Setting Up the Test Framework

On This Page:

Setting Up the Test Framework.

A guide to setting up the Revive Adserver development test framework .

Pre-Requisites

In order to set up the Revive Adserver test framework, you will need:

- An ***independent*** Revive Adserver test framework environment (that is, a Revive Adserver Development Environment, but without the final stage of having installed Revive Adserver via a web browser); and
- The PHP core directive `register_argc_argv` must be enabled.

⚠ **Please note that if you attempt to run the tests in an existing Revive Adserver development environment (that is, using a Revive Adserver directory which has been through an installation process), then many of the tests will fail. This is because the process of installing Revive Adserver will copy various files into the Revive Adserver `/var` and `/plugins` directories, which will prevent various tests from running correctly, as they will need to make changes in these directories, but will not be able to do so, due to the existing files from the installation process.**

Test Framework Environment Configuration

Copy & Configure the Test Configuration File

In your Revive Adserver test framework environment:

- Copy the `/etc/test.conf`.php file to the `/var` directory;
- Edit the `/var/test.conf`.php file:
    - In the `[database]` section:
        - Set the appropriate `database` type;
        - Set the appropriate `host` of your database;
        - If required, set the `socket` or `port`;
        - Set an appropriate `username` and `password` to connect to the database with (not that this username will need permissions to create, modify and drop the test database); and
        - Set the `name` of the test database, if a different name is needed.
    - Update the ownership or permissions of the `/var/test.conf`.php file to allow your web server user to write to this file.

⚠ **If tests are failing with HTTP 500 errors, the database connection details in the `test.conf`.php file may be the cause.**

In particular, when using a MySQL database, the `mysql4_compatibility` setting may be relevant.

If you are still not having any luck setting the value(s) for the `[database]` section manually, try copying a known good `[database]` section from a Revive Adserver development environment - but remember to update the details for the database name & any other required values

Accessing the Test Framework via a Web Browser

To access the test framework via a web browser, point your browser at the /tests directory of your Revive Adserver test framework environment.

So, for example, if your Revive Adserver test framework environment is accessible at `http://www.example.com/revive-adserver-tests`, then the test framework will be available at:

- `http://www.example.com/revive-adserver-tests/tests`

Test Execution

Web Browser

To execute tests, or a group of tests, simple click on the test (or group) in the web UI.

> Many of the tests, especially those that access the database, are known to leave data/the database in an invalid state that affect the running of the next test, which results in false negative test results.
>
> This is a known issue with the Test Framework's web UI. Nevertheless, the web UI is a particularly effective way of executing tests, particularly a specific test, for the purposes of checking a specific area of functionality, or to execute a test via `xdebug`.
>
> However, for bulk test execution (e.g. prior to committing code or releasing Revive Adserver), the command line is a far better way to execute *all* tests.

> When test crash or die, with no useful output in the browser, the best places to look for a reason (before resorting to stepping through the code with a debugger) are:
>
> - The Revive Adserver `/var/test.log` file; and
> - The web server log file.

Command Line

To run all of the unit tests, the following commands can be used:

```
cd /path/to/revive/tests
/path/to/php -q cli_test_runner.php /path/to/php unit
```

Alternatively, to run all of the integration tests, replace "unit" with "integration".

> **When running tests via a web browser and the command line, permissions can be an issue.**
>
> Running tests via the web browser will create test files & cache data in various directories that are owned by the web server user. However, running tests via the command line will create test files & cache data as the user which executes the test command.
>
> To ensure that tests work in *both* environments, you may need to either:
>
> - Ensure that you run the command line tests as the same user that the web server runs under; or
> - Bulk assign read/write permissions to the following directories & files:
>   - Everything in Revive Adserver's `/var` directory.

Reviewing Command Line Results

Instead of outputting the command line results in XML, you may find it easier to:

- Redirect the output of the command line tests to a file;
- Use the `simpletest2junit.xslt` in the Revive Adserver `/tests` directory to transform the XML into JUnit format; and
- Use your favourite tool that supports JUnit test results format to review the results.