

Options UI

Upon invoking Start > All Applications > Developer Tools > A.M.P. Web App Development > Options, or Options from the Tray Icon menu, or running the executable binary 'sampooptions', a window is presented.

Any Web App Developer logging into the OS has rights to run this UI and thus has rights to change these shared Server options. (!)

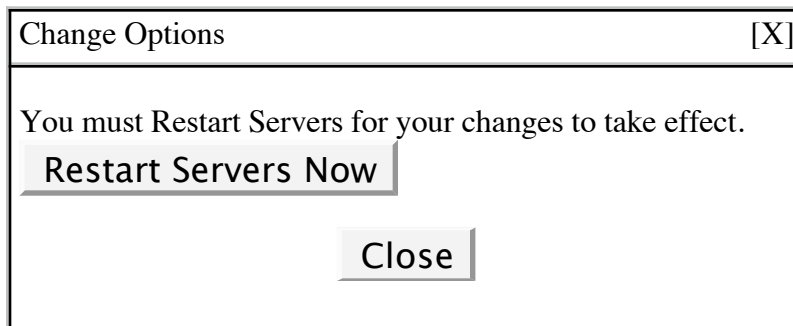
Any user can change Options; does not need to be logged in as root.

GENERAL INTERACTIONS

Tabs: The Ok button saves all unsaved edits on all tabs; switching between tabs does not lose edits. The tabbed view is implemented outside the portion of the window containing the Ok and Cancel buttons. The tabbed view area is surrounded by a thin line border to make this visually apparent. Both Ok and Cancel will close the window.

Ok:

After making changes and clicking Ok, if one of more of the servers are Started, this message is presented:



If both servers are Stopped, no message is presented.

INTERACTION WITH DIRECT CONFIG FILE EDITING

Everytime the Options window opens, the Apache, MySQL, PHP, and OS config files are parsed for the values shown on this window. **This UI does not maintain its own copies of the current value of those parameters.**

Thus manual edits of the config files will be picked up by this UI; it is merely a simplified alternate UI for editing the standard config files.

When any parameter is changed in a tab of the Options UI, the current value of **all** settings shown on that tab (for that service) are written to the underlying configuration files (not just the individual fields that are touched). Configuration file settings which have no UI here are

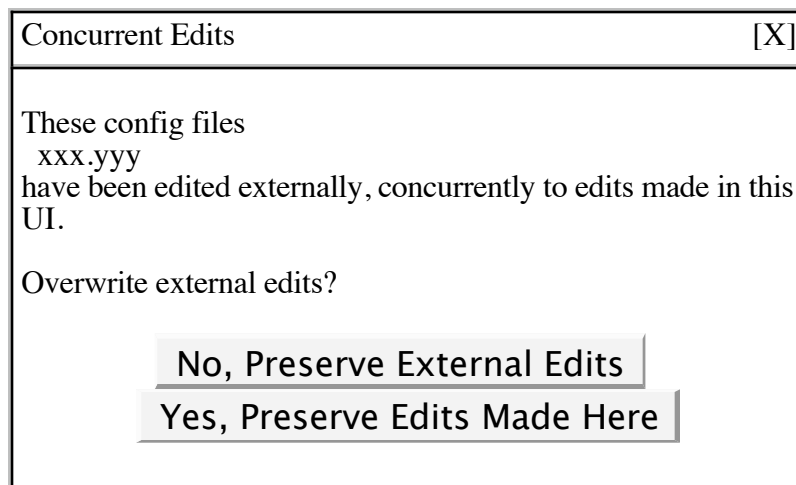
not affected in the file. Thus, the current values of all fields shown in the Options UI will always be correct when viewed after a change is applied.

If a user has "gone around" the Options UI and made configuration changes manually in the apache2, mysql, php, etc configuration files while this UI is open, then some current values shown may be incorrect. If then he (or another user) changes it back using the Options UI, the corresponding manually changed parameters in the config file will be overwritten (the remainder of the file and thus any settings not offered in the UI will be unaffected), though a timestamp comparison will detect this case and confirm it (see below).

Note that manual changes without a corresponding parameter in the Options UI will be unaffected by later manipulation of the Options UI. Thus the Options UI is write-only but only to portions of the config files; it has to read the files to find the parameter directives (to overwrite them) but rewrites the whole file so as to not affect other manual changes.

In other words, the UI only overwrites the parameters that it provides a UI for, leaving the rest alone. Technically this means that only certain directives ***inside*** the config files are overwritten, and the file is otherwise rewritten as it was before.

When the Options UI window opens, it records the timestamp of each of the various config files. When Ok is clicked, the config files' timestamps are again checked. If a config file needs updating due to user edits on a particular tab in this window, and the corresponding config file appears to have been edited after this window opened, the following message appears:



To support repair of messed-up Options or messed-up manual configuration file edits, a Repair action is available (see below).

The Options UI is not available if the A.M.P. Web App Development system has never been Initialized. Running the Options executable binary fails immediately with an error message if Initialize has never been done.

General tab

A.M.P. Web App Development Options					[X]
<input type="checkbox"/> General	<input type="checkbox"/> Apache 2	<input type="checkbox"/> PHP	<input type="checkbox"/> MySQL	<input type="checkbox"/> FTP	
<input type="checkbox"/> Start Servers when Solaris boots					
<input type="checkbox"/> Start Servers when I log in, Stop Servers when I log out					
<input type="button" value="View Advanced Details"/>					
<input type="button" value="Ok"/> <input type="button" value="Cancel"/>					

Checking a checkbox unchecks the other, to prevent both checkboxes from being checked simultaneously.

View Advanced Details:

Clicking View Advanced Details creates a temporary text file in /tmp containing a detailed human-readable description (not computer readable, NOT xml) of the installed AMP stack -- all the details about the installed stack that we do not expose in the UI, such as install paths, port numbers used, files modified or replaced, system commands issued, users created, groups created, etc. The contents of this file will vary depending on whether any user has Initialized their Web App Development. It is essentially a record of hidden automated actions have been invoked by the A.M.P. stack system. Entries in this file are suitably commented to be meaningful to a sophisticated reader.

The purpose of this feature is to allow the rest of the Options UI to remain as simple as possible. Sophisticated users can come here and discover some obscure parameter isn't set to their liking (or isn't changed from the factory default) and then change that parameter using other means (i.e. editing the config files directly). There are potential pitfalls doing this, see "INTERACTIONS..." above.

The temporary file is opened using the OS's default text editor.

This file is output only. It is never read. Editing, deleting, renaming, or overwriting this file has no effect, since it is regenerated each time this button is clicked.

This file is also useful for support purposes; customers requesting technical support should be asked to send the contents of that file along with their support request. Therefore, useful support information (i.e. cpu specs, available and total memory, disk space, OS version, versions of all other software installed, what services are running and their versions, etc) should be included in the file.

As soon as edits are made anywhere in the window (in this tab or other tab(s)), the View Advanced Details button is disabled.

Apache 2 tab

A.M.P. Web App Development Options
[X]

General
Apache 2
PHP
MySQL
FTP

Port Number:

Document Directory:

Website homepage URL:

Enable personal Home Directories for users
(i.e. http://xxx.8080/~username/)

(edit httpd.conf)

Note: The default Document Directory is configured at the first user's Initialization with an Access Control List with these permissions:

- any member of the unix group webDeveloper can read and write existing and future-created files, including those in existing and future-created subdirectories
- user apache2RunAs can read existing and future-created files, including those in existing and future-created subdirectories
- user apache2RunAs can execute existing and future-created .php files, including those in existing and future-created subdirectories

If the Document Directory is changed, the new directory is given an ACL as described above.

Syntax checks: When editing a port number, a check is made at the time the field loses focus. If that port is unavailable (in use by another service), Ok is disabled and the field is flagged with a red !. When editing the Document Directory, if the text entered is an invalid or non-existent path, Ok is disabled and the field is flagged with a red !.

As user edits the Port number field (on every user keystroke and cut and paste event), the Website homepage field is updated.

The URL is implemented as selectable, copyable text. "xxx" is the host's name; or 'localhost' if not networked. 8080 is replaced by the value entered in the field above. The Open button opens that URL using the OS's default web browser. If there are pending unsaved edits in the window, this button is disabled.

"Advanced Configuration" opens the http.conf file using the OS's default text editor. If there are unsaved edits on any tab anywhere in this window, the user is prompted whether to Save those edits (yes/no buttons). Regardless of which answer is given, this window is closed before the text editor is opened.

Repair:

Restores the configuration file to the state it was in immediately after the first Initialization. In other words, overwrites and repairs (corrects) any erroneous manual edits to the configuration file.

This feature is useful to restore messed up configurations.

Clicking this button yields:

Repair[X]

Are You Sure?

All manual Advanced configuration edits that may have been made will be lost, and the Apache 2 httpd.conf configuration file will be replaced with known good contents.

Webpages created and database data are unaffected.

All users on this system will be affected.

As soon as edits are made anywhere in the window (in this tab or other tab(s)), the Advanced Configuration and Open buttons are disabled.

PHP tab

A.M.P. Web App Development Options[X]

General Apache 2 PHP MySQL FTP

PHP version:

Error Reporting Level :

(edit php.ini)



Note: PHP is preconfigured to always check file dates (minimal caching of .php files), and with the debugger control module installed and ready for control by NetBeans.

"Advanced Configuration" opens the php.info file using the OS's default text editor. If there are unsaved edits on any tab anywhere in this window, the user is prompted whether to Save those edits (yes/no buttons). Regardless of which answer is given, this window is closed before the text editor is opened.

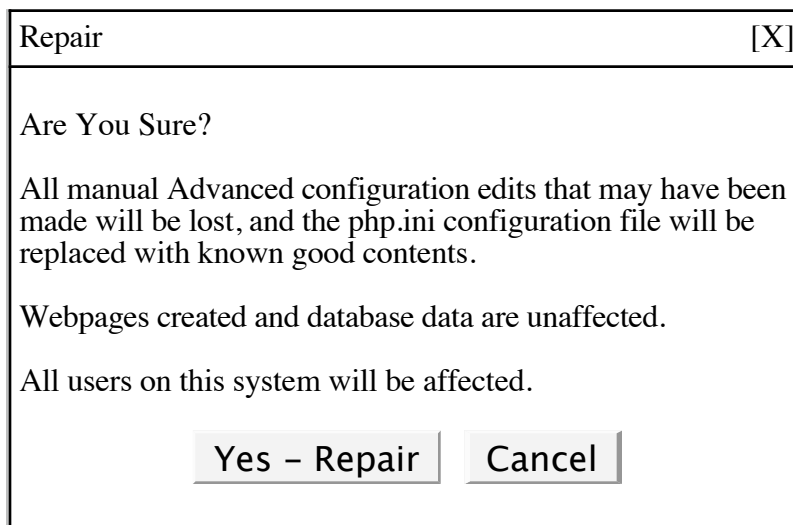
Show PHP Info writes a temporary-filenamed .php file into the docroot, and then opens it using the OS's default web browser. The contents of the file is a call to php_info() which dumps out pages of config data.

Repair:

Restores the configuration file to the state it was in immediately after the first Initialization. In other words, overwrites and repairs (corrects) any erroneous manual edits to the configuration file.

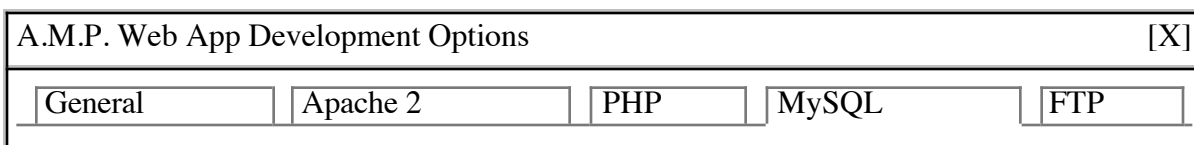
This feature is useful to restore messed up configurations.

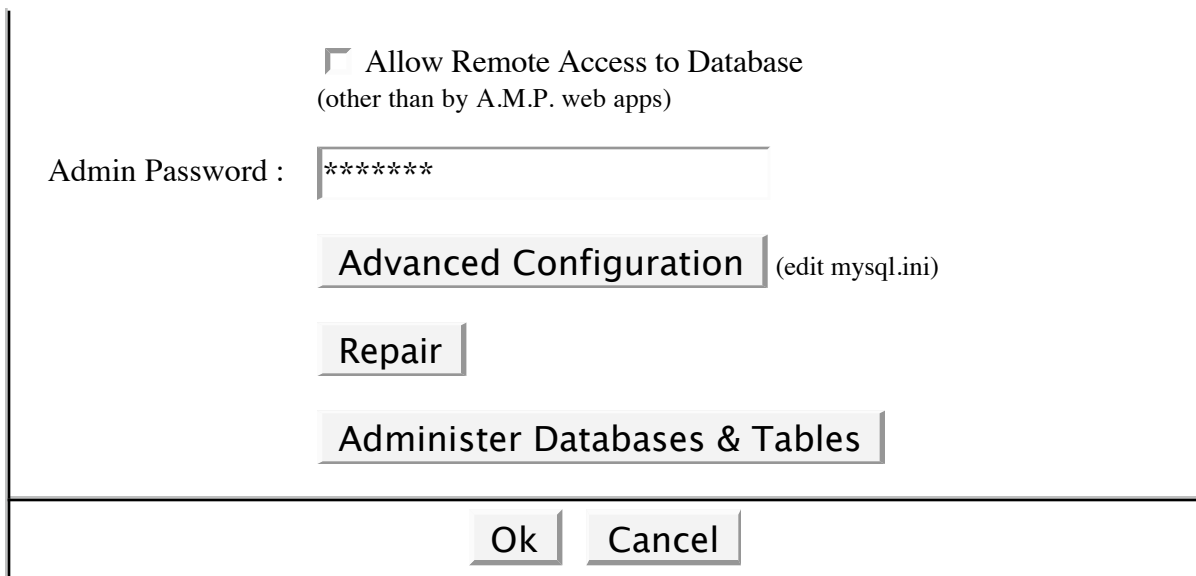
Clicking this button yields:



As soon as edits are made anywhere in the window (in this tab or other tab(s)), the Advanced Configuration and Show PHP Info buttons are disabled.

MySQL tab





Allow Remote Access to Database
(other than by A.M.P. web apps)

Admin Password :

Advanced Configuration (edit mysql.ini)

Repair

Administer Databases & Tables

Ok **Cancel**

The Allow Remote Access checkbox exists to support Web App Developers who wish to remain using their familiar Mac and Windows web app development tools (such as Dreamweaver) on their existing machines.

"Advanced Configuration" opens the mysql configuration file using the OS's default text editor. If there are unsaved edits on any tab anywhere in this window, the user is prompted whether to Save those edits (yes/no buttons). Regardless of which answer is given, this window is closed before the text editor is opened.

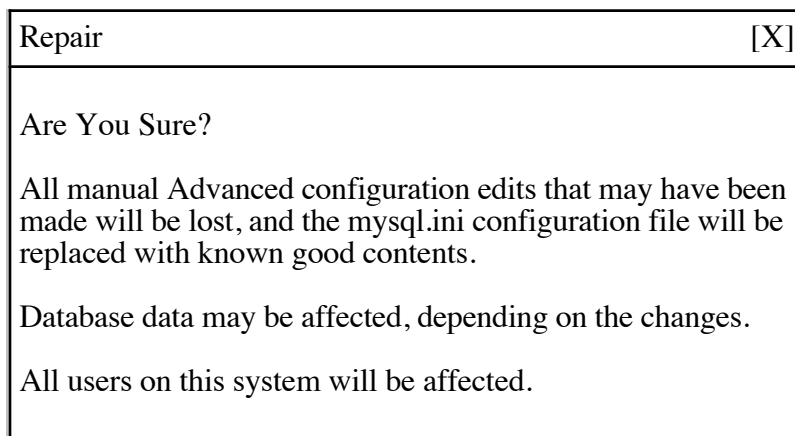
The Administer Databases & Tables button invokes phpMyAdmin in a new browser window; fails in the same way as the Start menu equivalent.

Repair:

Restores the configuration file to the state it was in immediately after the first Initialization. In other words, overwrites and repairs (corrects) any erroneous manual edits to the configuration file.

This feature is useful to restore messed up configurations.

Clicking this button yields:



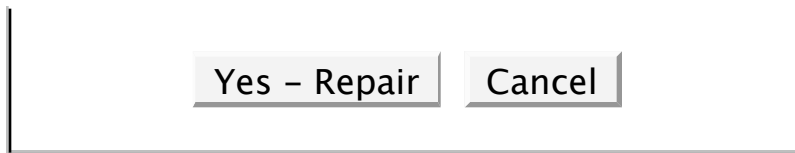
Repair [X]

Are You Sure?

All manual Advanced configuration edits that may have been made will be lost, and the mysql.ini configuration file will be replaced with known good contents.

Database data may be affected, depending on the changes.

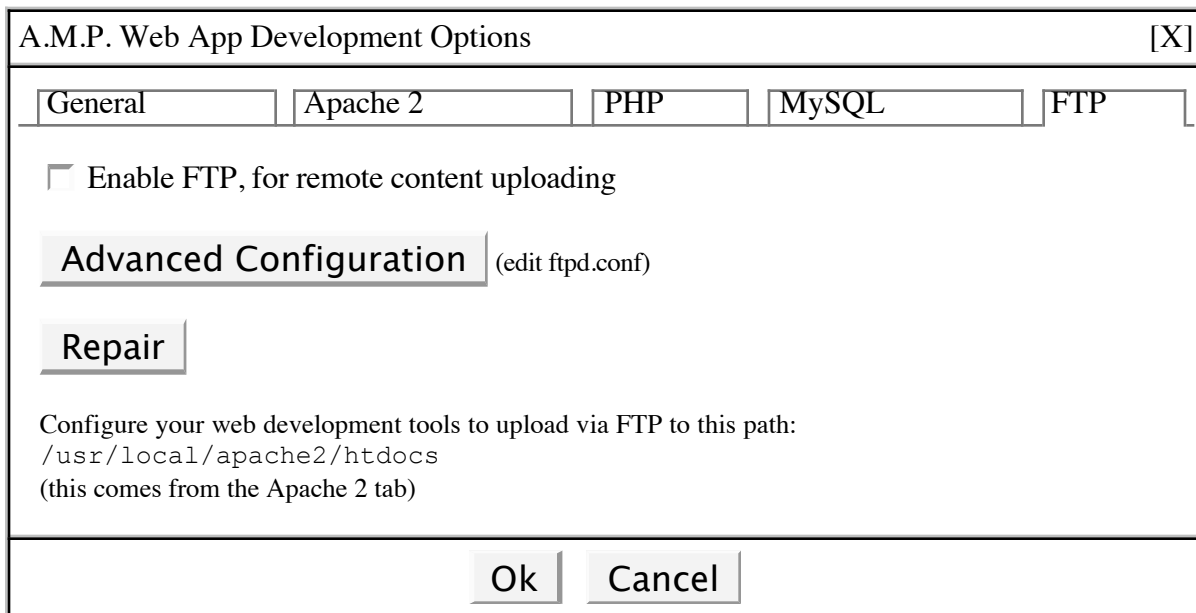
All users on this system will be affected.



As soon as edits are made anywhere in the window (in this tab or other tab(s)), the Advanced Configuration and Administer Databases & Tables buttons are disabled.

FTP tab

A UI to the built-in FTP daemon is provided to support Web App Developers who wish to remain using their familiar Mac and Windows web app development tools (such as Dreamweaver) on their existing machines.



The path shown echos the Document Directory as currently entered on the Apache 2 tab.

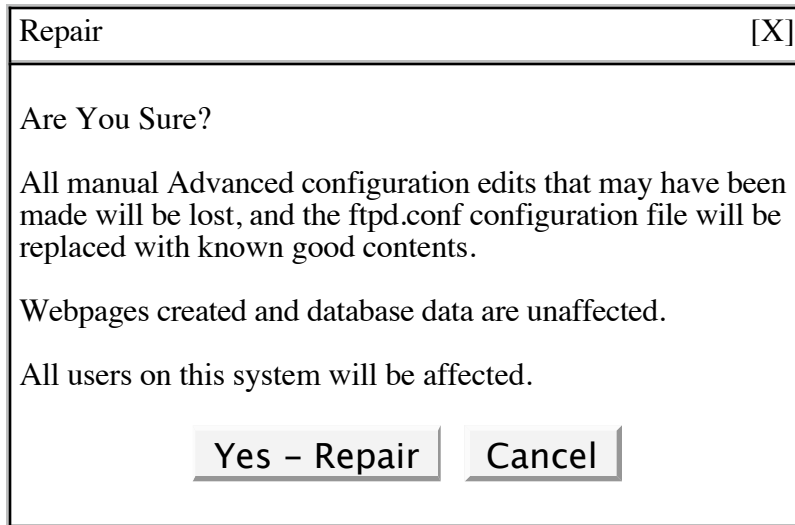
"Advanced Configuration" opens the ftp daemon's configuration file using the OS's default text editor. If there are unsaved edits on any tab anywhere in this window, the user is prompted whether to Save those edits (yes/no buttons). Regardless of which answer is given, this window is closed before the text editor is opened.

Repair:

Restores the configuration file to the state it was in immediately after the first Initialization. In other words, overwrites and repairs (corrects) any erroneous manual edits to the configuration file.

This feature is useful to restore messed up configurations.

Clicking this button yields:



As soon as edits are made anywhere in the window (in this tab or other tab(s)), the Advanced Configuration button is disabled.