

Managing Configuration Files

On This Page:


-
- [Single Installation, Multiple Delivery/Image Domains](#)
 - [Step 1: Set up the required domain & web server hosting](#)
 - [Step 2: Copy the default config file to a domain config file](#)
 - [Step 3: Extend the new domain config file to override the "default" domain settings](#)
- [Single Code Base, Multiple Installations](#)

Managing Revive Adserver's configuration files.

Most configuration of Revive Adserver should be done via the administrative UI, but adding support for additional domain name access needs to be done outside of Revive Adserver.

Single Installation, Multiple Delivery/Image Domains

As an administrator user under **Configuration > Global Settings > Banner Delivery Settings**, it is possible to edit the Revive Adserver "Server Access Paths":

 **Revive Adserver Server Access Paths**

Admin Interface URL	http(s)://	<input type="text" value="example.com/www/admin"/>
Delivery Engine URL	http://	<input type="text" value="example.com/www/delivery"/>
Delivery Engine URL (SSL)	https://	<input type="text" value="example.com/www/delivery"/>
Image Store URL	http://	<input type="text" value="example.com/www/images"/>
Image Store URL (SSL)	https://	<input type="text" value="example.com/www/images"/>

These are the main URLs that need to be used to access the admin interface, delivery scripts and locally stored images. While these can be edited via the Revive Adserver administrative UI, only one domain can be set for each option.

However, it is possible to configure additional delivery/image domains manually.

For example, to extend the above configuration so that the *same* installation of Revive Adserver also works on "foo.com":

Step 1: Set up the required domain & web server hosting

Before changing anything in Revive Adserver, configure the necessary domain name and DNS settings, and set up your web server so that it responds to the new domain (e.g. "foo.com") and that it has the same hosting configuration as your existing Revive Adserver host (e.g. "example.com").

If this is working correctly, you should be able to go to, for example, `http://foo.com/www/admin`, and your webserver will end up serving the same content in the same way as if you had loaded `http://example.com/www/admin` - and, because your Revive Adserver installation's `var` directory has a file called `default.conf.php` in it, access to the Revive Adserver installation via any domain that it doesn't recognise will be treated as if it came from the domain listed in the `realConfig` entry of that `default.conf.php` file - which in this case, will be "example.com" - so, your Revive Adserver installation should still work (using the `example.com` domain).

Step 2: Copy the default config file to a domain config file

Make a copy of the `default.conf.php` file in your Revive Adserver's `var` directory, named according to the domain you want to use, in this case: `foo.com.conf.php`.

Now, when you access your Revive Adserver installation via the `foo.com` domain (instead of `example.com`), you'll still get the same outcome as in Step 1 above - but this time, instead of access doing via the default configuration file (which informs Revive Adserver to use the configuration located in the `example.com.conf.php` file), access will now happen via the `foo.com.conf.php` file (which does the same thing - it informs Revive Adserver that the real configuration to use can be found in the `example.com.conf.php` file).

Step 3: Extend the new domain config file to override the "default" domain settings

The magic that allows this type of setup to be enabled is that Revive Adserver configuration files can contain a `realConfig` entry, to inform Revive Adserver that the "real" configuration for how it should operate is located in another file, but it is also possible to add regular configuration directives to the same file, which then "override" the configuration directives of the `realConfig` file.

So, to enable the new `foo.com.conf.php` file to instruct Revive Adserver that when the service is accessed via the `foo.com` domain, it should operate using a set of server access paths that are specific to the domain, but otherwise, all other configuration items should be used as configured via the `example.com` domain configuration file, the `foo.com.conf.php` file should be extended to be:

```
;<?php exit; ?>
;*** DO NOT REMOVE THE LINE ABOVE ***
realConfig="revive-dev.fornax.net"
[webpath]
admin="foo.com/www/admin"
delivery="foo.com/www/delivery"
deliverySSL="foo.com/www/delivery"
images="foo.com/www/images"
imagesSSL="foo.com/www/images"
```



Management of multiple domains is, as mentioned above, not supported by the Revive Adserver administrative UI. If you manually configure the Revive Adserver configuration in this way, it is highly recommended that you set permissions on all additional configuration files so that the Revive Adserver administrative UI *cannot* edit the files. This will ensure that you don't accidentally update the configuration when accessing the UI via a "secondary" domain; updating the configuration in this way would cause Revive Adserver to create a complete, "primary" domain style configuration, without a `realConfig` reference, allowing your two configuration files to potentially diverge in terms of Revive Adserver operation, which will likely lead to incorrect, and incredibly difficult to diagnose, behaviour.

Single Code Base, Multiple Installations

Using the same principles as above, it is possible to use one Revive Adserver code base to run two or more completely independent installations of Revive Adserver.

However, this is generally not recommended, because there are certain operations (e.g. initiating an upgrade) that depend on certain files existing in the Revive Adserver `var` directory. Thus, it becomes impossible to cleanly manage the upgrade process for each installation separately - all installations have to be taken offline to upgrade each one, and when one upgrade is completed, the upgraded server will be affected when upgrading the next one, etc.

What appears to be a time saving approach in the short term rapidly becomes a complex management issue.

As a result, figuring out how to run multiple installations on one code base is left as an exercise for the reader - if you can figure it out, hopefully, you can see why it's a bad idea. If you decide to use it anyway, and have the skills to code the necessary tools to automate the tasks required so that it's not a bad idea, please consider contributing to the Revive Adserver project!