates them down through the section tree. It processes them in this order:

1. Campaign Includes
2. Creative Includes
3. Campaign Excludes
4. Creative Excludes
5. Campaign Exclusives
6. Creative Exclusives

In this way, creative assignments always override those of their parent campaigns. Also, exclusive assignments override all others.

A large table is constructed. For each section *S* in the section tree, there is a hash. The keys to this hash are creative dimensions ('120x90', '468x60', etc.). The value associated with each key *D* is an array of CreativeIDs for all the creatives of dimensions *D* which are running on section *S*. The delivery engine will use this table to quickly acquire a list of all candidate creatives for a given section.

Compute hourly targets

For each entry in the DailyTargets table, the script looks at how many hours are left in the day, how much traffic is slated to run during those hours (from the values you've defined in the Traffic Shaping interface) and what percentage of that traffic is remaining in the current hour. This is done on a per-creative level, since each creative can run at different hours during the day.

Example: suppose it is hour 22. You have said that you get on average 10,000 pageviews in hour 22 and 5,000 pageviews in hour 23. You have two active creatives. Creative 1 runs all hours of the day. Creative 2 runs hours 10 a.m. to 10 p.m. Each still needs 600 impressions before the end of the day. Creative 1 will be scheduled for

$$600 * 10,000 / (10,000 + 5,000) = 400 \text{ impressions}$$

Creative 2, which does not run in hour 23, will be scheduled for

$$600 * 10,000 / (10,000) = 600 \text{ impressions}$$

Creatives which do not have specific impression targets will be entered into the HourlyTargets shared memory if they are slated to run during the given hour.

For each active creative, four values are stored in the HourlyTargets shared memory segment: the hourly impression target, the weight of the creative, the number of impressions remaining, and the number of clicks delivered.

For creatives with impression targets, the number of impressions remaining is equal to the hourly impression target when the table is loaded. As impressions are delivered, this number decrements until it hits 0, at which point no more impressions are served.

For creatives without impression targets, the number of impressions remaining is 0 when the table is loaded. As impressions are delivered, this number goes negative.

Release the semaphore

We've done all the really tricky stuff; so it's now safe to allow delivery to resume.

Process log file

This step happens only if no options were specified on the command line (presumably, if you're calling the script by hand, you're specifying options. When you call the script by hand, you don't want to process the logs -- we try to process them only once per hour to reduce the entries in the HourlyStats table).

The log file is slurped up and each entry is tallied. The result is that for each creative/section combination, a record is saved in the HourlyStats. Impressions, clicks, impression errors, and click errors are stored in the table. Reports are generated on the fly from these records.

In addition to the fairly detailed HourlyStats table, there is a table called CampaignDailyStats, which contains one record per campaign per day. This table exists for speed in generating invoices and the revenue report. You can't slice and dice the numbers like the HourlyStats, but it is a much more compact representation.

The contents of the log file are appended to the file `YYYY/MM-DD.log` in the directory specified by the `LogDir` preference. During hour 0, yesterday's log file is gzipped. Note that you can clean these files up at will; they are only there for your reference.

Update campaigns

Now we propagate the impressions and clicks recorded for each creative up to their parent campaigns.

Mark campaigns/creatives complete

Finally, we look for any campaigns or creatives which have either met their impression targets or have passed their end dates. At this point, we set the Status of those campaigns to "Complete".

Inventory Simulation

If it's currently hour 0 (and we're being run as a regular cron job, not manually), project available inventory and run a simulation. This is a compute-intensive job. It can take a couple of hours, depending on some of your preferences.

Adding `hourly_maint.php` to the crontab

Add this script to the Web server user's crontab. Do not run this script as anybody but that user, as this will result in creating shared memory segments that are not owned by the Web server user. This will interfere with normal operation of OASIS, as it will not be able to clear shared memory segments.

The job should run at the top of every hour, as so:

```
0 * * * * /path/to/oasis/mgmt/hourly_maint.php > /dev/null
```

Calling `hourly_maint.php` from the command-line

Under certain circumstances, you can run this script from the command-line. Do not run it as any other user but the one whose crontab runs the script normally (preferably your Web server user). You should call it in one of these ways:

```
/path/to/oasis/mgmt/hourly_maint.php start
```

This will force the running of `daily_maint.php`, and it will reload the hourly targets. It will not process the logs or attempt to record any traffic from the previous hour. Use this to start up OASIS after a reboot (and before you start your Web server).

```
/path/to/oasis/mgmt/hourly_maint.php force_daily
```

This is the same as "start", but it also records the last hour's traffic.

```
/path/to/oasis/mgmt/hourly_maint.php reload
```

This is basically the same as calling `hourly_maint.php` with no arguments, but it won't call the daily maintenance script (even if it is run during hour 0), and it won't run the inventory simulation (even if it is run during hour 0). It will record the last hour's traffic, process the logs, and reload the hourly targets.

OASIS 2.3 User's Manual

```
/path/to/oasis/mgmt/hourly_maint.php stop
```

This method of invocation will run the script and record the last hour's traffic. It then clears shared memory and processes the log. It does not attempt to calculate any targets for the next hour. In fact, if `oasisi.php` runs after the shutdown process, it will not be able to deliver anything.

You should run this immediately after shutting down your Web server. Note that you can safely stop your Web server at any time without doing anything to OASIS. But if you want to reboot the server, you should run 'stop' to record the last hour's data to disk first.

```
/path/to/oasis/mgmt/hourly_maint.php resume
```

This method of invocation will not attempt to process logs or record the last hour's traffic. It will not try to clear shared memory. All it will do is reload shared memory. Use this after you've run 'stop', before restarting your Web server.

The difference between this method and the 'start' method is that 'start' will force a recalculation of daily targets. If you've been running and you stop the server only to decide to restart it, you can use 'resume'.

[Documentation Home](#)

Minutely Maintenance

`minutely_maint.php` is run every 5 minutes (or as frequently as every minute, if you can afford the cycles and the hits to your database server).

The only real function of this script is to clean up the Delivery table. It uses the `ClickthroughWindow` preference to determine how long (in seconds) it should let entries live in the delivery table. You should set this preference to as large a number as you think your system can handle. The larger the number, the longer the entries stay in the table, but the more memory the Delivery table will take up (it is a heap table, stored completely in RAM).

To understand the implications of setting the `ClickthroughWindow` to a small value, use the following example. A user loads one of your site's pages, and along with it, gets an ad banner. The user then waits 12 minutes before clicking on the banner. If you have set your `ClickthroughWindow` to 600 seconds, the user will get a clickthrough error. If you have set your `ClickthroughWindow` to 900 seconds, the user will be redirected to the advertiser's URL.

Note that your delivery table will have to be able to accommodate the number of entries stored in (`ClickthroughWindow` + `cron_interval`) seconds, where `cron_interval` is the number of seconds between cron invocations of `minutely_maint.php`. So if you've set the `ClickthroughWindow` to 900 seconds, and you run `minutely_maint.php` every 5 minutes, you'll have to be able to accommodate 20 minutes' worth of entries in the delivery table.

Adding `minutely_maint.php` to the crontab

Add this script to the Web server user's crontab. If you want to run it every 5 minutes, add this entry to the crontab:

```
5,10,15,20,25,30,35,40,45,50,55 * * * * /home/webdocs/oasis/mgmt/minutely_maint.php > /dev/null
```

[Documentation Home](#)

Log files

OASIS generates two types of logs, those from the delivery engine that record each delivered impression and clickthrough, and those from the administrative tools.

Delivery Engine Logs

The live delivery engine log is written to the file `oasis.log` in the directory specified in the "LogDir" record in the OASISPrefs table. When `hourly_maint.php` runs, it reads in the contents of the file, processes the entries, and tacks them onto the daily log file, stored in the `YYYY/MM-DD.log` file under the LogDir directory.

Each night, the file is gzipped (the location of gzip is in the "GzipPath" record in the OASISPrefs table. These files are not used by the software. They exist merely to provide a record of OASIS's activity.

For more information on the meaning of some of the error codes logged by OASIS, see the Delivery Engine section, specifically, the sections on the individual scripts.

Administrative Logs

Administrative information is logged to `oasis_admin.log` in the LogDir directory. Each time the DailyTargets table is updated and each time the HourlyTargets table is cleared and reloaded, entries are made into this log. This log is purely for diagnostic purposes. It can, for example, help you figure out why the DailyTarget information for a campaign has been altered (each time you save a campaign or creative, the campaign's DailyTarget entries are recomputed and this is logged here).

[Documentation Home](#)

Shared Memory

OASIS utilizes a number of shared memory segments to facilitate rapid ad delivery and to cut down the number of queries made against the database.

Hourly Assignments

SHM ID: 0x4f415300 (OAS0)
0x4f415305 (OAS5), overflow

Read by: `oasisi.php`

Written by: `hourly_maint.php`

Size: 500 KB

Variables are hashed into the shared memory segment based on section number.

Each variable is a hash keyed by dimensions (WxH)

Each hash value is an array of creatives of the correct dimensions currently assigned to the section.

The size needed for this segment is dependent on the number of sections you have and the number of creatives.

A good formula to use is $11 * (\text{sections} * \text{creatives})$ where sections is the number of sections that have active creatives assigned to them, and creatives is the total number of active creatives. This is overkill, unless you have all creatives assigned to all sections, but it gives you a good safety margin.

In testing, 1024 creatives (of 8 different dimensions), all assigned to 128 sections required 1.37 MB of storage. This, of course, is a worst case scenario -- it assumes that you have 1024 active creatives (pretty unlikely), and that each and every one of those creatives is assigned to every single section of your site (again, very unlikely with a large site).

In the Admin interface, click on "Hourly Assignments" to see the contents of this segment. At the bottom, you can see how much of the allocated space is being used. If you think you need more, you can adjust the `ShmSizeHourlyAssignments` preference value.

Note that while the delivery engine is being reloaded, delivery takes place from the overflow tables; only creatives for which exact impression numbers are not important are listed in these tables. In this way, the pause in delivery is nearly eliminated.

Hourly Targets

SHM ID: 0x4f415301 (OAS1)
0x4f415306 (OAS6), overflow

Read by: `oasisi.php`
`oasisc.php`

Written by: `hourly_maint.php`
`oasisi.php`
`oasisc.php`

Size: 50 KB

Variables are hashed into the shared memory segment based on creative number.

Each variable is an array of the following values: (target, weight, remaining, clicks, media type)

It may seem a bit strange to use "remaining", since it often results in negative numbers, but this allows us to more quickly identify creatives that `_need_` impressions during the hour. Any creative with a remaining value > 0 has to get impressions. If we kept an incrementing counter instead, we'd have to perform a subtraction of the delivered number from the target number to determine who needs impressions. This is a slight inefficiency. I will admit, however, that the optimization of using a decrementing counter may not even make a significant difference, given all the other calculations that are going on.

The size needed for this segment is proportional to the maximum number of active creatives you have at any one time. It requires approximately 64 bytes for each active creative. 50 KB would allow for 780 active creatives in the system on any given day.

Note that while the delivery engine is being reloaded, delivery takes place from the overflow tables; only creatives for which exact impression numbers are not important are listed in these tables. In this way, the pause in delivery is nearly eliminated.

Creative Content

SHM ID: 0x4f415302 (OAS2)
0x4f415307 (OAS7), overflow

Read by: `oasisi.php`

Written by: `daily_maint.php`

Size: 4 MB

Variables are hashed into the shared memory segment based on creative number

Each variable is an array of the following values (Content, Redirect, MIMEType, Animated)

The amount of space required is equal to the number of active creatives * (Average Size + 85 Bytes). So if you expect your ad content to take up 15KB each, then you'll need a little bit more than 15 MB to hold 1000 active creatives. Keep in mind that creatives that use third-party ad delivery require considerably less storage.

In the Admin interface, click on "Creative Content" to see the contents of this segment. At the bottom, you can see how much of the allocated space is being used. If you think you need more, you can adjust the `ShmSizeCreativeContent` preference value.

Note for FreeBSD users:

In the FreeBSD default GENERIC kernel, there is maximum shared memory segment size of 4194304 (4MB).

If you want bigger `ShmSizeCreativeContent`, you must recompile kernel with this options:

```
options SHMMAXPGS=2049
options SHMMAX="(SHMMAXPGS*PAGE_SIZE+1)"
```

This sets maximum shared memory to 8MB. If you want 16MB, you must use 4097 and so on.

Creative Clickthrough

SHM ID: 0x4f415303 (OAS3)
0x4f415308 (OAS8), overflow

Read by: `oasisc.php`

Written by: `daily_maint.php`

Size: 500 KB

Variables are hashed into the shared memory segment based on creative number

Each variable is a string containing the clickthrough URL for the creative.

The size required for this segment is proportional to the maximum number of active creatives. It is roughly equal to the number of active creatives * 276 Bytes

This is a worst-case number; it assumes that each active creative has a 255-character clickthrough URL (the largest that can be stored in the database table).

A 300 KB shared memory segment can therefore store well over 1000 active creatives.

In the Admin interface, click on "Creative Clickthrough" to see the contents of this segment. At the bottom, you can see how much of the allocated space is being used. If you think you need more, you can adjust the `ShmSizeCreativeClickthroughs` preference value.

Deliveries

The original design of OASIS called for a Deliveries shared memory segment, where we would record the creatives seen by each user for future clickthroughs. Unfortunately, PHP's shared memory manager is too primitive to handle the rapid insertion of many thousands of keys.

Instead, I've opted to use a MySQL heap table to store the delivery information. The insertions are reasonably fast, and having an indexed table allows for much faster lookups.

Overflow flag

SHM ID: 0x4f415304 (OAS4)

Read by: `oasis?.php`

Written by: `hourly_maint.php`

Size: 100 B

This tiny shared memory segment indicates whether the delivery engine is in its overflow mode. It flips into overflow at the top of the hour while it reads the primary delivery table values, recalculates the next hour's numbers, and reloads the primary delivery tables.

During overflow mode, alternate assignment and target shared memory segments are used so that delivery can continue. Creatives scheduled in these tables are allowed to "overflow", meaning that they do not have specific impression goals, and have been designated as "Overflow OK". This allows delivery to continue without interruption during hourly scheduling calculations.

[Documentation Home](#)

Database Schema

This section details the schema of the various tables used in OASIS.

This section is out-of-date.

[Documentation Home](#)

Publishers Table

Publishers are arranged in a hierarchy; each has its own section tree rooted at a specific location in the server's overall section tree.

Column	Type	Default	Extra	Comment
PK PublisherID	int(11)		auto_increment	
PK PPublisherID	int(11)			ID of the publisher's parent publisher
Name	varchar(128)			
StreetAddress	text			
City	varchar(64)			
State	varchar(64)			
PostalCode	varchar(32)			
Country	varchar(64)			
Phone	varchar(64)			
ContactName	varchar(128)			
ContactEmail	varchar(255)			
HostingCharge	float(10,2)			Amount charged per invoice period by parent publisher
CPMCharge	float(10,2)			Flat CPM charged by parent publisher
CPMChargePerc	float(10,2)			Percentage of publisher's CPM revenues charged by parent publisher
CPCCharge	float(10,2)			Flat CPC charged by parent publisher
CPCChargePerc	float(10,2)			Percentage of publisher's CPC revenues charged by parent publisher
FixedCharge	float(10,2)			Flat fixed charge charged by parent publisher for each campaign
FixedChargePerc	float(10,2)			Percentage of publisher's fixed charge campaign revenues charged by parent publisher
CPM	float(10,2)			Flat CPM paid to publisher by parent publisher
CPMPerc	float(10,2)			Percentage of parent publisher's CPM revenues paid to publisher by parent publisher
CPC	float(10,2)			Flat CPC paid to publisher by parent publisher
CPCPerc	float(10,2)			Percentage of parent publisher's CPC revenues paid to publisher by parent publisher
Fixed	float(10,2)			Flat fixed charge paid by parent publisher for each campaign
FixedPerc	float(10,2)			Percentage of parent publisher's fixed charge campaign revenues paid to publisher
Status	enum('Active','Inactive')			
RootSectionID	smallint(6)			ID of the section that is the root of this publisher's section tree

[Documentation Home](#)

Advertisers Table

Column	Type	Default	Extra	Comment
AdvertiserID	int(11)		auto_increment	
Name	varchar(128)			
URL	varchar(255)			
ContactName	varchar(128)			
StreetAddress	text			
City	varchar(64)			
State	varchar(64)			
PostalCode	varchar(32)			
Country	varchar(64)			
Phone	varchar(64)			
Fax	varchar(64)			
ContactEmail	varchar(255)			
Note	text			
Login	varchar(32)			
Password	varchar(32)			
PublisherID	int(11)			The ID of the publisher to which this advertiser belongs

[Documentation Home](#)

Campaigns Table

This table stores all the information about campaigns.

Note that there is an analogous table, SimCampaigns, that is used during the inventory simulation.

Column	Type	Default	Extra	Comment
🔑 CampaignID	int(11)		auto_increment	
🔑 AdvertiserID	int(11)			
Name	varchar(128)			
🔑 Status	enum('Active', 'Cancelled', 'Completed', 'Inactive', 'Suspended')	Inactive		By default, new campaigns are given a status of "Inactive". Only "Active" campaigns have their creatives scheduled. When a campaign meets its impression targets or its EndDate, it will be marked as "Complete". You can manually set a campaign to "Suspended" or "Cancelled". Campaigns with Status="Completed" or "Cancelled" do not show up in the list of campaigns.
🔑 StartDate	date	0000-00-00		
🔑 EndDate	date	0000-00-00		
OverflowOK	enum('Y', 'N')	Y		Indicates whether the campaign's creatives can be delivered during the overflow period at each hourly reload.
CompanionSharing	enum('Y', 'N')	N		Indicates whether the campaign's creatives can be delivered simultaneously on companion sections.
ImpressionsGuaranteed	int(11)			The number of impressions to be delivered for the campaign. If this value is set to 0, the campaign will be treated as a "house" campaign, and will run only after campaigns which have non-zero values have fulfilled their hourly targets.
EvenDelivery	enum('Day', 'Week', 'Month')	Day		If ImpressionsGuaranteed > 0, how are the impressions to be scheduled? Should the same amount be delivered each day, each week (or fraction of a week), or each month (or fraction of a month)?
ImpressionsDelivered	int(11)			Each hour, the hourly maintenance script will update this value based on the number of impressions in the

OASIS 2.3 User's Manual

				previous hour.
ClicksGuaranteed	int(11)			Number of clicks at which the delivery engine will stop scheduling the campaign's creatives.
ClicksDelivered	int(11)			Each hour, the hourly maintenance script will update this value based on the number of clicks in the previous hour.
DaysOfWeek	tinyint(3) unsigned	127		A bitmask indicating which days of the week the campaign should run; low bit is Sunday, high bit is Saturday.
HoursOfDay	mediumint(8) unsigned	16777215		A bitmask indicating which hours of the day the campaign should run; low bit is hour 0, high bit is hour 23.
Weight	int(11)	1		The weight only comes into play if ImpressionsGuaranteed=0. All campaigns with ImpressionsGuaranteed=0 will be served in numbers proportional to their Weight values.
ImpCap	int(11)	0		Per-visitor impression cap for the campaign.
ImpCapInterval	int(11)	0		Interval over which the impression cap should not be exceeded (in seconds).
Note	text			
CPM	float(10,2)	0.00		Cost per 1000 impressions for the campaign.
CPC	float(10,2)	0.00		Cost-per-click for the campaign.
Fixed	float(10,2)	0.00		A fixed charge associated with the campaign.
PayFixed	enum('By Period', 'CIA', 'End of Campaign')	End of Campaign		This field indicates how the advertiser is to be invoiced for the fixed charges. If it is "By Period", the cost will be pro-rated by number of days in each invoicing period. If it is "CIA", it will all be charged in the first invoice for the campaign. If it is "End of Campaign", it will all be charged in the last invoice for the campaign.
AgencyCommission	float(10,2)	0.00		This percentage is applied to the total charges on each invoice and is subtracted from the amount due.
PurchaseOrder	varchar(64)			This field allows you to specify a PO number that will

OASIS 2.3 User's Manual

				appear on the invoice.
ForceInvoice	enum('Y','N')	N		Determines whether an invoice is to be generated for the campaign if CPM, CPC, and Fixed are all 0.
Notify	text			E-mail addresses to which weekly campaign reports should be sent (separated by commas)
PublisherID	int(11)			

[Documentation Home](#)

Creatives Table

This table stores all the information about creatives.

Note that there is an analogous table, SimCreatives, that is used during the inventory simulation.

Column	Type	Default	Extra	Comment
☒ CreativeID	int(11)		auto_increment	
☒ CampaignID	int(11)			
Name	varchar(64)			
☒ Status	enum('Active', 'Cancelled', 'Completed', 'Suspended')	Active		A creative is by default "Active". When it reaches its completion date or its impression target, it is marked as "Completed". You can manually mark it as "Suspended" or "Cancelled".
StartDate	date	0000-00-00		Creatives can have start dates that are independent of the campaign's start date -- note that an start date set earlier than the campaign will be overridden by the campaign's start date.
EndDate	date	0000-00-00		Creatives can have end dates that are independent of the campaign's end date -- note that an end date set later than the campaign will be overridden by the campaign's end date.
OverflowOK	enum('Y', 'N')	Y		Indicates whether the creative can be delivered during the overflow period at each hourly reload.
Redirect	enum('Y', 'N')	N		Set to "Y" if this is a third-party redirection.
MediaType	enum('Image', 'Rich Media')	Image		
MIMEType	varchar(64)			This is set to the MIME type of the content. For images, the graphic file format is used to determine the MIME type, but it can be overridden if it is misidentified.
Animated	enum('Y', 'N')	N		This is set automatically by the ImageMagick tool identify when images are uploaded. It is important for the delivery of cache-busting headers (the headers can't be used with animated GIFs).
☒ Width	smallint(5) unsigned			
☒ Height	smallint(5) unsigned			
Weight	int(11)	1		This is the weight of creatives relative to the other creatives in their campaign. If a daily impression target is specified for the campaign, the weight determines what share of those impressions will go to this creative. If the campaign has no impression target, then the creative

OASIS 2.3 User's Manual

				will run with a weight calculated from both the campaign weight and the creative weight.
Content	blob			This can contain the actual digital image for the creative, or in the case of a third-party redirection, it can contain the URL.
AltText	varchar(255)			
ClickthroughURL	varchar(255)			
Value1	varchar(128)			
Value2	varchar(128)			
Value3	varchar(128)			
Value4	varchar(128)			
Value5	varchar(128)			
ImpressionsGuaranteed	int(11)			The number of impressions to be delivered for this creative. This only has an effect if the campaign itself has a non-zero ImpressionsGuaranteed.
ImpressionsDelivered	int(11)			Each hour, the hourly maintenance script will update this value based on the number of impressions in the previous hour.
ClicksDelivered	int(11)			Each hour, the hourly maintenance script will update this value based on the number of clickthroughs in the previous hour.
DaysOfWeek	tinyint(3) unsigned	127		A bitmask indicating which days of the week the campaign should run; low bit is Sunday, high bit is Saturday. This value is ANDed with the bitmask from the campaign.
HoursOfDay	mediumint(8) unsigned	16777215		A bitmask indicating which hours of the day the campaign should run; low bit is hour 0, high bit is hour 23. This value is ANDed with the bitmask from the campaign.
ImpCap	int(11)	0		Per-visitor impression cap for the creative.
ImpCapInterval	int(11)	0		Interval over which the impression cap should not be exceeded (in seconds).
SupportingFile1	varchar(128)			References the file for Richmedia
SupportingFile2	varchar(128)			References the file for Richmedia
TemplateID	int(5)			
ClicksGuaranteed	int(11)			Number of clicks at which the delivery engine will stop scheduling the creative.

[Documentation Home](#)

Sections Table

Column	Type	Default	Extra	Comment
☒ SectionID	smallint(6)		auto_increment	
☒ PSectionID	smallint(6)			The SectionID of this section's parent.
Name	char(64)			
ReportName	char(64)			The name that will be used on reports. It is generally a bit longer and more descriptive than the Name, since the Name might only be informative in the context of its location in the section tree.
Active	enum('Y', 'N')			The delivery engine looks at this field when assigning creatives to sections. If it is 'N', no creatives will be assigned to this section or any of its children. Also, during inventory checking, if an active section is found to have no creatives assigned to it, mail will be sent to the administrator.
☒ PublisherID	int(11)			The ID of the publisher which owns this section.
Notes	text			

[Documentation Home](#)

CampaignAssignments Table

Column	Type	Default	Extra	Comment
CampaignID	int(11)			
SectionID	smallint(6)			
Type	enum('Include', 'Exclude', 'Exclusive')	Include		<p>If Type is set to "Include", the campaign will run on the section specified (and all sections beneath it in the section tree). If it is set to "Exclude", it will not run on the section specified (or any beneath it), overriding Include assignments made higher up on the tree. If it is set to "Exclusive", it will run on the section specified (and all beneath it), and no other campaigns will be allowed to run on those sections.</p> <p>Note that CreativeAssignments override CampaignAssignments.</p>

[Documentation Home](#)

CreativesTemplates Table

This table stores all the information about Richmedis.

Column	Type	Default	Extra	Comment
 CreativeTemplateID	int(11)		auto_increment	
CreativeTemplateName	text			Name of the Rich Media Template
CreativeTemplateContents	text			The script of the Rich Media Contents
Height	enum('Y', 'N')	Y		
Width	enum('Y', 'N')	Y		
ClickthroughURL	enum('Y', 'N')	Y		
Value1Title	varchar(128)			Title of the field.
Value1	enum('Y', 'N')			If 'Y', this field will be displayed; if 'N',not displayed.
Value2Title	varchar(128)			Title of the field.
Value2	enum('Y', 'N')			If 'Y', this field will be displayed; if 'N',not displayed.
Value3Title	varchar(128)			Title of the field.
Value3	enum('Y', 'N')			If 'Y', this field will be displayed; if 'N',not displayed.
Value4Title	varchar(128)			Title of the field.
Value4	enum('Y', 'N')			If 'Y', this field will be displayed; if 'N',not displayed.
Value5Title	varchar(128)			Title of the field.
Value5	enum('Y', 'N')			If 'Y', this field will be displayed; if 'N',not displayed.
SuppFile1Title	varchar(255)			Title of the Supporting File Browse field.
SupportingFile1	enum('Y', 'N')			If 'Y', this field will be displayed; if 'N',not displayed.
SuppFile2Title	varchar(128)			Title of the Supporting File Browse field.
SupportingFile2	enum('Y', 'N')			If 'Y', this field will be displayed; if 'N',not displayed.
UpdateButton	enum('Y', 'N')			If 'Y', this field will be displayed; if 'N',not displayed.
quote	varchar(10)	"		

[Documentation Home](#)

CreativeAssignments Table

Column	Type	Default	Extra	Comment
CreativeID	int(11)			
SectionID	smallint(6)			
Type	enum('Include', 'Exclude', 'Exclusive')	Include		<p>If Type is set to "Include", the creative will run on the section specified (and all sections beneath it in the section tree). If it is set to "Exclude", it will not run on the section specified (or any beneath it), overriding Include assignments made higher up on the tree. If it is set to "Exclusive", it will run on the section specified (and all beneath it), and no other creatives will be allowed to run on those sections.</p> <p>Note that CreativeAssignments override CampaignAssignments.</p>

[Documentation Home](#)

CompanionGroups Table

This table is used to represent each group of companion sections.

Column	Type	Default	Extra	Comment
☞ CompanionGroupID	int(11)			

[Documentation Home](#)

CompanionSections Table

This table is used to enumerate the sections in each companion group.

Column	Type	Default	Extra	Comment
CompanionGroupID	int(11)			
SectionID	int(11)			

[Documentation Home](#)

DailyTargets Table

This table is the master list of all creatives that are active during the current day.

Note that there is an analogous table, SimDailyTargets, that is used during the inventory simulation.

Column	Type	Default	Extra	Comment
☞ CreativeID	int(11)			
Target	int(11)			The number of impressions this creative should get during the current day.
Weight	int(11)	1		The weight assigned to the creative. This weight is relative to all other active creatives. It only matters for delivery purposes if Target=0.
Remaining	int(11)			The number of impressions remaining to be served. This starts out equal to Target, and is decremented from there.
HoursOfDay	mediumint(8) unsigned	16777215		A bitmask indicating which hours of the day the campaign should run; low bit is hour 0, high bit is hour 23. This value is computed by ANDing Creatives.HoursOfDay with the Campaigns.HoursOfDay.

[Documentation Home](#)

DeliveryControls Table

This table records the delivery controls assigned to campaigns and creatives.

Column	Type	Default	Extra	Comment
DeliveryControlID	int(11)		auto_increment	
CampaignID	int(11)			
CreativeID	int(11)			
Deliver	enum('deliver', 'deliver_only', 'no_deliver')			
Type	varchar(32)			defines the type of delivery control: IP, domain, language, user_agent, keyword, gt_country (geotarget/country), gt_region (geotarget/region), gt_metro_area (geotarget/metro area), gt_city (geotarget/city), expr (PHP expression)
Sense	int(11)			1 if delivery control is satisfied when the filter string matches the client's properties, 0 if satisfied when no match
Matches	enum('exact', 'regex', 'wildcard')			defines the type of string match to apply
FilterString	varchar(255)			the filter string itself

[Documentation Home](#)

RunningCreatives Table

This table contains all the CreativeID's of the creatives that have run during a given hour. It is used by the hourly maintenance script to record the deliveries for all creatives active during the past hour. Even if a creative is marked as "Suspended" or "Cancelled" during the hour, its ID will still be in this table, and thus, the hourly maintenance script can still retrieve its deliveries at the end of the hour.

After recording the hourly deliveries, the maintenance script clears RunningCreatives, and then while reloading shared memory, it reloads this table.

Column	Type	Default	Extra	Comment
☞ CreativeID	int(11)			
☞ Hour	tinyint(4)			

[Documentation Home](#)

Invoices Table

Column	Type	Default	Extra	Comment
InvoiceID	int(11)		auto_increment	
CampaignID	int(11)			
InvoiceNumber	char(64)			The invoice number which will show up on the printed invoice. It is built like this: AdvertiserID-CampaignID-YYYYMMDD, where YYYYMMDD is the date the payment is due.
FromDate	date			The beginning of the invoiced period.
ToDate	date			The end of the invoiced period.
InvoiceDate	date			The date the invoice is generated.
DueDate	date			The date the invoice is due to be paid. This is calculated by adding the NetDue preference value to the InvoiceDate.
ImpressionsDelivered	int(11)			Total number of impressions delivered during the invoice period.
ClicksDelivered	int(11)			Total number of clickthroughs delivered during the invoice period.
CPM	float(10,2)			The actual CPM used during the invoice period.
CPC	float(10,2)			The actual CPC used during the invoice period.
Fixed	float(10,2)			The fixed charge for the campaign.
AgencyCommission	float(10,2)			The agency commission rate used during this invoice period.
CPMCharge	float(10,2)			The actual CPM charges: CPM * ImpressionsDelivered.
CPCCharge	float(10,2)			The actual CPC charges: CPC * ClicksDelivered.
FixedCharge	float(10,2)			The fixed charge associated with this invoice period.
AgencyCommissionCharge	float(10,2)			The actual agency commission: (FixedCharge + CPMCharge + CPCCharge) * AgencyCommission
PurchaseOrder	char(64)			The PO number (inherited from the Campaign)
Sent	enum('Y', 'N')	N		Once an invoice has been marked as sent, this is set to "Y".
Paid	enum('Y', 'N')	N		Once an invoice has been marked as paid, this is set to "Y".

[Documentation Home](#)

PublisherInvoices Table

Column	Type	Default	Extra	Comment
InvoiceID	int(11)		auto_increment	
PublisherID	int(11)			The publisher being invoiced (not the publisher who created the invoice)
InvoiceNumber	char(64)			The invoice number which will show up on the printed invoice. It is built like this: AdvertiserID-CampaignID-YYYYMMDD, where YYYYMMDD is the date the payment is due.
FromDate	date			The beginning of the invoiced period.
ToDate	date			The end of the invoiced period.
InvoiceDate	date			The date the invoice is generated.
DueDate	date			The date the invoice is due to be paid. This is calculated by adding the NetDue preference value to the InvoiceDate.
ImpressionsDeliveredCredit	int(11)			Total number of impressions delivered for campaigns placed higher on the publisher hierarchy during the invoice period.
ClicksDeliveredCredit	int(11)			Total number of clickthroughs delivered for campaigns placed higher on the publisher hierarchy during the invoice period.
ImpressionsDelivered	int(11)			Total number of impressions delivered for the publisher's own campaigns during the invoice period.
ClicksDelivered	int(11)			Total number of clickthroughs delivered for the publisher's own campaigns during the invoice period.
CPMCredit	float(10,2)			The dollar amount credited for CPM (for campaigns running higher on the hierarchy).
CPCCredit	float(10,2)			The dollar amount credited for CPC (for campaigns running higher on the hierarchy).
FixedCredit	float(10,2)			The dollar amount credited for fixed campaign income (for campaigns running higher on the hierarchy).
CPMCharge	float(10,2)			The dollar amount charged for CPM (for the publisher's own campaigns).
CPCCharge	float(10,2)			The dollar amount charged for CPC (for the publisher's own campaigns).
FixedCharge	float(10,2)			The dollar amount charged for fixed campaign income (for the publisher's own campaigns).
HostingCharge	float(10,2)			Fixed hosting charge for the invoice period.
Sent	enum('Y', 'N')	N		Once an invoice has been marked as sent, this is set to "Y".
Paid	enum('Y', 'N')	N		Once an invoice has been marked as paid, this is set to "Y".

[Documentation Home](#)

HourlyStats Table

Column	Type	Default	Extra	Comment
Day	date	0000-00-00		
Hour	tinyint(4)			
SectionID	smallint(6)			
CreativeID	int(11)			
Impressions	int(11)			
Clicks	int(11)			
ImpressionErrors	int(11)			
ClickErrors	int(11)			

[Documentation Home](#)

CampaignDailyStats Table

Column	Type	Default	Extra	Comment
Day	date	0000-00-00		
CampaignID	int(11)			
PublisherID	int(11)			The ID of the publisher to which the stats are attributed.
Impressions	int(11)			
Clicks	int(11)			
ImpressionErrors	int(11)			
ClickErrors	int(11)			

[Documentation Home](#)

Delivery Table

This table stores the user IDs and IP addresses of visitors as they view ad banners on the various sections of your site. If a user clicks on an ad banner, this table is used to look up the last-seen creative so that the appropriate clickthrough can be delivered.

For more on why this is done using a heap table (and not using shared memory or other techniques), see [Delivery Table Design History](#).

Column	Type	Default	Extra	Comment
⚡ IDString	char(25)			
CreativeID	int(11)			
LastSeen	timestamp(14)			This is used by the minutely maintenance script to clean up old entries in the table. This column is not indexed. While indexing it would improve the speed with which the cleanup could be performed, it would slow down insertions and use more space, both of which would outweigh the benefits to the maintenance code.

[Documentation Home](#)

SectionDailyTraffic Table

This table is used to store just the impressions that are recorded for each section each day. The Impression number is actually the sum of the recorded Impressions and ImpressionErrors, since this table is really a measure of demand for each section.

Note that there is an analogous table, SimSectionDailyTraffic, that is used during the inventory simulation. Projected traffic numbers are written to that table.

Column	Type	Default	Extra	Comment
Day	date	0000-00-00		
SectionID	int(11)			
Width	smallint(5) unsigned			
Height	smallint(5) unsigned			
Impressions	int(11)			

[Documentation Home](#)

Users Table

Column	Type	Default	Extra	Comment
☒ Login	char(32)			
Password	char(32)			
FirstName	char(64)			
LastName	char(64)			
EmailAddress	char(255)			
Campaigns	enum('Y', 'N')	N		Can the user access the Campaign management interface?
Advertisers	enum('Y', 'N')	N		Can the user access the Advertiser management interface?
Sections	enum('Y', 'N')	N		Can the user access the Section management interface?
Reports	enum('Y', 'N')	Y		Can the user access the Report interface?
Admin	enum('Y', 'N')	N		Can the user access the Admin interface?
CampaignInsertion	enum('Y', 'N')	Y		Can the user access the Campaign Insertion interface?
Invoicing	enum('Y', 'N')	N		Can the user access the Invoicing interface?
LimitedCampaigns	enum('Y', 'N')	N		Can the user get limited access to campaigns?
PublisherID	int(11)			The ID of the publisher to which this user belongs.

[Documentation Home](#)

CampaignAccess Table

This table determines which campaigns are available to users who have "LimitedCampaigns" access.

Column	Type	Default	Extra	Comment
☞ CampaignID	int(11)			
☞ Login	char(32)			

[Documentation Home](#)

TrafficProfile Table

There should be 24 records in this table, one for each hour 0 through 23. It should be noted that the pageview values entered for each hour are used in a relative fashion to determine the percentage of traffic that runs at each hour. All you really have to do is enter the traffic numbers from a fairly representative day.

Column	Type	Default	Extra	Comment
Hour	int(11)			
Pageviews	int(11)			

[Documentation Home](#)

MaxSizes Table

For each dimension of creative you would like to be able to insert into the system, you must enter the maximum size (in bytes) of creatives of that dimension.

Whenever new creatives are uploaded, the software checks against these values to determine whether to accept the creative. If there is no value in this table, the creative will be rejected. If the uploaded creative exceeds the value in this table, the creative will be rejected.

Column	Type	Default	Extra	Comment
Width	int(11)			
Height	int(11)			
Size	int(11)			
Name	char(64)			

[Documentation Home](#)

OASISPrefs Table

This table is used to store a set of preference variable/value pairs. Those pairs are detailed in [Preferences](#).

Column	Type	Default	Extra	Comment
 PrefKey	varchar(32)			
PrefValue	text			
PerPublisher	enum('Y', 'N')			If 'Y', this preference is unique for each publisher; if 'N', it is global for the entire server.
PublisherID	int(11)			The ID of the publisher to which this preference belongs.

[Documentation Home](#)

Tips

Watch your user

DO NOT run the hourly script as anybody but the user who will be running the cron jobs! This will result in shared memory segments being created with the wrong ownership. It's possible then that the cron jobs will not be able to clean out the shared memory (A Bad Thing).

Be careful with DaysOfWeek and HoursOfDay

You can get in a lot of trouble if you try to specify different days of the week for a campaign and its creatives. For example, suppose you set up a campaign to run M-F. You put in a creative for Tu,Th. The campaign runs for one week, at 100,000 impressions. `daily_maint.php` will determine that the campaign can get 20,000 per day. The creative, however, only runs Tu,Th. On Monday, it will get no impressions. Tuesday, the engine will notice this short and will try to schedule 25,000 for the campaign, which will get assigned to the creative. Wednesday, `daily_maint.php` will schedule 25,000 impressions to the campaign, but the creative will get nothing, and thus the campaign will not run. Thursday, the campaign will get 37,500 impressions, all of which will go to the creative. On Friday, `daily_maint.php` will try to schedule 37,500 impressions to the campaign, but the creative will get nothing, and thus the campaign will not run. Now our campaign is over, and we've come up 37,500 impressions short.

Best practice is to have all your creatives set to the defaults and control your days/hours at the campaign level.

If you don't have impression targets, you don't need to worry about this. You could create a campaign with default days/hours, and then have a variety of creatives that run on different days.

Linking directly to creatives

You can link directly to a specific creative, and the impressions and clickthroughs can be tracked. Simply use the `c` CGI parameter to specify the creative number. Note that if you use IMG tagging, you'll have to also specify the `l` argument (that's a lowercase "L").

You may wish to create a special section that you can use for such ads, so that you'll know that the traffic to those creatives came from direct links.

For example:

```
<A HREF="http://your.server.name/oasisc.php?c=27&s=52"><IMG  
SRC="http://your.server.name/oasisi.php?c=27&s=52&l=1" BORDER=0 ALT="My  
Creative"></A>
```

This example will run creative 27, and any impressions or clicks will be logged to section 52. The `l=1` parameter makes sure that `oasisi.php` will log the impression (by default, it does not log impressions with the `c` parameter, because these URLs are usually generated by the embedded impression scripts, in which case the impression has already been logged).

When using `oasisi-e.php`, you can use the optional 5th argument to `get_creative()` to specify the creative id.

Note that if you link directly to creatives, you should not schedule the creatives for specific impression numbers, since the delivery engine will not have good control over how many times the creative gets delivered.

About clickthrough errors

If you're using IMG tagging, you'll notice in the reports and log files that there will be a certain number of clickthrough errors. This is to be expected. You will see clickthrough errors whenever a spider hits your site and follows the clickthrough link on an ad banner. Unfortunately, since clickthrough rates are generally low, these clickthrough errors often represent a large portion of the total clicks.

If you are not comfortable with the clickthrough error rates you're seeing, you might want to sift through the log files yourself to try to verify that these are legitimate errors:

- look for accesses to the clickthrough URL without a preceding access to the impression URL.
- look for obvious spider user agents

More about clickthrough errors

When a client presents a cookie to the clickthrough engine, it tries to look the cookie up in the delivery table. If it fails, it then falls back to the IP address. You might wonder how you could get a cookie from the client but not be able to find an entry in the delivery table. The first explanation is obvious; if a client views an ad banner and then waits so long to click through that the entry has already been cleared from the delivery table (ClickthroughWindow preference setting).

The second explanation is not so obvious. When ad tags for two different sections (let's say sections X and Y) appear on one page, the client may request the creatives for both in parallel. Both requests come in without cookies, so the impression engine hands out two different cookies. Only the second cookie received (let's say the one given out with the ad for section Y) is kept by the client. When the clickthrough comes in for section X, the cookie is presented. However, it was not the one recorded in the delivery table for section X. Thankfully, unless the user is behind a very busy proxy server, the fallback to the IP address lookup will generally result in the right clickthrough. It's also worth noting that this situation can only happen the very first time somebody comes to your ad server (or whenever he clears his cookie table) and he clicks on one of the first ads he sees. Presumably this will be a fairly rare situation.

Last hour of the day

If your site's traffic profile dips at night to low levels, you might want to take advantage of that in entering your traffic profile numbers. You can enter 0 for hour 23, and use that hour entirely for making up any shortfalls during the other hours of the day. If all goes well, none of the campaigns with impression targets will be scheduled during that hour, but in the off chance that there is a shortfall in hour 22, you'll have hour 23 to make up the difference.

This strategy also reduces the chance that you might have an hour 23 shortfall on the last day of a campaign and come up short for the entire campaign.

Campaign/Creative Weights

Campaign weights are relative to other campaigns. Creative weights are relative to the other creatives within the same campaign. The final weight that a creative gets when it is scheduled is

$$\text{Final Weight} = \text{Campaign Weight} * (\text{Creative Weight} / \text{Sum of all Creative Weights in Campaign}) * 100$$

This formula preserves the overall weight of the campaign when the creatives are scheduled. Here's an example:

Campaign, Creative Weights

Campaign A: 10

Campaign A, Creative 1: 20

Campaign A, Creative 2: 5

Campaign B: 20

Campaign B, Creative 3: 10

Final Weights

Creative 1: $10 * (20 / 25) * 100 = 800$

Creative 2: $10 * (5 / 25) * 100 = 200$

Creative 3: $20 * (10 / 10) * 100 = 2000$

Now the ratio of Creative 1 to Creative 2 is still 4:1. The ratio of Campaign A to Campaign B is still 1:2 (800 + 200 for Campaign A and 2000 for Campaign B).

This weighting works very well if all the creatives are assigned to the same section(s). But one scenario in which this weighting can cause some difficulty is if you have one campaign with a single creative running on many sections and another campaign with many creatives each running on one section. If you want to balance the two campaigns so that each gets roughly the same number of impressions, you may find that you have to make some adjustments. Here's an example: **Campaign, Creative Weights**

Campaign A: 10

Campaign A, Creative 1: 10

Campaign A, Creative 2: 10

Campaign A, Creative 3: 10

Campaign A, Creative 4: 10

Campaign A, Creative 5: 10

Campaign B: 10

Campaign B, Creative 6: 10

Final Weights

Creative 1: $10 * (10 / 50) * 100 = 200$

Creative 2: $10 * (10 / 50) * 100 = 200$

Creative 3: $10 * (10 / 50) * 100 = 200$

Creative 4: $10 * (10 / 50) * 100 = 200$

Creative 5: $10 * (10 / 50) * 100 = 200$

Creative 6: $10 * (10 / 10) * 100 = 1000$

Now suppose that creatives 1-5 each run on a single section 1-5, respectively. Creative 6 runs on all 5 sections. Since the two campaigns have equal weights, you might expect their creatives to run with the same frequency. But they won't. Creative 6 will run 5 times as often as the other combined. When the delivery engine makes its decision as to which creative to run, it compares the weights of all creatives assigned to run on the given section. In each case, it will be comparing a creative with weight 200 to one with a weight of 1000. Creative 6 will win 5/6 (1000/1200) times.

To balance the campaigns more evenly, you might want to compensate so that Campaign A gets a weight of 50 (multiply the original campaign weight by the number of creatives).

Note that this method gets considerably more complex if you schedule the creatives for overlapping sections (or if the sections covered by Campaign A are not the same as those for Campaign B). Reliable balancing may be impossible in such cases.

Importing Campaigns in Progress

If you are setting up campaigns that are currently running on another ad server, you may need to do some tweaking in the database to get the campaigns to look right on an invoice. This advice really only applies if the campaign has a non-zero Impressions Guaranteed value.

Suppose you have a six-month campaign that gets 100,000 impressions per month running on some other ad server. Halfway through the campaign, you want to move the campaign from the old ad server to OASIS. You should set up the campaign *exactly* like it appears on the old server (with the original start/end dates and impression target). Don't make the campaign active in OASIS yet. You then need to figure out exactly how many impressions have been delivered thus far on the old server. Get the CampaignID for the OASIS campaign (you can see it in the URL on the campaign edit page). Then go into MySQL and run this query on the oasis database:

```
update Campaigns set ImpressionsDelivered=300000 where CampaignID=36
```

(Of course, you'll have a different CampaignID and impression number). You'll also need to do this for a creative in the campaign. If you only have one creative in the campaign, you can do it quite easily:

```
update Creatives set ImpressionsDelivered=300000 where CampaignID=36
```

Otherwise, you'll need to have the CreativeIDs of the creatives. You can pick one of the creatives and assign all the impressions to that creative. (If you have specific Impressions Guaranteed values for creatives in the campaign, you'll need to be judicious about which creative you give the impressions to, or maybe you can get actual breakdowns from your old server).

If by some chance, you have to do this on an active campaign, you can still do it, but keep in mind that ImpressionsDelivered may not be zero anymore (and the hourly maintenance script might need to write to this record, so you have to be careful when you do this; it's best to avoid the minutes right before and after the top of the hour). The query for an active campaign would be this:

```
update Campaigns set ImpressionsDelivered=ImpressionsDelivered+300000 where CampaignID=36
```

Be careful with Exclusives

Your best bet if you are using exclusive assignments is to use it with campaigns that do not have impression targets. There are two types of problems that can occur if you use exclusive assignments for campaigns with impression targets.

First, your campaign may need more impressions than can be delivered by the section to which you assigned it exclusively. You can compensate for this by assigning the campaign to other sections in addition to the exclusive assignment.

Second, the section may get more impressions than the campaign needs. When the campaign hits its hourly target, blank ads will be served. This problem may be compounded if you've assigned the campaign to other sections, as some of its hourly target impressions will be delivered on the other sections, leaving even fewer for the exclusive assignment. A workaround for this situation is to assign another campaign (perhaps a house campaign) with no impression target to the section as exclusive. Both campaigns will run on the section (of course, the campaign with an impression target will get its impressions first). This won't be a true "exclusive" for either campaign, but it will avoid having blank banners run.

Stopping MySQL

In general, you don't want to stop MySQL while you're live. But sometimes a system administrator may find it necessary to do so.

Do **NOT** stop it while one of the maintenance scripts is running. This will cause you to lose an hour's worth of stats and wreak havoc on the scheduler, among other things.

If you stop MySQL and you're using IMG tagging to deliver your banners, you will experience some problems. Banners will continue to be delivered, but they won't go into the Delivery table, so if users try to click through, they won't get redirected properly.

Note that since the Delivery table is stored in RAM, when you restart MySQL, it will have a completely cleaned out Delivery table.

If you're not using IMG tagging to deliver banners, you won't have any problems, since it does not depend on the Delivery table.

[Documentation Home](#)

Troubleshooting

Shared Memory Errors

A number of people have reported that they aren't seeing banners, even though they have set up sections and creatives (and assigned campaigns/creatives to the sections!). When they look at the "Hourly Assignments" admin page, they get an error like this:

```
Warning: shm_attach() failed for key 0x4f415300: memorysize too small in /home/ads/www/oasis/mo
```

or you might see this error when running `hourly_maint.php`:

```
Error clearing shared memory segment OAS0
```

This indicates that the shared memory segment holding the hourly assignments was not created properly by the `hourly_maint.php` script.

You can verify this by typing "ipcs" at the command prompt. You will see a list of shared memory segments, their sizes, owners, permissions, and their keys. If none has a key of "0x4f415300" then you know something has gone wrong.

Either you never ran `hourly_maint.php` (not likely, since `install.php` tries to do it for you), or something went wrong when you did.

It is important to note that the first time `hourly_maint.php` is run, it needs to get the `start` option:

```
hourly_maint.php start
```

Another thing that you *must* ensure is that the maintenance scripts are always run by the Web user (the user as which your Web server runs). If any of the shared memory segments are created by another user (especially root), you'll never get off the ground.

Use `ipcs` to check the ownership of your shared memory segments. If any are owned by the wrong user, use `ipcrm` to remove all of OASIS' shared memory segments (and the semaphore as well). You can identify them easily, since all have keys that start with "0x4f41". Once you've cleared them out, run `hourly_maint.php start`. Now run `ipcs` again. You should have nine shared memory segments, all owned by the right user.

register_argv_argc set incorrectly in php.ini

If your `php.ini` has

```
register_argv_argc = Off;
```

the maintenance scripts will be unable to get their arguments, which will lead to incorrect behavior.

Errors running hourly_maint.php

The most common problem with `hourly_maint.php` is that it is run for the first time without the `start` argument. Be especially careful when running it with `su`

```
su nobody -c /path/to/hourly_maint.php start
```

will not work. The system will pass "start" as an argument to su, not to hourly_maint.php. Instead, do this:

```
su nobody -c '/path/to/hourly_maint.php start'
```

Of course, use the appropriate username and path for your system!

Errors Reloading in OASIS Enterprise

If you have problems reloading your slave servers, check the following:

- MySQL running on all slaves
- Adequate disk space on master and slaves
- `post_max_size` large enough to accommodate the XML produced and sent to slaves (/tmp/OASIS-SL2*.xml)
- `memory_limit` large enough to accommodate the XML produced; figure at least a factor of 4 over the filesizes to parse the XML and hold an XML DOM in memory
- Clock sync between master and slave; for security purposes, the digital signatures generated by the master are valid for only a short period of time; if the clocks get out of sync, the slave will reject reload requests from the master.
- Look at the files /tmp/OASIS* on the master. Here's a brief rundown of what should be in each:
 - ◆ OASIS-SL1-hostname - creative traffic statistics from hostname
 - ◆ OASIS-SL1-hostname.done - a "flag" file to indicate that master_reload1.php is done
 - ◆ OASIS-SL2-hostname - overflow creative traffic statistics from hostname
 - ◆ OASIS-SL2-hostname.done - a "flag" file to indicate that master_reload2.php is done
 - ◆ OASIS-SL2-hostname - overflow creative traffic statistics from
 - ◆ OASIS-SL3-hostname.done - a "flag" file to indicate that master_reload3.php is done
 - ◆ OASIS-assignments - simple text file with all creative-to-section assignments; used to build XML reload message
 - ◆ OASIS-creative_content - XML for the creative content; inserted into XML reload message
 - ◆ OASIS-delivery_controls - simple text file with delivery controls; used to build XML reload message
 - ◆ OASIS-imp_caps - simple text file with impression caps; used to build XML reload message

If you don't see all of the appropriate host-specific files, something may have gone wrong in the client-server communications. Check the master's OASIS admin log, and check the slave's Apache logs for errors.

- Look at the files /tmp/OASIS* on the slave. Here's a brief rundown of what should be in each:
 - ◆ OASIS-MS1 - list of all running creatives (along with a timestamp and a digital signature)
 - ◆ OASIS-MS2 - XML for the complete reload of the server
 - ◆ OASIS-MS3 - timestamp and digital signature

If you don't see all of these files, something has gone wrong in the client-server communications. Check the slave's Apache logs for errors.

Banners working for IMG tags, but not for embedded or IFRAME/ILAYERS

At install time, if you forget a trailing slash (or otherwise mis-define the URL) when you define `oasis_url` in your `oasis.cfg`, `oasisi-i.php` and `oasisi-e.php` will generate bad URLs for `oasisi.php` and `oasisc.php`.

OASIS 2.3 User's Manual

To fix it, manually edit `oasisi-i.php` and `oasisi-e.php` to correct the bad URLs.

Cookie Problems

If OASIS runs on a machine that has no hostname, the cookie that is sent to browsers will incorrectly specify its domain and will not be returned by browsers. Thus, users may not click through to the right URL if you're using IMG tagging (they might click through correctly based on their IP addresses, but this is not reliable for hosts that share an IP address, like those behind a NAT device).

This must be fixed in three places: `oasisi-e.php`, `oasisi-i.php`, and `oasisi.php`. Change

```
setcookie("OASISID", $id, time() + 157680000, "/", ".yourdomain.com");
```

to

```
setcookie("OASISID", $id, time() + 157680000, "/", "");
```

MySQL Problems

Some users have reported problems with persistent MySQL connections exhausting the number of available connections. If you have this option enabled and are running out of available connections, you may want to try disabling it.

[Documentation Home](#)

Appendix: Delivery Table Design History

I've tried a number of different ways to store the delivery table: shared memory, a dbm file on a ramdisk, and a MySQL heap table. The shared memory did not work, because PHP's shared memory manager is a bit too simplistic. It uses a flat linked list to manage the variables you've put into shared memory. Once you get over a couple thousand variables, it grinds to a halt. In addition, when you store a variable at a key that is already in use, the shared memory manager relocates everything that is stored after that key to make room for your new variable. Imagine if you've got 20 MB of data in shared memory, and you want to change the variable located at key #1!

The dbm file idea didn't get very far. It was too slow, and I didn't want to waste any more time trying to figure out why.

The heap table in MySQL rocks! I was able to insert 200,000 records into it (that roughly represents 100,000 unique visitors) in 81 seconds (0.0008 seconds per insertion, which in my mind is fantastic). The resulting table occupied 10.3 MB (that's the 1024**2 kind of MB, not 1000**2). This suggests that if you want to use no more than 32 MB for your delivery table, you could support approximately 300,000 unique visitors. If you clean out the delivery table every 15 minutes, you could conceivably handle 1.2 million unique visitors per hour.

Of course, as the number of sections that each visitor views grows, the number of unique visitors you can handle goes down. If each visitor sees on average 3 different sections, you'll only be able to handle 100,000 in 15 minutes.

To complicate matters further, users behind proxy servers will require roughly half the storage, because they share IP addresses. Users not accepting cookies will require more storage, because each time they view a page, we'll record a new ID.

[Documentation Home](#)