Uniform Server – Coral

Uniform Server – Coral

Quick Start Guide

The Uniform Server's default configuration is set to allow only local access; users on your network or the Internet cannot access your server. This allows you to develop and view your website locally in a secure environment.

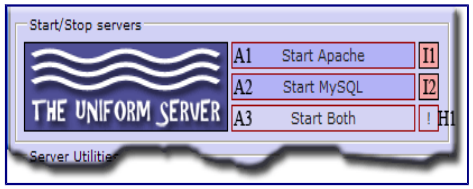
This guide shows you how to run the servers as standard programs. When run as standard programs, nothing is installed to the Windows registry. The servers are therefore fully portable.

Contents

Start the Servers

- **A)** Navigate to the UniServer folder
- Locate the file **Start_as_program.exe**
- Double click. This starts the server menu
- Click No to pop-ups (But see Set New MySQL root password following).

- **From Server Menu**
- Click **Start Both** button **(A3)**. Starts servers
- If challenged by your firewall, allow access
- A browser opens, displaying splash page.
- Buttons/Indicators change state



Note: There are various start-up conditions. If you receive a message unable to connect and are provided an option from your browser to try again, wait until all disk activity completes and click the "try again" button. Alternatively, wait a reasonable time to view the initial splash page, and enter the following into your browser: http://localhost/us_splash/index.php

If you do not have the VC9 library, when you attempt to start the servers, you will get a runtime error popup. See VC9 Libraries to resolve this.

Set New MySQL root password

Although you can skip setting a new MySQL root password, it is highly recommended that you set this password at the first opportunity. Well! This is the first opportunity; click **Yes** at the pop-up.

- The Set New MySQL Password menu opens (See image on right)
- **A)** Enter a new password
- **B)** Click the **Change Password** button.

Root folder www - Test

The root folder www contains a very simple web-site consisting of a single page named index.php.

Clicking menu button **View www** displays this page confirming that Apache is serving pages from this folder.

Add your Website or test pages

Copy your Website or test pages into the root folder www.

Before doing this you can delete the content of this root folder, with the exception of files **.htaccess** and **favicon.ico**.

- **.htaccess** - This file is part of the server security and should not be deleted.
- **favicon.ico** - This is a small image file displayed in a browser next to a Web page name. You can replace this with your own file. The name is important and should always be **favicon.ico**

View pages

The default server name is localhost. Use this to access pages as follows:

Index Pages	**Specific pages**
To access index pages (such as index.htm, index.html, index.php) type the following into your browser.	Display a specific page by appending its name to the server name. For example, if your test page is named test.php, then type the following into your browser:
http://localhost/	http://localhost/test.php

The Apache server automatically searches for index pages and displays the first one it finds. If it cannot find an index page, a directory listing is displayed instead.

Stop the Servers

Note: Server control buttons toggle between start and stop.

- **From the Server Menu**
- **A3)** Click **Stop Both button**. Stops servers
- Buttons/Indicators change state
- You can now close the server menu.

This quick guide shows how to start and stop the servers and display a test page. You can copy an entire web-site into the root folder www and it will then be served by Apache.

Quick Start Guide - More Detail

This guide adds more detail to the basic Quick Start Guide. It covers installation and how to run The Uniform Server as a program or service.

Contents

Installation

The Unform Server does not have a typical program installation. Instead, just execute the downloaded file (Coral_x_x_x.exe). The installation extraction defaults to creating "C:\UniServer", with the folder structure as shown on the right. Once extracted, the servers are ready to run.

While it may seem that by this point you have successfully completed the installation, there is one aspect that needs to be validated.

Server Paths

You can install The Uniform Server in nearly any location, with the exception that the ***path to folder UniServer must not contain spaces***. Do not use "C:\Program Files", for example (which has other problems as well, such as restricted permissions and system restore inconsistencies). We recommend installing to the drive root, such as "C:\" or "D:\" for simplicity, but other locations, such as "C:\test1\appserver" work just as well.

Note: The space character restriction is generic and applies to other WAMP stacks as well. Many applications that will be hosted on the server do not tolerate spaces in paths. The system restore issue applies mainly to individual component installs.

If the current installation is on a faulty path, the situation can easily be resolved by just moving the UniServer directory to a proper location. You do not have to delete the directory and reinstall. For example, just rename "D:\bad path\UniServer" to "D:\good_loc\UniServer".

Installing your Website or Test pages

Web pages are served from folder UniServer**www** (B), which is commonly referred to as the server root folder. Simply copy your Website into the root folder www.

Note 1: Unix installations use a root folder named **htdocs**. When using an application installation guide that refers to htdocs, use folder www instead.
Note 2: The file structure is not static. For example, running Vhost or database backup creates the new folders UniServer\vhosts or UniServer\db_backup respectively.

The root folder **www** already contains a very simple web-site consisting of a single page named index.php. Its purpose is to provide a test page, allowing you to confirm that the server is functioning properly.

Before installing your Website, you can delete the index.php file. Don't delete the .htaccess and favicon.ico files. They have the following purposes:

- **.htaccess** - This file is part of the server security and should not be deleted.
- **favicon.ico** - This is a small image file displayed in a browser next to a Web page name. You can replace this with your own file, but use the name **favicon.ico** so browsers will display it properly.

Note: Having a favicon image is not a requirement, but modern browsers attempt to load this image by default. If your server does not contain one, an error is logged in Apache's log file.

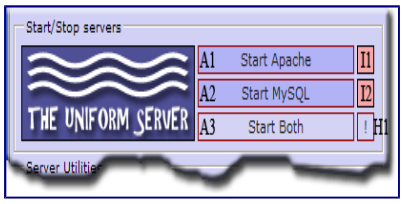
Server Control

The UniServer folder contains two files **Start_as_program.exe** and **Start_as_service.exe** which open The Uniform Server's control menu for running the servers either as a standard program or as a service respectively. This control menu is referred to as UniController.

Run as Program

- **Start_as_program.exe** Opens a menu that runs the servers as a standard program. When using this menu nothing is installed to your operating system's registry. This is ideal for a development server, especially when running from a USB memory stick, since you do not have to perform anything special to stop or remove the servers.

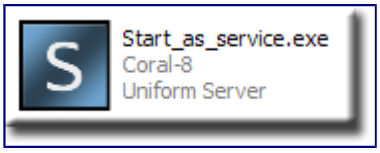
- **Run servers** - Click the button "Start Both" (A3). This starts both servers. Some pop-ups are initially displayed; you may click "no" to these (but see Set New MySQL root password which follows). If challenged by your firewall, allow access for both Apache and MySQL. Your browser will automatically start and display the



server splash page. Be aware that this is **not** the index.php file in wwww.

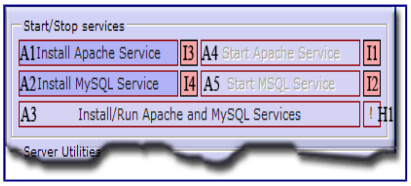
- **Your Web site** - If you installed a Website in root folder www containing either an index.html or index.php page, clicking View www will display your page instead of the test page.

Run as Service



- **Start_as_service.exe** Opens a menu that runs the servers as a service. The purpose of running them as a service is that when you restart your PC, the servers will automatically re-start. Using this menu you add the servers' program information to your operating system's registry. Although you can install and run services from a USB memory stick, you must remember to stop and uninstall the services before removing your USB stick.

NOTE: For Windows 7 and Vista, you must have admin privileges.



- **Install and Run services** - Click the button "Install/Run Apache and MySQL service" (A3). This installs and runs both servers. Some pop-ups are initially displayed; click no to all pop-ups (but see Set New MySQL root password following). If challenged by your firewall allow access for both Apache and MySQL. Your browser will automatically start and display the server splash page.
- **Your Web site** - If you installed a Website in root folder www containing either an index.html or index.php page, clicking **View www** will display your page instead of the test page.

Server Utilities

Although there are a bewildering number of menu items. Running the servers either as a standard program or as a service requies just a single button click as explained above. UniController has several additional menu items, providing independent control of many server features and functions.

Server utilities:

These menu items provide useful features, such as single button access to PHPMyAdmin and server configuration.

- **B7) View www** - Displays your website in the default browser. It's equivalent to typing the following into a browser: **http://localhost/**

Important - Security Feature

The servers are safe to use even when connected to the Internet, since they are locked down, allowing only local access. In other words, the servers are accessible only to the PC on which they are running. Internet and Intranet access are denied.

In order to allow access to your server by other users (Internet or Intranet), you must manually enable the required functions, thereby putting your servers on-line.

Set New MySQL root password

Although you can skip setting a new MySQL root password it is recommended you set this password at the first opportunity.
Well! This is the first opportunity; click **Yes** at the pop-up.

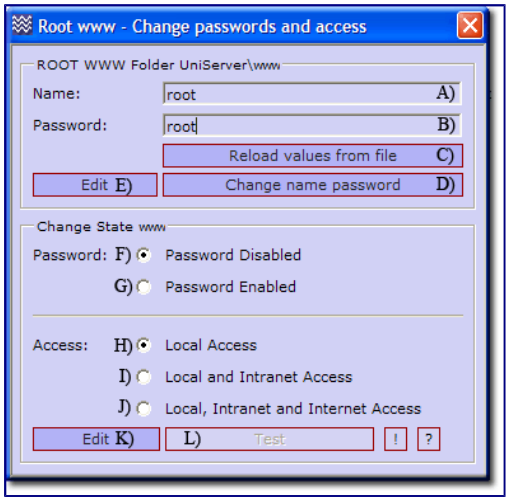
- Set New MySQL Password menu opens (See image on right)
- **A)** Enter a new password
- **B)** Click the **Change Password** button.
- **Note:** You can re-open this menu from UniController as follows:

Server Configuration > MySQL > Change Password

How to put the servers on-line

Putting your servers on-line takes just a few clicks as follows:

- On the server menu, click "**Server Configuration**" button (B4 in Server Utilities above)
- In the new window click dropdown menu "**General**"
- From the dropdown click "**Root www - Change password and access**"
- In the new window click "**Local, Intranet and Internet Access**" (J) see image on right.



If your servers are accessible from the Internet, users will now be able to view your web-site.

Note 1:

The **Access** section controls who can gain access to your server.
Three radio buttons H, I and J provide Local, Intranet and Internet access respectively.

Note 2: A similar menu is provided for ssl folder control.

Server Configuration > General > Root www - Change password and access > MySQL > Change Password

How to check Server Internet Access

Clicking the **Server Status** menu button opens a pop-up displaying a list of the current server parameters and their status. Server Internet accessibility is provided by the first two items. The first provides your IP address as seen from the Internet while the second confirms whether your server is accessible.

- Servers must be running and connected to Internet
- Run server status, click **Server Status** button (B2 see Server Utilities above)
- IP address displayed with green indicator (A) Confirms IP
- Accessible from Internet **= Yes** with green indicator (B) confirms server is accessible Note: Your server, if it is accessible, will pass this test even if you have restricted access to local access only.

If you have put your servers on-line, any user on the Internet now has access to your server by typing the following into their browser address bar: **http://xxx.xxx.xxx.xxx/** (replace the x's with your IP address)

Possible Problems

IP Address: Red indicator and no ip address. Your PC is not connected to the Internet!

Access: Red indicator and accessible from Internet **= No**

- Must be connected to Internet and an IP address displayed (A).
- Your PC is connected to a router. You must forward port 80 in the router settings.
- For Windows 7 and Vista, check the firewall for public access to the servers.
- Still no access? Is your service provider blocking port 80?

User Configuration

The Uniform Server Coral uses a configuration file which is split into two sections: user defined configuration and server tracking. The server tracking section is automatically updated by UniController and should not be changed by a user. It tracks items such as the installation path and the current Apache and MySQL running status.

This page covers the user defined configuration section. You can disable or enable the start up splash page and optionally set a new page to display. During start up, a user is nagged to change server security items such as the MySQL password. Nagging is enabled or disabled in this section. Two user buttons, www and ssl, have been pre-configured to display the root folder index pages; these can be re-configured to display specific pages. In addition each button text is changeable by setting a language file constant.

Contents

Splash Page

Background

At start-up, the server displays a splash page confirming that it is working. It is only a single page mini web-site with associated image and css folders. This mini-site is in its own folder (UniServer\home\us_splash) and mapped as an alias. The folder contains an .htaccess file limiting access only to localhost.

This mini web-site may be changed to fit your own requirements. Alternatively it can be disabled in the control configuration file as explained below. After that, it can be deleted, and the alias section removed from the configuration file.

Enable/Disable splash page

A single line in the control configuration file enables or disables the splash page at start-up. Edit file:

UniServer\coral_con\config_menu\config_tracker.ini

Locate this section and change line **start=true** as appropriate :

```
[REDIRECT]
;Redirect allows you to specify a page to display either
;at startup or when www or ssl buttons pressed.
;Folders can be an Alias eg us_splash. Root folder assumed
;If no path or file specified root is assumed first index
;page is displayed. No index page found the root folder
;content is displayed.

; Display page at start up. true display page false no page
start=true
```

Change Splash Page

After enabling the start-up splash page, you can set the specific page to be displayed. Locate this section in the configuration file and change the line **display_page=us_splash/index.php** accordingly:

```
; Page to display from folder alias us_splash
; Default index.php
display_page=us_splash/index.php
```

For example, if you want to display the index.php page in root folder www change line to: **display_page=index.php**

Note 1: The default page displayed uses redirection and is constructed with the following format: **http://localhost:80/display_page** (display_page is the variable containing path/file to display.)

Note 2: This part of the **URL http://localhost:80/** is automatically adjusted to match the server

port configuration.

User buttons www and ssl

Both server menus, initiated by start as program and start as service, contain two user configurable buttons:

- **B7)** View www
- **B8)** View ssl

These have been pre-configured to display the root folder index pages. Their corresponding button text is defined in the language file

Change www page

A single line in the control configuration file sets the page to be displayed. Edit file:

UniServer\coral_con\config_menu**config_tracker.ini**

Locate this section in the configuration file and change line **"www="** accordingly:

```
; Page to display from folder www when www button clicked
; Default any index page. Note: http:// is used
www=
```

For example, if you want to display the index.php page in root folder www change line to:
www=**index.php**

Note 1: The page displayed uses redirection and is constructed with the following format: **http://localhost:80/www** (www is the variable containing path/file to display.)

Note 2: This part of the URL **http://localhost:80/** is automatically adjusted to match server configuration.

Change ssl page

A single line in the control configuration file sets the page to be displayed. Edit file:

UniServer\coral_con\config_menu**config_tracker.ini**

Locate this section in the configuration file and change line **"ssl=index.php"** accordingly:

```
; Page to display from folder ssl when ssl button clicked
; Default index.php Note: https:// is used
ssl=index.php
```

For example, if you want to display the fred.html page in root folder ssl change line to:
ssl=**fred.html**

Note 1: The page displayed uses redirection and is constructed with the following format:

https://localhost:443/ssl (sll is the variable containing path/file to display.)

Note 2: This part of the URL **https://localhost:443/** is automatically adjusted to match server configuration.

Change button names (www or ssl)

Button names are defined in the language configuration files. The actual file used is lang.vbs, but this is just a copy of the active language. When a language is selected with the Select Language button, that language file is copied to lang.vbs and replaces it.

Currently there are two language files; other languages will be added as translations are made available.

For English, edit:

UniServer\coral_con\lang**English.vbs**

Locate this section in the configuration file and change lines accordingly:

```
'*****************************************************************************
' User Configuration
'*****************************************************************************

Const lang_view_www   = "View www"   'User button text
Const lang_view_ssl   = "View ssl"   'User button text

'------------------------------------------------- End User Configuration ---
```

For example, if you want to change the displayed text "View www" to "My Index" and change "View ssl" to "Information", then the block becomes:

```
'*****************************************************************************
' User Configuration
'*****************************************************************************

Const lang_view_www   = "My Index"     'User button text
Const lang_view_ssl   = "Information"  'User button text

'------------------------------------------------- End User Configuration ---
```

Note:
To activate your change, you can either duplicate the change to lang.vbs or use the Select Language button, which will do the copy for you.

Start-up - Nagging

I personally hate the concept of nagging users, hence the ability to turn it off when first encountered. That said, it is essential to set a new MySQL server root password, and nagging stresses the security issue.

Enable/Disable - Nagging

If you accidentally turn off nagging, it is easy to restore by editing the control configuration file.

UniServer\coral_con\config_menu**config_tracker.ini**

Search for this section:

```
[NAG]
; Enables nagging to get user to change MYSQL password
; Default true - nag user
nag_user=false
```

Change the section as shown to reenable nagging:

```
[NAG]
; Enables nagging to get user to change MYSQL password
; Default true - nag user
nag_user=true
```

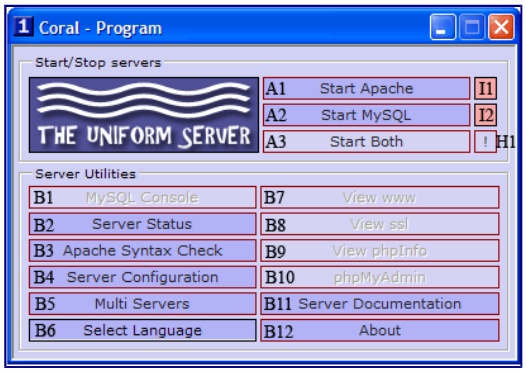
Run as a program

Folder UniServer contains a file **Start_as_program.exe** which opens the server control menu UniController. This menu allows you to run The Uniform Server as a standard program.

Contents

UniController Menu

When using this menu, nothing is installed in your operating system's registry. This is ideal for a development server, especially when running from a USB memory stick, since you do not have to perform anything special to stop or remove the servers.



- **Run servers** - Click button **Start Both** (A3). This starts both servers. If challenged by your firewall allow access. Your browser will automatically start and display the server splash test page. Buttons A1, A2 and A3 toggle to display "stop". The indicators change from red to green when the servers are running.

- **Stop servers** - Click button **Stop Both** (A3). This stops both servers. The server menu returns to the initial state as shown on the right.

- Buttons A1 and A2 provide independent control of the Apache and MySQL servers respectively.

Interlinked server control

The server control menus are interlinked; this prevents you running both menus at the same time. If you have a single server Apache or MySQL running as a service and close the “run as a service” menu and then open the “run as a program menu”, you will find its corresponding server buttons are greyed out.

Similarly the reverse is true. A single server Apache or MySQL running as a program will have its corresponding server buttons greyed out in the “run as a service menu”

This interlinking applies when running multi-servers; the first server started (Apache or MySQL) will have its corresponding buttons greyed out in all other server menus. This means you can have a single instance of the MySQL server running and have several different Apache servers running, all accessing the one MySQL server.

Run as a service

Folder UniServer contains a file Start_as_service.exe which opens the server control menu UniController. This menu allows you to install and run The Uniform Server as a service.

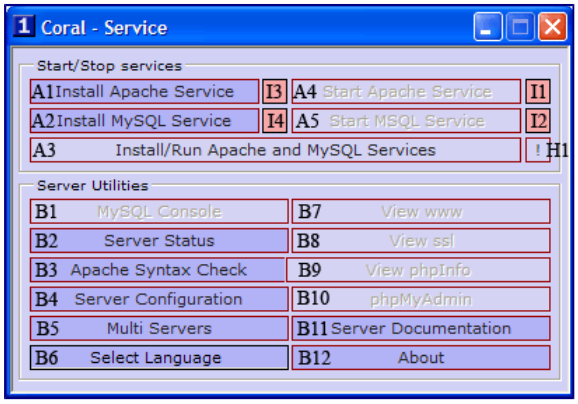
Contents

UniController Menu

When using this menu, you install the servers as services; it adds data to your operating system's registry. The dvantages of running as a service, when you restart your PC, servers' will automatically re-start. You can install and run services from a USB memory stick however, you must remember to stop and uninstall services before removing your USB stick.

NOTE: For Vista and Win7, admin privileges are required. To run as administrator, right click on **Start_as_service.exe** and click Properties and then choose run as administrator.

- **Install and Run services** - Click button **Install/Run Apache and MySQL service** (A3). This installs and runs both servers. If challenged by your firewall allow access. Your browser will automatically start and display the server splash test page. Buttons A1, A2 and A3 toggle to display uninstall. Buttons A4 and A5 are enabled and display stop service. Indicators change from red to green when services have been installed and are running.



- **Stop and Uninstall services** - Click button **Stop/Uninstall Apache and MySQL service** (A3). This stops and uninstalls both servers. Server menu returns to initial state as shown at right.

- Buttons (A1,A4) and (A2,A5) provide independent control of Apache and MySQL servers respectively.

Interlinked server control

The server control menus are interlinked; this prevents you running both menus at the same time. If you have a single server Apache or MySQL running as a service and close the "run as a service" menu and then open the "run as a program menu", you will find its corresponding server buttons are greyed out.

Similarly the reverse is true. A single server Apache or MySQL running as a program will have

its corresponding server buttons greyed out in the “run as a service menu”

This interlinking applies when running multi-servers; the first server started (Apache or MySQL) will have its corresponding buttons greyed out in all other server menus. This means you can have a single instance of the MySQL server running and have several different Apache servers running, all accessing the one MySQL server.

Server Utilities

Both UniController menus (started by run as program or run as a service) have a menu section titled Server Utilities which provides useful features, such as single button access to PHPMyAdmin and server configuration.

B1 - MySQL Console

The **MySQL console** button opens a command window and changes the working directory to the MySQL binary folder. From here you can login to the MySQL server and open a MySQL prompt.

Logging into the server requires a user name and password. The defaults are root and root (root user account). At the first opportunity you should have changed this password! If you have forgotten the root user password, it is retrievable in one of two ways.

- Open file UniServer\coral_con**config_tracker.ini** and look for section **[MYSQL]** line **password =** .
- Or from UniController: **Server Configuration** (B4) > **MySQL** > **Change Password** - The password is displayed in the bottom left. Do not make any changes; just close the window.

Login to MySQL console

- At the command prompt, enter: **mysql –uroot –proot**
- Quit MySQL monitor by typing: **exit**

B2 - Server Status

Clicking the **Server Status** menu button opens a pop-up that displays a list of current server parameters.

General Status

- Internet accessibility is indicated by the first two items. See Quick start details for more.
- Newer UniServer version. A server **version displayed** means that version is newer and available for download. **Current** displayed your server is up to date. **Unavailable** indicates, unable to contact the web-server or there was an error.
- Cron is a built in scheduler. It runs pre-defined scripts periodically. By default, it is not running; you can either start it manually or install and run it as a service.
- PHP configuration file (php.ini) shows as either Production or Development. The default is that it is a copy of Production.

Apache Server & MySQL Server

- Displays ports as set in the configuration file.
- Executable file name changes (digit change) when you run multi-servers (default as shown).
- Service name changes (digit change) when you run multi-servers (default as shown).
- Displays server run status.
- Displays server install status: either as program, service or free to install.

Port Status

- Displays the current port status.
- Free – No processes are using this port.
- In use by this server – The current server instance is using this port.
- In use by another server - Another process is using the port.

B3 - Apache Syntax Check

The **Apache Syntax Check** button initiates a syntax check on Apache's configuration files. The results are displayed in a pop-up message box. Any errors found require correction before the Apache server will start. The check is equivalent to the following manual procedure:

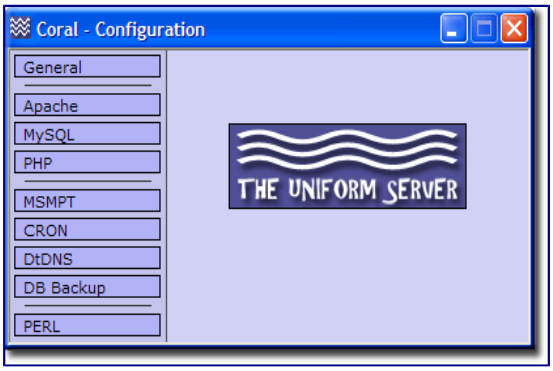
1. Start > All Programs > Accessories > Command Prompt - Opens a command window
2. In the command window type: **cd C:\UniServer\usr\local\apache2\bin**
3. In the command window type: **httpd1.exe -t**

Note 1: The path to the Apache binary in step 2 may be different depending on where you installed The Uniform Server

Note 2: If you previously ran multi-servers, the Apache binary httpd1.exe in step 3 will have a different digit.

B4 - Server Configuration

Clicking the **Server Configuration** button opens a pop-up displaying the server configuration menu.



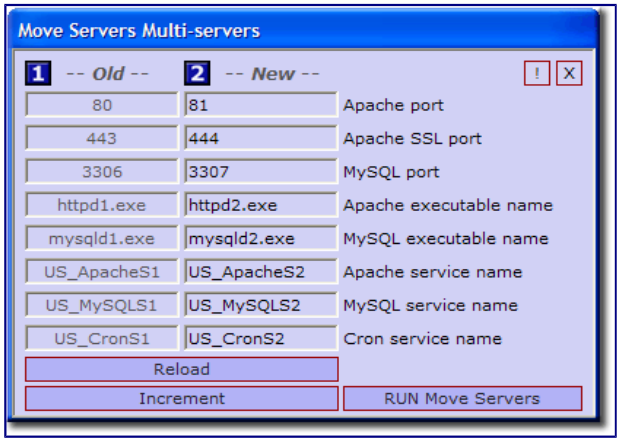
- Clicking each menu heading opens a sub-menu.
- Each sub-menu runs either a configuration menu or opens the default text editor to display or to edit the configuration file.

Note:

Configuration menu headings are reflected in documentation top left menu. Documentation menu also opens sub-menu items corresponding to configuration items.

B5 - Multi Servers

Multi-servers enables you to run more than one copy of The Uniform Server (Apache and MySQL) on the same PC. To achieve this, the executables and services are renamed and the ports the servers run on are changed.



Values under the **old** column are the current server values. After clicking “Run Move Servers” these values are overwritten in their corresponding configurations files with the values shown in the **new** column.

Changes are reflected in the tray icon, which is numbered 1 to 9. This number generally corresponds to the digit appended to executable and service names. For server sets past 9 the tray icon will not use a digit.

Note 1: Each server requires its own separate directory. For example, Server set 1 and Server set 2 may have the following directories: C:\UniServer1 and C:\UniServer2.
Note 2: Normally you run this menu on a newly extracted copy of The Uniform Server, change the directory name, and then add any applications using appropriate configuration parameters.

B6 - Select Language

This drop down menu displays the currently supported languages.

Language files are contained in folder UniServer\uni_con\lang. This folder is scanned for all language files (with the exception of the active language file, lang.vbs). The names are then listed in the dropdown menu. Selecting a language overwrites the lang.vbs file with the appropriate language.

A language file name has a corresponding folder name in the documentation folder UniServer\docs. Any references to document pages are automatically switched to the appropriate language folder.

Note: You are welcome to submit a language file for inclusion in the next release. Use file UniServer\uni_con\lang\English.vbs as a template.

B7 - View www

This is a user definable button. The default action is to open an html page in the default browser, which then redirects to an index page served by Apache.

- UniServer\tmp\redirect.html - Location and name of redirect file dynamically updated.
- url=http://localhost:80/index.php" - Redirected to this page on server

B8 - View ssl

This is a user definable button. The default action is to open an html page in the default browser, which then redirects to an index page served by Apache.

- UniServer\tmp\redirect.html - Location and name of redirect file dynamically updated.
- url=https://localhost:443/index.php" - Redirected to this page on server

B9 - View phpInfo

This menu button opens an html page in the default browser, which redirects to a page served by Apache. This in turn runs the PHP function phpinfo() displaying a wealth of information relating to PHP. The page provides very detailed information, and a menu splits this into smaller sections, making it easier to read.

- UniServer\tmp\redirect.html - Location and name of redirect file dynamically updated.
- url=http://localhost:80/us_extra/phpinfo.php" - Redirected to this page on server

PHP Info Menu

- Configuration - Displays PHP Core configuration.
- Enviroment - PHP current environment
- Modules - List all modules
- Extensions - Overview and details (includes functions) for all loaded extensions.

- Variables - List all currently available PHP variables.
- General - Intro information

B10 - phpMyAdmin

phpMyAdmin is a MySQL admininstration tool written in PHP. It allows you to perform tasks such as creating, modifying or deleting databases. You can execute SQL statements and manage users and their permissions.

The **phpMyAdmin menu** button opens an html page in the default browser, which redirects to the phpMyAdmin index page served by Apache.

- UniServer\tmp\redirect.html - Location and name of the dynamically updated redirect file.
- url=http://localhost:80/us_phpmyadmin/index.php" - Target of redirection on the server

phpMyAdmin Access
By default phpMyAdmin is accessible only to localhost. To allow access for external users (Intranet and Internet), refer to the phpMyAdmin Access page for details.

B11 - Server Documentation

The Server Documentation button opens the index.html page from which the pages you are reading can be accessed. It follows the structure of the UniController menu. The top level describes commonly used functionality while sub-menus detail the next level of functionality and provide more detailed information. It is self-contained, requiring no external references. This notwithstanding, there is quite a bit of additional information available at The Uniform Server's Wiki site wiki.uniformserver.com

B12 - About

About provides an overview of "UniController", in particular why it was implemented in VbScript and HTAs. In addition to providing support for different languages, a user can easily tailor and modify the menu structure to meet their differing requirements.

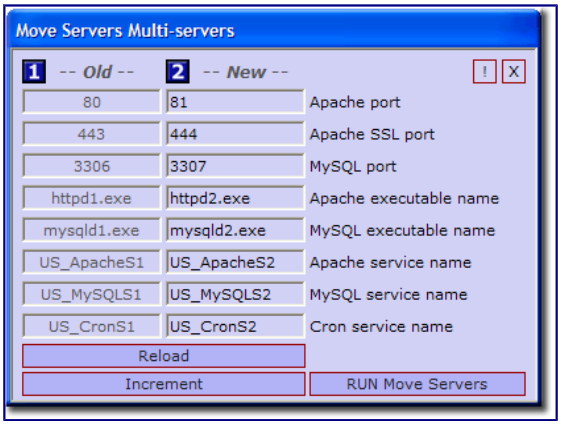
Multi-Servers

Multi-servers enable you to run more than one copy of The Uniform Server (Apache and MySQL) on the same PC. Moving servers to different ports also allows you to run The Uniform Server alongside other servers such as IIS. It also allows you to have multiple separate instances of The Uniform Server running at the same time. This page details the changes made when the Multi-servers menu is invoked.

NOTE: It it best to run this procedure on a newly-installed version of The Uniform Server, since applications will be unaware of the changes. If moved after installation, many applications will require changes to database fields in order to run properly.

Contents

Multi Servers



Multi-servers enables you to run more than one copy of The Uniform Server (Apache and MySQL) on the same PC. To achieve this, the executables and services are renamed and the ports the servers run on are changed.

Values under the **old** column are the current server values. After clicking “Run Move Servers” these values are overwritten in their corresponding configurations files with the values shown in the **new** column.

Changes are reflected in the tray icon, which is numbered 1 to 9. This number generally corresponds to the digit appended to the executable and service names. The number of separate server sets is not limited to 9, but subsequent sets will not show a digit.

Note:
Normally you would run this menu on a newly extracted copy of The Uniform Server and then add any applications using appropriate configuration parameters. Changing the applications after moving the ports is difficult and error-prone.

When installing the applications, you will need to explicitly specify the new ports on any URLs, such as in http://localhost:83 (site location or server name) and 127.0.0.1:3308 (MySQL).

Files modified

After running Multi-servers the following files are updated:

Apache

- UniServer\usr\local\apache2\conf**httpd.conf**
- UniServer\usr\local\apache2\conf\extra**httpd-vhosts.conf**
- UniServer\usr\local\apache2\conf\extra**ssl.conf**

MySQL

- UniServer\usr\local\mysql**my.ini**
- UniServer\usr\local\mysql**small_my.ini**
- UniServer\usr\local\mysql**medium_my.ini**

PHP

- UniServer\usr\local\php**php.ini**

UniController

- UniServer\uni_con**uni_con_program.hta**

- UniServer\usr\local\php**php.ini_development**
- UniServer\usr\local\php**php.ini_production**

- UniServer\uni_con**uni_con_service.hta**
- UniServer\uni_con**config_tracker.ini**

Cron

- UniServer\uni_con\cron_service**srvstart.ini**
- UniServer\uni_con\cron_service**start_automatic.vbs**
- UniServer\uni_con\cron_service**z_install.bat**

Cron

- UniServer\uni_con\cron_service**z_service_start.bat**
- UniServer\uni_con\cron_service**z_service_stop.bat**
- UniServer\uni_con\cron_service**z_uninstall_service.bat**

After running Multi-servers the following executables are renamed. Note: x is new digit appended to executable name.

- UniServer\usr\local\apache2\bin\httpd1.exe to UniServer\usr\local\apache2\bin\httpd**x**.exe
- UniServer\usr\local\mysql\bin**mysqld1.exe** to UniServer\usr\local\mysql\bin\mysqld**x**.exe

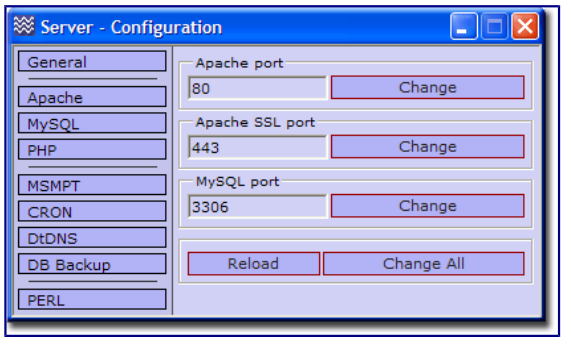
General - Introduction

This dropdown menu groups items that do not easily fit into other categories.

Contents

Change Apache and MySQL ports

UniController: Server Configuration > General > Change ports, Apache and MySQL



Change a Single port

- Current port setting is displayed.
- Change the port to required value.
- Click the corresponding **change** button.

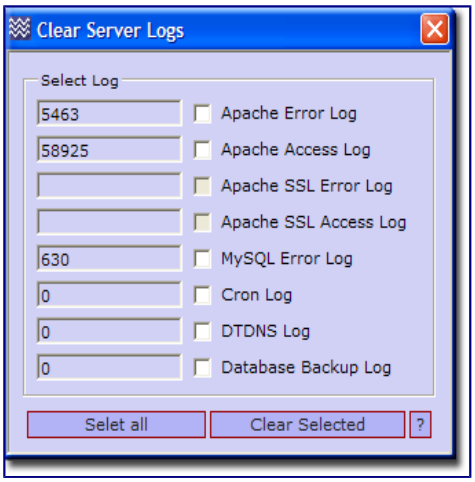
Change all ports

- Change all ports to required value.
- Click **change** all button.

Note 1: For new ports to become effective you must restart the servers.
Note 2: For detailed information see Change Ports
Note 3: Normally you should run the multi-server script to change ports. Use this for specific port adjustments to resolve conflicts.

Clear server log files

UniController: Server Configuration > General > Clear Server log files



Clear a Single Log file

- Select the log file to delete (Tick check box).
- Click **Clear Selected** button.

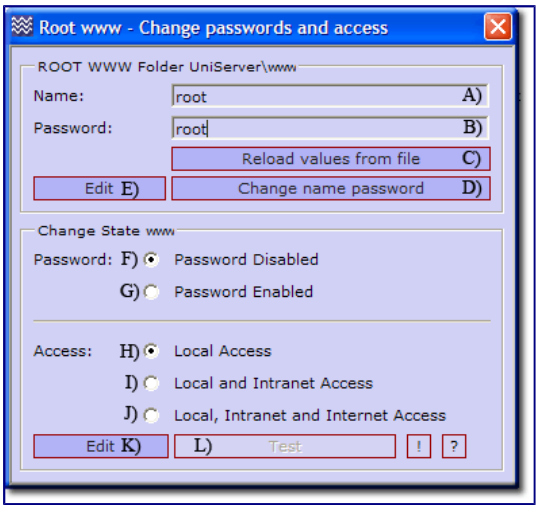
Clear All log files

- Click Select All button.
- Click **Clear Selected** button.

Note 1: The current file size is displayed.
Note 2: To clear Apache and MySQL logs, you must first stop the servers.
Note 3: For detailed information see Clear Logs

Root folder www - Access

UniController: Server Configuration > General > Root www - Change passwords and access



This menu is split into three sections as follows:

1) The top section allows you to change the access name and password; the defaults are set to root.

- Generally for a private server a single name and password are used.
- These are displayed and changeable using buttons A) B) D)
- Note: When using a list, only the first entry is displayed and directly changeable.
- Using a list of names and passwords requires that you directly edit the associated name-password configuration file. Use edit button E) to open the file in notepad. The Reload values button C) displays just the first entry in list.

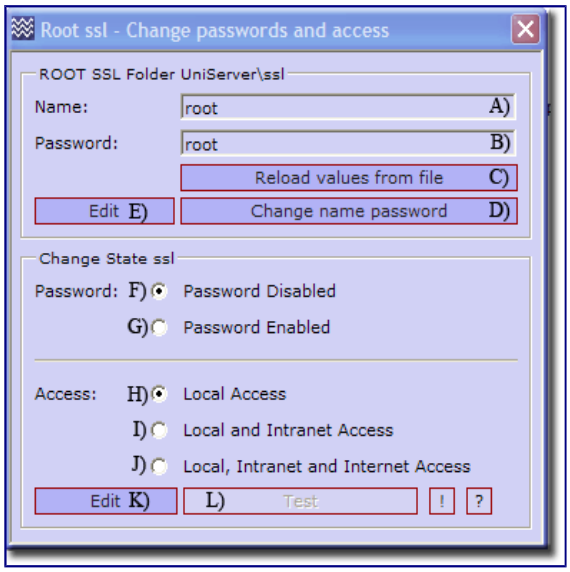
2) Change state: enables or disables the use of a password to access root folder www. Default is disabled.

3) Access section: allows you to set who can gain access to root folder www. Default is local access only.

Note 1: Server access is controlled using an .htaccess file. This can be directly edited using Edit button K)
Note 2: Apache does not require restarting all changes take effect immediately.
Note 3: Test button L) opens the index page in the default browser allowing you to quickly test changes.
Note 4: For detailed information see Root folder www access

Root folder ssl - Access

UniController: Server Configuration > General > Root ssl - Change passwords and access



This menu is split into three sections as follows:

1) The top section allows you to change the access name and password; the defaults are set to root.

- Generally for a private server a single name and password are used.
- These are displayed and changeable using buttons A) B) D)
- Note: When using a list, only the first entry is displayed and directly changeable.
- Using a list of names and passwords requires that you directly edit the associated name-password configuration file. Use edit button E) to open the file in notepad. The Reload values button C) displays just the first entry in list.

2) Change state: enables or disables the use of a password to access root folder ssl. Default is disabled.

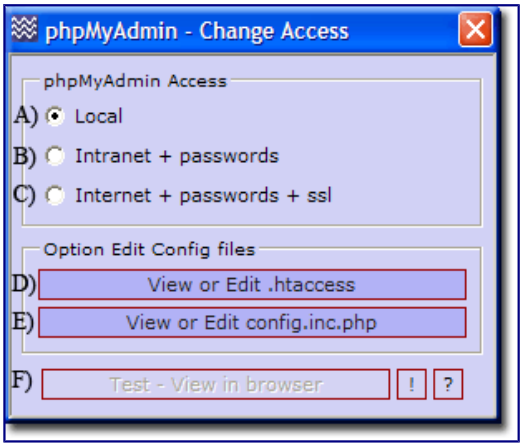
3) Access section: allows you to set who can gain access to root folder ssl. Default is local access only.

Note 1: Server access is controlled using an .htaccess file. This can be directly edited using Edit button K)
Note 2: Apache does not require restarting. All changes take effect immediately.
Note 3: Test button L) opens the index page in the default browser allowing you to quickly test changes.

Note 4: For detailed information see Root folder ssl access

Root phpMyAdmin Access

UniController: Server Configuration > General > Root phpMyAdmin - Change Access



- A) **Local** access is the default and prevents both Intranet and Internet users access to phpMyAdmin. Only localhost may connect. It allows the root MySQL administrator direct access without the need to enter a name or password. For security reasons you must change the default MySQL password this prevents accidental cross-site scripting.
- B) **Intranet + passwords**. Anyone on your Intranet is allowed access to phpMyAdmin, however to gain access to the MySQL server requires a name and password. A user name and password are stored on the MySQL server for each restricted user created. A restricted user is one that is assigned access only to specific databases, and also has limited privileges for them.
- C) **Internet + passwords +ssl** This is similar to Intranet, except that all transactions are performed over a secure encrypted connection using ssl. If you have not installed or created a server certificate, clicking this radio button will produce a warning and the access selection will not change. This security feature forces you to use ssl and prevents sending names and passwords over the Internet in plain text.
- D) phpMyAdmin root folder is protected by an **.htaccess** file. This menu button opens this file in the default text editor allowing you to view or edit the file.
- E) phpMyAdmin is configured using the user configuration file **config.inc.php**. This menu button opens this file in the default text editor allowing you to view or edit the file.
- F) This menu button opens phpMyAdmin in the default browser. A greyed out button indicates that Apache is not running and its function are unavailable.

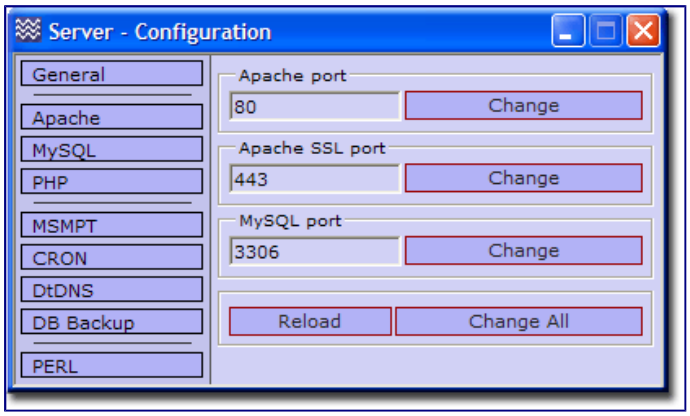
General - Change Ports

The Uniform Server provides two ways of changing the ports used by the servers. Normally you should run the multi-server script. This automatically increments all the ports and renames the executables. However there are situations where you need to change individual server ports. This menu allows you to specify the Apache (standard and secure) and MySQL ports.

Contents

Change Apache and MySQL ports

UniController: Server Configuration > General > Change ports, Apache and MySQL



Change a Single port

- Current port setting is displayed.
- Change the port to required value.
- Click the corresponding **change** button.

Change all ports

- Change all ports to required value.
- Click **change** all button.

Note: For new ports to become effective you must restart the servers.

Files modified

The Uniform Server scripts that use server ports will read them directly from the appropriate configuration file. The following files are updated when a port is changed:

Apache standard port

- UniServer\usr\local\apache2\conf**httpd.conf**
- UniServer\usr\local\apache2\conf\extra**httpd-vhosts.conf**

Apache ssl port

- UniServer\usr\local\apache2\conf\extra**ssl.conf**

MySQL standard port

- UniServer\usr\local\mysql**my.ini**
- UniServer\usr\local\mysql**small_my.ini**
- UniServer\usr\local\mysql**medium_my.ini**

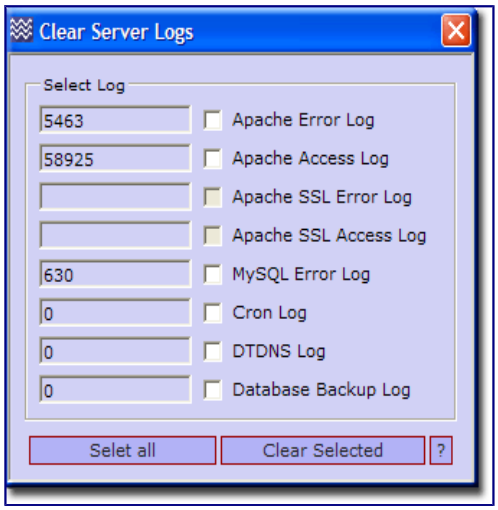
General - Clear Logs

Log files are an important resource when developing or debugging applications. Over time, log files can become very large. Generally there is no need to keep these. When running servers from a USB memory stick, freeing up space is essential. This menu allows you to delete all or just certain log files.

Contents

Clear server log files

UniController: Server Configuration > General > Clear Server log files



Clear a Single Log file

- Select the log file to delete (Tick check box).
- Click **Clear Selected** button.

Clear All log files

- Click Select All button.
- Click **Clear Selected** button.

Note 1: The current file size is displayed.
Note 2: To clear Apache and MySQL logs, you must first stop the servers.
Note 3: To clear Apache and MySQL logs, you must first stop the servers.

Files modified

Depending on what is selected, the following files are modified:

- UniServer\usr\local\apache2**logserror.log**
- UniServer\usr\local\apache2\logs**access.log**
- UniServer\usr\local\apache2\logs**error_ssl.log**
- UniServer\usr\local\apache2\logs**access_ssl.log**
- UniServer\usr\local\mysql\data**mysql.err**
- UniServer\uni_con\cron**cron.log**
- UniServer\uni_con\dtdns_updater**dtdns.log**
- UniServer\uni_con\db_backup**db_backup.log**

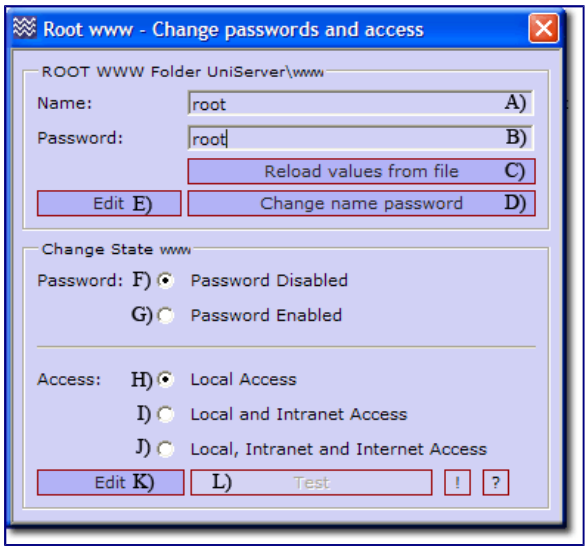
Root folder www - Access

All public pages are served from root folder www (also known as htdocs in a Unix environment). By default this folder is locked down, only allowing local access. Opening up your server for public or intranet access and optionally restricting access using a name and password are described below.

Contents

Overview

UniController: Server Configuration > General > Root www - Change passwords and access



This menu is split into three sections as follows:

1) The top section allows you to change the access name and password; the defaults are set to root.

- Generally for a private server a single name and password are used.
- These are displayed and changeable using buttons A) B) D)
- Note: When using a list, only the first entry is displayed and directly changeable.
- Using a list of names and passwords requires that you directly edit the associated name-password configuration file. Use edit button E) to open the file in notepad. The Reload values button C) displays just the first entry in list.

2) Change state: enables or disables the use of a password to access root folder www. Default is disabled.

3) Access section: allows you to set who can gain access to root folder www. Default is local access only.

Note 1: Server access is controlled using an .htaccess file. This can be directly edited using Edit button K)
Note 2: Apache does not require restarting all changes take effect immediately.
Note 3: Test button L) opens the index page in the default browser allowing you to quickly test changes.
Note 4: For detailed information see **Root folder www access**

Background information

There are three ways of restricting access to documents: either by IP address, hostname or by asking for a username and password. The first two are very similar and can be used to restrict access on an intranet. The last option allows users to be restricted on an individual basis using a username and password before being allowed access. This process is commonly referred to as user authentication. The Uniform Server can restrict access using both IP address and user authentication.

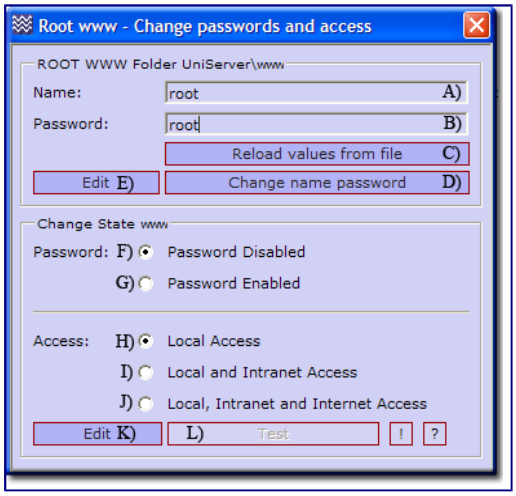
Setting up user authentication requires two steps. First you create a file containing a list of usernames and passwords. Secondly you specify what resources are to be protected and which users are allowed to access them. Setting up IP address access requires only server configuration. Server configuration uses an **.htaccess** file located in the folder it is protecting (root folder www). This allows configuration to take immediate effect and does not require a server restart.

Creating a password file

A list of users and passwords needs to be created in a file. For security, this file must be located outside of the document root. The Uniform Server is pre-configured and uses file: UniServer\htpasswd\www**.htpasswd**

The file contains a list of name and password pairs,
each on a single line.
The name and password are separated using a colon.
A password file can consist of a single line.
The default is **root:root**

```
root:root
mike:fred123
john:xyz45
dave_smith:za123
mike smith:good1234
```



Edit or change password

- A) Enter new name or edit existing one.
- B) Enter new password or edit existing one.
- D) Click **change name password** Note: This changes only the first entry in the list.

Add name/password pair to list

- E) Click **Edit** button. Opens file **.htpasswd** in notepad.
- Enter new **name** and **password** with format name:password

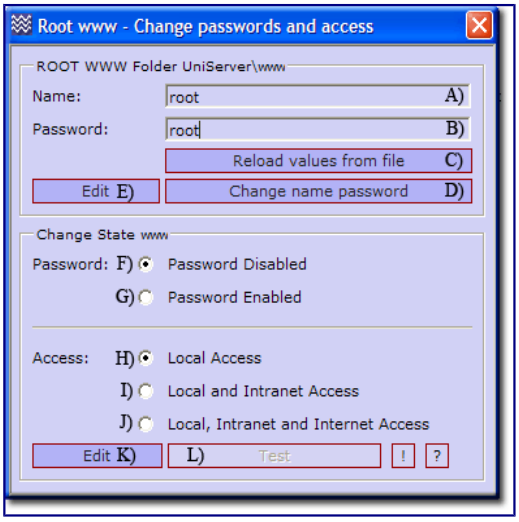
- Save file.
- Optionally click C) which refreshes first name/password displayed.

Note: The menu provides a convenient way to enter names and passwords. You can edit the file UniServer\htpasswd\www\.htpasswd directly using an alternative text editor.

Enable or disable name password access

Server configuration uses an .htaccess file located in root folder UniServer\www. It is pre-configured for basic authentication, default is basic authentication disabled. Four lines (Apache directives) in this configuration file control authentication and are edited as follows:

Enable authentication



- G) Click radio button Password enabled
- Alternatively edit file UniServer\www\.htaccess and uncomment (remove #) four lines as shown:

```
AuthName "Uniform Server - Server Access"
AuthType Basic
AuthUserFile ../../../htpasswd/www/.htpasswd
Require valid-user
```

Disable authentication

- F) Click radio button Password disabled
- Alternatively edit file UniServer\www\.htaccess and comment (add #) to four lines as shown:

```
#AuthName "Uniform Server - Server Access"
#AuthType Basic
#AuthUserFile ../../../htpasswd/www/.htpasswd
#Require valid-user
```

Note: Menu provides a convenient way to enable or disable authentication.

Change Access

Server configuration uses an **.htaccess** file located in root folder UniServer\www. It is pre-configured to allow local access only. Four lines (Apache directives) control who can access to the server. These lines are edited as follows:

Local Access

- **H)** Click radio button **Local Access**
- Alternatively edit file UniServer\www**.htaccess** and uncomment (remove #) four lines as shown:

```
Order Deny,Allow
Deny from all
Allow from 127.0.0.1
Allow from ::1
```

Local and Intranet Access

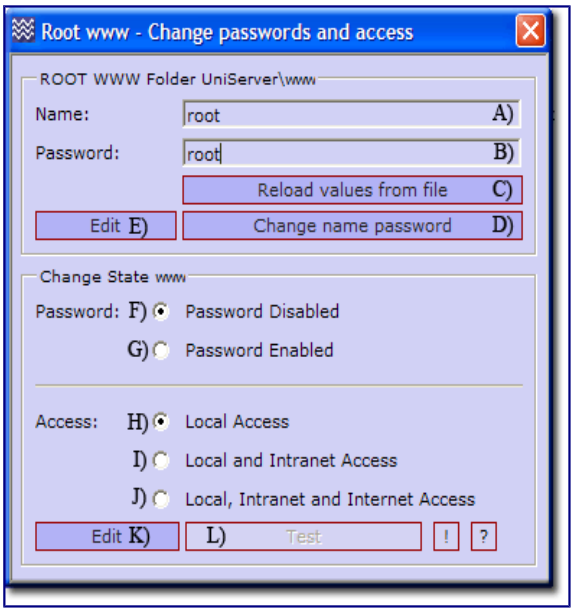
- **I)** Click radio button **Local and Intranet Access**
- Alternatively edit file UniServer\www**.htaccess** and uncomment (remove #) four lines as shown:

```
Order Deny,Allow
Deny from all
Allow from 127.0.0.1 192.168
Allow from ::1
```

Local, Intranet and Internet Access

- J) Click radio button **Local, Intranet and Internet Access**
- Alternatively edit file UniServer\www**.htaccess** and comment (add #) to four lines as shown:

```
#Order Deny,Allow
#Deny from all
#Allow from 127.0.0.1
#Allow from ::1
```



Deny from all. No one allowed access
Allow from 127.0.0.1 192.168 Access is restricted to locahost and Intranet
Allow from 127.0.0.1 Access is restricted to locahost (127.0.0.1) ipv4
Allow from ::1 Access is restricted to locahost (::1) ipv6

Note: The menu provides a convenient way to change and restrict access.

Root folder ssl - Access

Creating a new server certificate for The Uniform Server automatically enables SSL in Apache's configuration file. You can now access your server using either http or https. The Uniform Server has a predefined folder (ssl root folder) that is reserved for https access. By default this folder is restricted to local access only. The following covers opening up your server for public or intranet access, and optionally restricting access using a name and password.

Contents

Overview

UniController: Server Configuration > General > Root ssl - Change passwords and access

This menu is split into three sections as follows:

1) The top section allows you to change the access name and password; the defaults are set to root.

- Generally for a private server a single name and password are used.
- These are displayed and changeable using buttons A) B) D)
- Note: When using a list, only the first entry is displayed and directly changeable.
- Using a list of names and passwords requires that you directly edit the associated name-password configuration file. Use edit button E) to open the file in notepad. The Reload values button C) displays just the first entry in list.

2) Change state: enables or disables the use of a password to access root folder ssl. Default is disabled.

3) Access section: allows you to set who can gain access to root folder ssl. Default is local access only.

Note 1: Server access is controlled using an .htaccess file. This can be directly edited using Edit button K)
Note 2: Apache does not require restarting. All changes take effect immediately.
Note 3: Test button L) opens the index page in the default browser, allowing you to quickly test changes. This index page (root folder ssl) is user configurable with the Change ssl page.

Background information

There are three ways of restricting access to documents: either by IP address, hostname or by asking for a username and password. The first two are very similar and can be used to restrict access on an intranet. The last option allows users to be restricted on an individual basis using a username and password before being allowed access. This process is commonly referred to

as user authentication. The Uniform Server can restrict access using both IP address and user authentication.

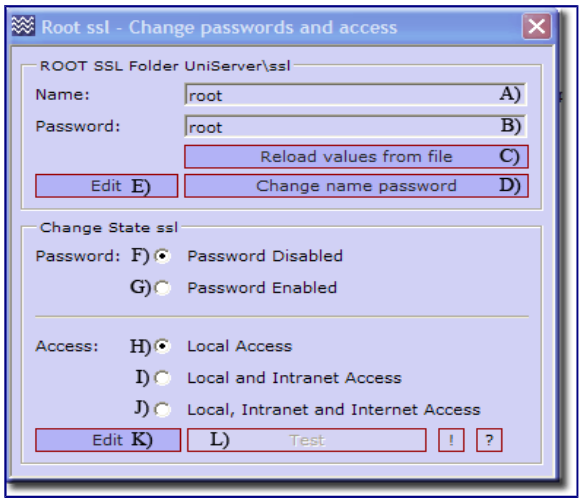
Setting up user authentication requires two steps. First you create a file containing a list of usernames and passwords. Secondly you specify what resources are to be protected and which users are allowed to access them. Setting up IP address access requires only server configuration. Server configuration uses an **.htaccess** file located in the folder it is protecting (root folder ssl). This allows configuration to take immediate effect and does not require a server restart.

Creating a password file

A list of users and passwords needs to be created in a file. For security, this file must be located outside of the document root. The Uniform Server is pre-configured and uses file: UniServer\htpasswd\ssl**.htpasswd**

The file contains a list of name and password pairs,
each on a single line.
The name and password are separated using a colon.
A password file can consist of a single line.
The default is **root:root**

```
root:root
mike:fred123
john:xyz45
dave_smith:za123
mike smith:good1234
```



Edit or change password

- A) Enter new name or edit existing one.
- B) Enter new password or edit existing one.
- D) Click **change name password** Note: This changes only the first entry in the list.

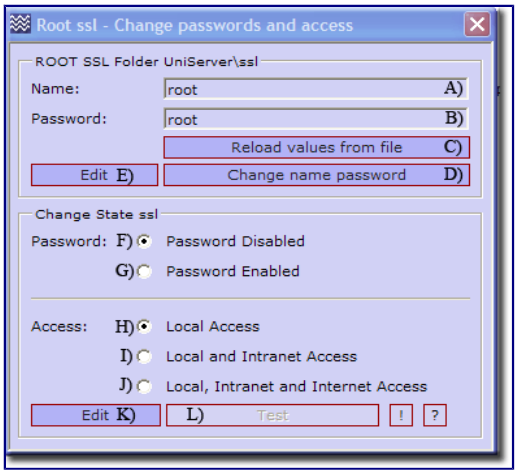
Add name/password pair to list

- E) Click **Edit** button. Opens file **.htpasswd** in notepad.
- Enter new **name** and **password** with format name:password
- Save file.
- Optionally click C) which refreshes first name/password displayed.

Note: The menu provides a convenient way to enter names and passwords. You can edit the file UniServer\htpasswd\ssl\.htpasswd directly using an alternative text editor.

Enable or disable name password access

Server configuration uses an .htaccess file located in root folder UniServer\ssl. It is pre-configured for basic authentication, default is basic authentication disabled. Four lines (Apache directives) in this configuration file control authentication and are edited as follows:



Enable authentication

- G) Click radio button Password enabled
- Alternatively edit file UniServer\ssl\.htaccess and uncomment (remove #) four lines as shown:

```
AuthName "Uniform Server - Secure Server Access"
AuthType Basic
AuthUserFile ../../../htpasswd/ssl/.htpasswd
Require valid-user
```

Disable authentication

- F) Click radio button Password disabled
- Alternatively edit file UniServer\ssl\.htaccess and comment (add #) to four lines as shown:

```
#AuthName "Uniform Server - Secure Server Access"
#AuthType Basic
#AuthUserFile ../../../htpasswd/ssl/.htpasswd
#Require valid-user
```

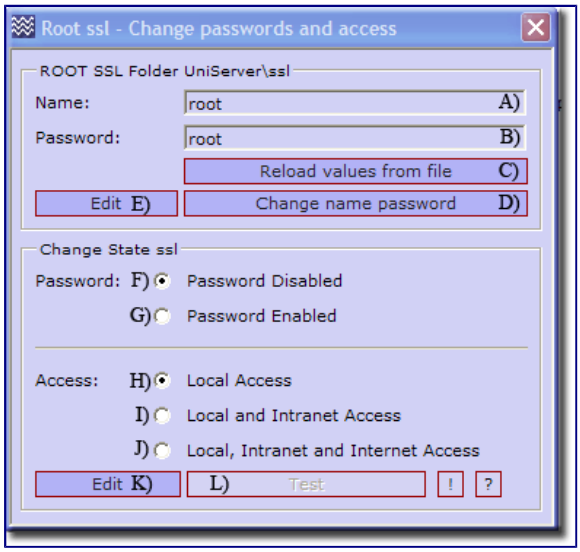
Note: Menu provides a convenient way to enable or disable authentication.

Change Access

Server configuration uses an **.htaccess** file located in root folder UniServer\ssl. It is pre-configured to allow local access only. Four lines (Apache directives) control who can access to the server. These lines are edited as follows:

Local Access

- **H)** Click radio button **Local Access**
- Alternatively edit file UniServer\ssl**.htaccess** and uncomment (remove #) four lines as shown:




```
Order Deny,Allow
Deny from all
Allow from 127.0.0.1
Allow from ::1
```

Local and Intranet Access

- **I)** Click radio button **Local and Intranet Access**
- Alternatively edit file UniServer\ssl**.htaccess** and uncomment (remove #) four lines as shown:

```
Order Deny,Allow
Deny from all
Allow from 127.0.0.1 192.168
Allow from ::1
```

Deny from all. No one allowed access
Allow from 127.0.0.1 192.168 Access is restricted to locahost and Intranet
Allow from 127.0.0.1 Access is restricted to locahost (127.0.0.1) ipv4
Allow from ::1 Access is restricted to locahost (::1) ipv6

Local, Intranet and Internet Access

- J) Click radio button **Local, Intranet and Internet Access**
- Alternatively edit file UniServer\ssl**.htaccess** and comment (add #) to four lines as shown:

```
#Order Deny,Allow
#Deny from all
#Allow from 127.0.0.1
#Allow from ::1
```

Note: The menu provides a convenient way to change and restrict access.

General - phpMyAdmin Access

phpMyAdmin is integrated into The Uniform Server's structure. It is used to administer the MySQL server locally using a browser, or (optionally) over the Internet. It performs various tasks such as creating, modifying or deleting databases, tables, fields or rows; executing SQL statements; or managing database users and permissions. To run phpMyAdmin, simply click the phpMyAdmin button.

Because of the powerful capabilities, phpMyAdmin must be properly secured before allowing external access. The Uniform Server attempts to do this as transparently as possible. By default, access is restricted to localhost only, while Intranet and Internet access is provided by menu options explained below.

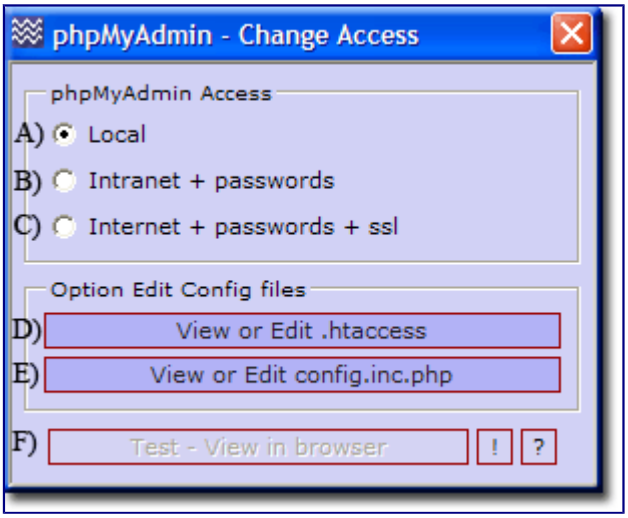
Contents

Features

- Local access - Requires no name/password. These are provided transparently (as user root)
- Intranet + password - Requires name and password as defined for restricted MySQL user (all users)
- Internet + password + ssl - Requires name and password as defined for restricted MySQL user (all users). All transactions performed over a secure encrypted connection
- Disable Internet selection if server certificate is not generated.
- Automatically tracks SSL port used. User may change main server configuration.
- Automatically tracks server name. User may change main server configuration.

Root phpMyAdmin - Change access type

UniController: Server Configuration > General > Root phpMyAdmin - Change access type



- A) **Local** access is the default and prevents both Intranet and Internet users access to phpMyAdmin. Only localhost may connect. It allows the root MySQL administrator direct access without the need to enter a name or password. For security reasons you must change the default MySQL password. This prevents accidental cross-site scripting.
- B) **Intranet + passwords**. Anyone on your Intranet is allowed access to phpMyAdmin, however to gain access to the MySQL server requires a name and password. A user name and password are stored on the MySQL server for each restricted user created. A restricted user is one that is assigned access only to specific databases, and also has limited privileges for them.
- C) **Internet + passwords +ssl** This is similar to Intranet, except that all transactions are performed over a secure encrypted connection using ssl. If you have not installed or created a server certificate, clicking this radio button will produce a warning and the access selection will not change. This security feature forces you to use ssl and prevents sending names and passwords

over the Internet in plain text.

- D) phpMyAdmin root folder is protected by an **.htaccess** file. This menu button opens this file in the default text editor allowing you to view or edit the file.
- E) phpMyAdmin is configured using the user configuration file **config.inc.php**. This menu button opens this file in the default text editor allowing you to view or edit the file.
- F) This menu button opens phpMyAdmin in the default browser. A greyed out button indicates that Apache is not running and its function are unavailable.

Background information

Two files govern phpMyAdmin access and ultimately MySQL server access:
UniServer\home\phpMyAdmin**.htaccess** - Apache configuration
UniServer\home\phpMyAdmin**config.inc.php** - phpMyAdmin user configuration

You can modify the above files to suite your own requirements, but be sure you read the following:

File .htaccess

When selecting a particular access type, certain directives are overwritten as follows:

Local access

- Access is restricted to locahost (127.0.0.1) IPv4
- Access is restricted to locahost (::1) IPv6
- Force SSL disabled (# disables lines)

```
Order Deny,Allow
Deny from all
Allow from 127.0.0.1
Allow from ::1

#RewriteCond %{SERVER_PORT} !=443
#RewriteRule ^ https://localhost:443%{REQUEST_URI} [NS,R,L]
```

Intranet + passwords

- Access is restricted to locahost (127.0.0.1) and Intranet (192.168) IPv4
- Access is restricted to locahost (::1) IPv6

```
Order Deny,Allow
Deny from all
Allow from 127.0.0.1 192.168
Allow from ::1

#RewriteCond %{SERVER_PORT} !=443
#RewriteRule ^ https://localhost:443%{REQUEST_URI} [NS,R,L]
```

- Force SSL disabled (# disables lines)

Intranet + passwords +ssl

- Access restriction removed (# disables lines) allows everyone access
- Force SSL enabled (removed # enables lines)

```
#Order Deny,Allow
#Deny from all
#Allow from 127.0.0.1
#Allow from ::1

RewriteCond %{SERVER_PORT} !=443
RewriteRule ^ https://localhost:443%{REQUEST_URI} [NS,R,L]
```

File config.inc.php

When selecting a particular access type, certain directives are overwritten as follows:

Local access

Automatically use root user password, allowing transparent access.

```
/* Authentication section */
$cfg['Servers'][$i]['auth_type']      = 'config';  // Authentication method (config, http or cookie based)?
$cfg['Servers'][$i]['user']           = 'root';    // MySQL user
$cfg['Servers'][$i]['password']     = $password; // MySQL password (only needed with 'config' auth_type)
$cfg['Servers'][$i]['AllowNoPassword'] = false;     // Must use password
```

Intranet + passwords and **Internet + passwords +ssl**
Force all users to enter a name and password.

```
/* Authentication section */
$cfg['Servers'][$i]['auth_type']      = 'cookie';  // Authentication method (config, http or cookie based)?
$cfg['Servers'][$i]['user']           = '';        // MySQL user
$cfg['Servers'][$i]['password']       = '';        // MySQL password (only needed with 'config' auth_type)
$cfg['Servers'][$i]['AllowNoPassword'] = false;     // Must use password
```

Apache - Introduction

Apache HTTP Server is an open-source web server platform. It is one of the main components of The Uniform Server. Apache has been pre-configured to run as a local web server. There are several configuration options allowing the server to be accessed from an Intranet or the Internet. The Uniform Server provides an easy-to-use control interface, UniController, allowing you to run servers as either a standard program or a service. UniController contains a sub-menu specifically targeting Apache. From this sub-menu you can put your server on-line, change ports or create a server certificate. This page covers all the sub-menu options.

Contents

Edit Basic Configuration

This form allows you to change commonly configured Apache options.

UniController: Server Configuration > Apache > Edit Basic Configuration

A default installation of The Uniform Server configures the main server for localhost.
This is intended for testing. Before putting your servers on-line, if you have registered a domain name, change "Server name" and "Server Admin Email" to match that name.

- 1) **Server Name** sets the hostname of your server; for example, fred.com
- 2) **Admin E-mail** Use a real e-mail address for users to contact; for example, admin@fred.com
- 3) **Directory Index Files** This list is searched when a client requests an index page without explicitly specifying it. If a match is found that index page is served.
- 4) Apache processes files with the extensions listed; any **SSI Server Side Includes** directives found in these files are executed.
- 5) **Server Signature** This allows Apache to add a footer line for server-generated documents (Value is On or Off).
- 6) **Listen Port** This tells Apache to listen to a specific port and to accept incoming requests only on that specified port.
- 7) **Update Configuration** button applies the changes.

Note 1: For the new settings to become effective you must restart the server.
Note 2: For more information, see Apache Basic Configuration

Note: For the new settings to become effective, you must **restart Apache**.

Edit configuration files

Edit httpd.conf

UniController: Server Configuration > Apache > Edit httpd.conf

Opens the configuration file in notepad, allowing you to directly edit the file.

File: UniServer\usr\local\apache2\conf**httpd.conf**

Edit ssl.conf

UniController: Server Configuration > Apache > ssl.conf

Opens the configuration file in notepad, allowing you to directly edit the file.

File: UniServer\usr\local\apache2\conf\extra**ssl.conf**

View log files

The following menu options provide convenient short cuts allowing you to view server log files. If you are experiencing problems with the server, these should be your first port of call; they may shed light on a particular issue.

Below are the paths to each log file along with a small extract:

View Error log file error.log

Location:UniServer\usr\local\apache2\logs**error.log**

```
[Fri Jul 08 21:25:46 2011] [notice] Apache/2.2.19 (Win32) mod_ssl/2.2.19....
[Fri Jul 08 21:25:46 2011] [notice] Server built: May 20 2011 21:39:40
[Fri Jul 08 21:25:46 2011] [notice] Parent: Created child process 1028
[Fri Jul 08 21:25:50 2011] [notice] Child 1028: Child process is running
[Fri Jul 08 21:25:50 2011] [notice] Child 1028: Acquired the start mutex.
[Fri Jul 08 21:25:50 2011] [notice] Child 1028: Starting 250 worker threads.
[Fri Jul 08 21:25:50 2011] [notice] Child 1028: Starting thread to listen on port 443.
[Fri Jul 08 21:25:50 2011] [notice] Child 1028: Starting thread to listen on port 80.
```

View Access log file access.log

Location:UniServer\usr\local\apache2\logs**access.log**

```
127.0.0.1 - - [08/Jul/2011:21:25:53 +0100] "GET /us_splash/index.php HTTP/1.1" 200 1290
"-" "Mozilla/5.0 (Windows; U; Windows NT 5.1; en-GB; rv:1.9.2.18) Gecko/20110614
Firefox/3.6.18 ( .NET CLR 3.5.30729)"
```

View SSL Error log file error.log

Location:UniServer\usr\local\apache2\logs**error_ssl.log**

```
[Fri Jul 08 21:29:44 2011] [warn] RSA server certificate CommonName (CN)
 `localhost123' does NOT match server name!?
[Fri Jul 08 21:29:45 2011] [warn] RSA server certificate CommonName (CN)
 `localhost123' does NOT match server name!?
```

View SSL Access log file access.log

Location:UniServer\usr\local\apache2\logs**access_ssl.log**

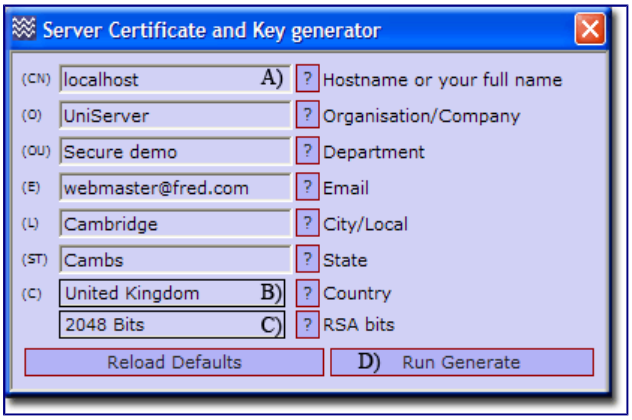
```
127.0.0.1 - - [08/Jul/2011:21:28:47 +0100] "GET /index.php HTTP/1.1" 200 762
127.0.0.1 - - [08/Jul/2011:21:28:47 +0100] "GET /us_splash/css/style.css HTTP/1.1" 304 -
```

Generate Certificate

This form allows you to quickly generate a self-signed server certificate or a certificate-signing request (CSR).

UniController: Server Configuration > Apache > Generate Certificate

- This opens the **Server Certificate and Key generator** menu shown at right
- D) Click **Run Generate** After a short time, a confirmation pop-up is displayed.
- For the new configuration to become effective, you must **restart Apache server**.



The above generates a self-signed certificate.

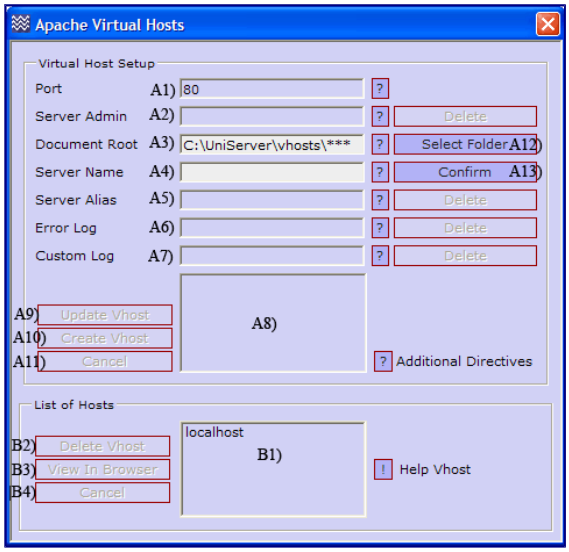
Note 1: For detailed information about enabling SSL, see Apache - SSL
Note 2: For detailed information about self-signed certificates, see Server Certificate Self-Signed
Note 3: For detailed information about Certificate Signing Requests, see Free Server Certificate

Apache Vhosts

This menu allows you to quickly create virtual host blocks.

UniController: Server Configuration > Apache > Apache Vhosts

Creating a Vhost requires only two form entries, a server root folder and a host name. After entering these, click the confirm button. All other directives are optional, and are automatically filled in with values based on the above two form entries. Clicking the Create Vhost button creates the root folder if it does not exist. A Vhost is created in the Vhost configuration file and an entry is written to your PC's host file. Here is a summary of the steps:



- **A3)** Either replace ******* with the folder name you wish to use as your server root folder or click **Select Folder** button and navigate to an existing folder and select it.
- **A4)** Enter your server name (host name). This is the domain name you registered with a registrar, for example **fred.com**. It is the name you enter in a browser to access your web site (excluding the **http://** part).
- **A13)** Click the **Confirm** button. This enables the greyed out buttons and fills in form fields with selected values for other Apache directives. Change or delete these as appropriate.
- **A10)** Click **Create Vhost** button to create your virtual host.

Note 1: After creating a Vhost you **must restart** your Apache server otherwise Apache will not recognise the new configuration.

Note 2: For detailed information see Apache Vhosts

Apache Data

These menu options provide convenient short cuts that allow you to view the server status and other important information. They are shown along with a small extract of each file. Each menu option runs a small script that redirects your browser to the appropriate page. You also can type this address directly into your browser.

Apache server-status

Menu item redirects to:
http://localhost/**server-status**

Apache Server Status for localhost

Server Version: Apache/2.2.19 (Win32) mod_ssl/2.2.19 OpenSSL/1.0.0d PHP/5.3.6 DAV/2
Server Built: May 20 2011 21:39:40

Current Time: Friday, 08-Jul-2011 22:43:16 GMT Daylight Time
Restart Time: Friday, 08-Jul-2011 22:29:47 GMT Daylight Time
Parent Server Generation: 0
Server uptime: 13 minutes 28 seconds
Total accesses: 13 - Total Traffic: 2 kB
.0161 requests/sec - 2 B/second - 157 B/request
1 requests currently being processed, 249 idle workers

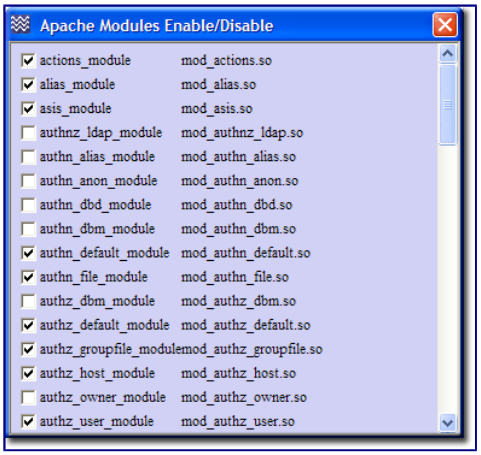
Apache server-info

Menu item redirects to:
http://localhost/**server-info**

Server Settings
Server Version: Apache/2.2.19 (Win32) mod_ssl/2.2.19 OpenSSL/1.0.0d PHP/5.3.6 DAV/2
Server Built: May 20 2011 21:39:40
Server loaded APR Version: 1.4.5
Compiled with APR Version: 1.4.5
Server loaded APU Version: 1.3.12
Compiled with APU Version: 1.3.12
Module Magic Number: 20051115:28
Hostname/port: localhost:80
Timeouts: connection: 300 keep-alive: 5

Apache Modules Enable Disable

This sub-menu allows you to enable or disable Apache Modules.
NOTE: improper use can totally disable your server!
Be sure you know what a module does before you enable or disable it.



Server Configuration > Apache > Apache Modules Enable/Disable

- If checkbox is unchecked: Module disabled
- If checkbox is checked: Module enabled

Note: After changing state you must **restart** the Apache server for the new configuration to be recognised.

This modifies the file
UniServer\usr\local\apache2\conf**httpd.conf**

Apache Basic Configuration

This sub-menu allows you to change commonly configured Apache options.

Contents

Overview

The Uniform Server's default installation configures the main server with the values shown at right.

UniController: Server Configuration > Apache > Edit Basic Configuration

- 1) **Server Name** sets the hostname of your server; for example, fred.com
- 2) **Admin E-mail** Use a real e-mail address for users to contact; for example, admin@fred.com
- 3) **Directory Index Files** This list is searched when a client requests an index page without explicitly specifying it. If a match is found that index page is served.
- 4) Apache processes files with the extensions listed; any **SSI Server Side Includes** directives found in these files are executed.
- 5) **Server Signature** This allows Apache to add a footer line for server-generated documents (Value is On or Off).
- 6) **Listen Port** This tells Apache to listen to a specific port and to accept incoming requests only on that specified port.
- 7) **Update Configuration** button applies the changes.

Note: For the new settings to become effective, you must **restart Apache**.

Server Name

Server Name sets the hostname of your server; this is used when creating redirection URLs.

If you put your server online, change this to your web address. For example, if you registered the domain www.fred.com, then use that.

Note: To test your web site without being connected to the Internet and to prevent your browser complaining that the "Server cannot be found", you need an entry in your PC's hosts file. For the above example, put in 127.0.0.1 www.fred.com

- Navigate to folder: **C:\WINDOWS\system32\drivers\etc**
- Open the file **hosts**
- At the end of this file, add: **127.0.0.1 www.fred.com** (use your own domain)
- Save this file. Now when you type **http://www.fred.com** (your domain) into your browser address bar you will be able to view your web site.

If you are hosting more than one domain using name-based virtual hosting, you can leave the default setting, but you must use the registered web address in the virtual hosts.

Server Admin Email

The Server Admin Email address is included with some automatically generated error pages sent to a client (user). Use a real e-mail address for users to contact you in case problems are encountered.

Directory Index Files

When a client requests an index page without explicitly specifying it (for example **www.fred.com**) the list of **index pages** specified in **Direct Index Files** is searched in the order left to right. If a match is found, that index page is served. The first one that matches is served and any others that may be in the same folder are ignored.

Default list: index.html index.shtml index.html.var index.htm index.php3 index.php index.pl index.cgi

The Uniform Server's default installation uses index.php, which is found in the root folder **www**. If you insert your own **index.html** page, this will be picked up first, overriding **index.php**

Server Side Includes

SSI is a server side scripting language allowing you to include the contents of a file into another using SSI commands.

To insert these commands, a special comment tag is used; for example:

```
<!--#include file="external.htm"-->
<!--#include virtual="/external.htm"-->
```

The first command assumes that the file is located in the same directory as the document containing it, while the second syntax uses an absolute reference to "external.htm", starting from your root HTML directory.

You can insert these into an html file, but in order to process them, Apache needs to know which files types to use. The standard file extensions are:

- .shtml .shtm .sht

You can specify any file extension you like, even .htm and .html, but there is a small processing hit for each file accessed.

Server Signature

This allows Apache to add a footer line for server-generated documents. It contains the server version number, server name, and creates a "mailto:" reference to ServerAdmin.

For security reasons, this information is normally not displayed except for testing and the default is set to **off**.

Listen Port

This tells Apache to listen to a specific port and to accept incoming requests only on that port. Web page servers and all web browsers default to this standard port (80), which is reserverd for World Wide Web traffic (WWW). When a user types an address into a browser, the port is automatically assumed unless overridden, as explained later.

Port 80 already in use
If another program is using port 80, you will want to change the listening port to avoid conflicts, or your Apache server will not start. Before changing the Apache port, be aware that this makes the server non-standard. A preferable solution is to change the port of the conflicting program.

Apache already running
You can change the port while the server is running, but you must restart the servers after making any changes. The new settings will then be picked up by Apache.

General Note
After changing Apache's listing port, in order to display a file you must include this port number in the host name. For example suppose you have changed the port to 8080.

To view your home page you enter the following address **http://localhost:8080/** into your browser address bar. Likewise to view any other page in your root folder www you would enter **http://localhost:8080/mypage.html** to display that page.

Note: When the server starts, the correct port number is automatically added to the address and redirection takes place to the splash page. Using the above port, you will see **http://localhost:8080/us_splash/index.php** displayed in the address bar.

Apache Vhosts

The Uniform Server is configured for name-based virtual hosting. This allows you to run any number of web sites from a single IP address. Each domain-name you host must first be registered and an “A” record created on a DNS server. This “A” record translates the domain-name to your IP address.

Note: For testing, “A” records are simulated using a PC's hosts files.

Contents

Features

The main features of The Uniform Server's Vhost configuration menu are:

- Include separate log files and server alias
- Auto enable when first vhost created
- Auto disable when last vhost deleted
- Port tracking follows main server port (defaults to port 80)
- Writes host name to Windows hosts file when vhost created
- Deletes host name from Windows hosts file when vhost deleted

Overview

Vhosts are configured using a separate configuration file UniServer\usr\local\apache2\conf\extra**httpd-vhosts.conf**.

This file requires enabling in Apache's main configuration file UniServer\usr\local\apache2\conf**httpd.conf**

Search this file for the following line:< br />	Remove the hash "#" to enable as shown below:< br />
#Include conf/extra/httpd-vhosts.conf	Include conf/extra/httpd-vhosts.conf

Note: The above is automatically performed when using The Uniform Server's Vhost configuration menu.

Inform Apache to use Vhosts

Apache requires the directive **NameVirtualHost address:port** which instructs it to use virtual hosts and resolve an address on the specified port. The Uniform Server uses ***:80**, where the * is a wildcard meaning use all addresses. Port 80 is the standard web server listening port. This can be changed if required, for example, to port 8080.

A separate **NameVirtualHost** directive is required for each different port used. These NameVirtualHost directives must have a corresponding listening directive. The Uniform Server's Vhost configuration file contains the following block:

```
#--US_START----------Do not remove
NameVirtualHost *:80
#--US_END----------Do not remove
```

The "Do not remove" Lines are used by The Uniform Server to locate the block and insert lines as appropriate.
The NameVirtualHost directive has a matching Listen directive Listen 80 located in the main configuration file so it is not required in this configuration block.

However, adding a virtual host that uses port 8080, a listen directive is required. The configuration block becomes:

```
#--US_START--------------Do not remove
Listen 8080
NameVirtualHost *:80
NameVirtualHost *:8080
#--US_END----------------Do not remove
```

The main configuration file (httpd.conf) does not contain a Listen 8080 directive and it is required and added to this configuration block along with the NameVirtualHost directive.

Note: These are automatically added when using The Uniform Server's Vhost configuration menu.

Adding a Vhost

Each virtual host requires a separate configuration block. These blocks inherit their configuration from the main configuration file httpd.conf. Specifying an Apache directive in a Vhost block overrides that of the main configuration file. The advantage of this inheritance is to reduce number of Apache directives required for each Vhost. The absolute minimum required is to specify a **DocumentRoot** and **ServerName**.

Apache resolves a Vhost by matching its port and ServerName. On finding a match, the server's content is fetched from the folder specified by DocumentRoot. If it cannot find a match, it uses the first Vhost defined (the default). Note: to avoid conflicts, Apache uses a _**default**_ directive for the address of this first virtual host.

Example: to add a virtual host for fred.com on port 8080 and to serve content from folder UniServer\vhosts\fred, the minimum configuration required is:

```
#--US_START--------------Do not remove
Listen 8080
NameVirtualHost *:80
NameVirtualHost *:8080
#--US_END----------------Do not remove

<VirtualHost _default_:80>
 DocumentRoot C:/Coral/UniServer/www
 ServerName localhost
</VirtualHost>

<VirtualHost *:8080>
 DocumentRoot C:/Coral/UniServer/vhosts/fred
 ServerName fred.com
</VirtualHost>
```

The default Vhost maps to The Uniform Server's folder www. If you wish, this can be changed as well as the server name.

The new Vhost fred.com maps to folder UniServer\vhosts\fred. This can be any folder you wish. All vhost root folders are created in the folder **vhosts**. This is not a strict requirment, but it is provided for portability (as when using a USB memory stick).

Note: Port 8080 is shown as an example. Generally all Vhosts would use the standard port 80.

Vhost additional Apache directives

Other Apache directives can be included in a Vhost block. Common directives are shown in this example block:

ServerAlias Allows you to access the server using sub-domains. Each sub-domain must have a corresponding DNS entry.

```
<VirtualHost *:80>
  ServerAdmin webmaster@fred.net
  DocumentRoot C:/Coral/UniServer/vhosts/fred_net
  ServerName fred.net
  ServerAlias www.fred.net  *.fred.net
  ErrorLog logs/fred.net-error.log
  CustomLog logs/fred.net-access.log common
</VirtualHost>
```

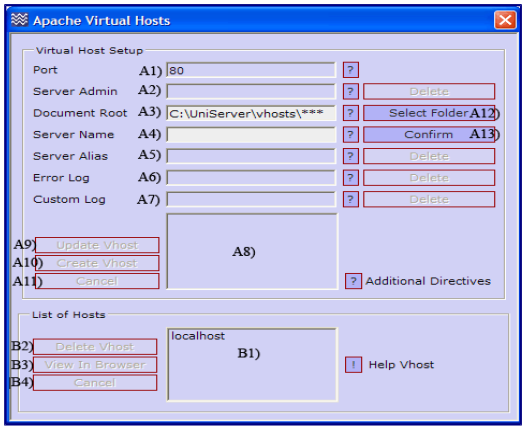
Each Vhost can have a separate **ErrorLog** and **CustomLog** log file. If you do not specify these log files, the main server log files are used.

Note: Separate log files will eat into your file allocation resources, depending on the total number of Vhosts you wish to host. This may become an issue .

Apache Vhosts

This menu allows you to quickly create virtual host blocks.

UniController: Server Configuration > Apache > Apache Vhosts



Creating a Vhost requires only two form entries, a server root folder and a host name. After entering these, click the confirm button. All other directives are optional, and are automatically filled in with values based on the above two form entries. Clicking the Create Vhost button creates the root folder if it does not exist. A Vhost is created in the Vhost configuration file and an entry is written to your PC's host file. Here is a summary of the steps:

- **A3)** Either replace ******* with the folder name you wish to use as your server root folder or click **Select Folder** button and navigate to an existing folder and select it.
- **A4)** Enter your server name (host name). This is the domain name you registered with a registrar, for example **fred.com**. It is the name you enter in a browser to access your web site (excluding

the **http://** part).

- **A13)** Click the **Confirm** button. This enables the greyed out buttons and fills in form fields with selected values for other Apache directives. Change or delete these as appropriate.
- **A10)** Click **Create Vhost** button to create your virtual host.

Note 1: After creating a Vhost you **must restart** your Apache server otherwise Apache will not recognise the new configuration.

Note 2: For detailed information see **Apache Vhosts**

Vhost Configuration Menu Example

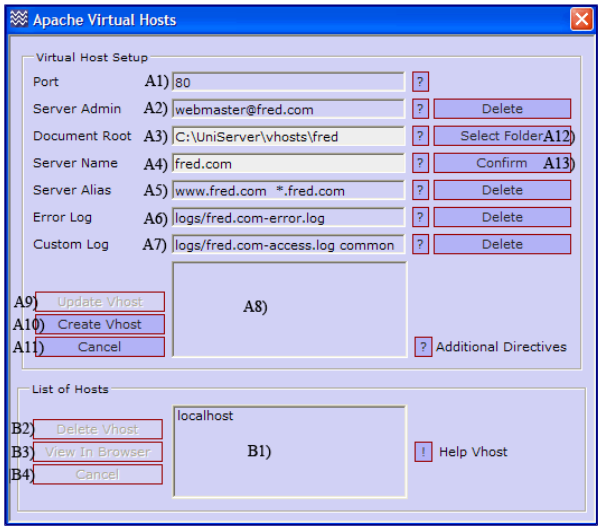
For clarity following example shows Vhost form filled in and the corresponding Vhost block created in the configuration file for host name fred.com and server root folder **fred**.

The block created in httpd-vhosts.conf

```
<VirtualHost *:80>
  ServerAdmin webmaster@fred.com
  DocumentRoot C:/UniServer/vhosts/fred
  ServerName fred.com
  ServerAlias www.fred.com  *.fred.com
  ErrorLog logs/fred.com-error.log
  CustomLog logs/fred.com-access.log common
</VirtualHost>
```

The entry created in host file

```
127.0.0.1 fred.com
```



Note: An entry will be created in the List of Hosts (B1). After restarting Apache and returning to this menu you can click an entry. Clicking an entry enables the greyed out buttons, allowing you to perform the following tasks:

- **A9)** The Virtual host setup form is filled in allowing you to edit the Vhost. Click the **Update Vhost** button to apply your modification.
- **B2)** To remove a Vhost, click the **Delete Vhost** button. It deletes the Vhost block from the configuration file and removes the entry in

the Hosts file. The root folder and all its content are deleted, however you are requested to confirm this action before it is implemented.

- **B3) View In Browser** opens default browser and displays the Vhost selected.

Apache - SSL

Secure Sockets Layer (SSL) offers privacy for client-server communication. SSL establishes an encrypted tunnel using cryptography algorithms and keys through which other protocols such as HTTP are transported.

By default, The Uniform Server installation has SSL disabled, for the reason of security. A certificate/key pair are required and must be unique to the particular server. After creating a new server certificate/key pair, SSL will be automatically enabled in Apache's configuration file.

Contents

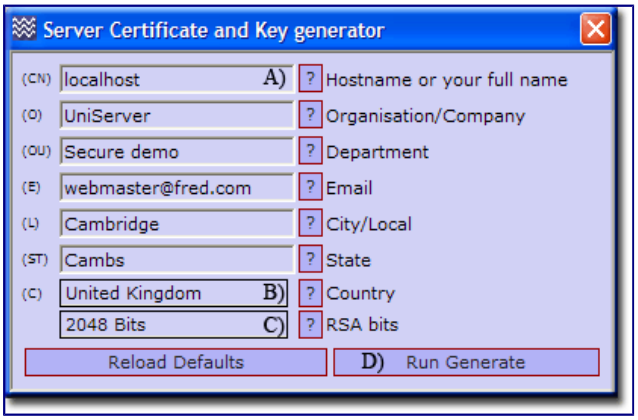
How to Enable SSL

After generating a self-signed certificate, SSL is automatically enabled. The "Server Certificate and Key generator" form has been pre-configured for a self-signed certificate and there is no need to change these values. Just click "Run Generate".

Please note, however, that a self-signed certificate is not considered secure. Your browser will most likely complain about it. Nevertheless, it is fine for local testing, and you can set an override for most browsers. For the case of a production server environment, **DO NOT USE** a self-signed certificate, since it will not be accepted by your users.

UniController: Server Configuration > Apache > Generate Certificate

- This opens **Server Certificate and Key generator** menu shown on right
- D) Click Run Generate. After a short time a confirmation pop-up is displayed.
- For the new configuration to become effective, **restart Apache server**.



Note 1: A) If you have changed the server name using Apache configuration menu, that name will be displayed instead of localhost.

Note 2: B) C) Are dropdown menus.

Note 3: C) 2048 Bits provide high-grade encryption; no need to change this.

After generating a self-signed certificate the following configuration changes are made:

Apache configuration file changes

Both php_openssl.dll and ssl.conf are enabled as follows:

- Apache configuration file: UniServer\usr\local\apache2\conf\httpd.conf
- Existing line: #LoadModule ssl_module modules/mod_ssl.so
- Changed to: LoadModule ssl_module modules/mod_ssl.so

The above change in turn enables ssl.conf via this block

```
<IfModule mod_ssl.c>
  Include conf/extra/ssl.conf
</IfModule>
```

Background

The following provides an overview of SSL and background information on how SSL is implemented on The Uniform Server. This section can be skipped.

SSL Overview

The following outlines the SSL process with respect to a client. A Client is generally a user's browser. Let's assume your web-site server has the registered domain fred.com

- A client makes a connection to fred.com on the SSL port (standard port is 443) by typing https://fred.com into their browser. Note the use of https instead of http. On connecting to web server, the client provides a list of available ciphers it can use.
- The Server picks the strongest cipher that both understand and support. The Server sends back a certificate with its name and public encryption key, signed by a trusted Certificate Authority.
- The Client checks the certificate with the CA. Browsers have a collection of CAs stored locally. These are checked first, avoiding the need to directely contact the CA, and thus speeding up the process.
- If the certificate is approved, the Client sends back a random number encrypted with the server's public key. This Number is unique to the client and can only be decrypted by the server using its private key.
- The Server and the Client use this random number to generate encrypted packets. Both Client browser and Server now communicate using encryption and all transactions are secured. The browser displays the secure icon.

IP addresses and SSL

An SSL certificate is bound to your fully qualified domain name, which is encrypted into the certificate. Modern browsers send the server name identification (SNI) along with a request. Apache can use this in Vhosts to resolve certificates.

Unfortunately IE remains in the dark ages and expects servers to resolve using IP addresses. If you attempt to have more than one SSL certificate associated with the same IP address you will get undesired results. The bottom line: to appease IE, you are restricted to using a single Apache SSL Vhosts name.

SSL Virtual Host

Generating a self-signed certificate enables the SSL Virtual Host configuration file. You can now access your server using either http or htpps; when using https all transactions are encrypted.

Using https incurs a small speed penalty so generally a web-site is accessed using http and not https. If a user comes in on http and that linked resource requires https, it is forced (switched) to https. You can define a folder (ssl root folder) to specifically use only https.

The Uniform Server is pre-configured to run both a secure server (on port 443) and a regular

server (on port 80). These are separated using VirtualHosts, which has the advantage of maintainability.

To highlight this separation, a default Server installation has a user configuration button View ssl pre-assigned to it. By default this button is greyed out and enabled only after a server certificate is generated. It then allows the secure folder's index page to be viewed in a browser. Note that you can re-assign this user button.

Default Virtual Host - Configuration

Configuration file: UniServer\usr\local\apache2\conf\extra**ssl.conf**

```
# File name: ssl.conf
# Created By: The Uniform Server Development Team
# Edited Last By: Mike Gleaves (ric)
# Main Apache HTTP server configuration file.
# V 1.0 27-6-2011
#======================================================

##################### Global SSL ##########################
Listen 443
#== Some MIME-types for downloading Certificates and CRLs
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl

#== Pass Phrase Dialog:(`builtin' is a internal terminal dialog)
SSLPassPhraseDialog  builtin

#== Inter-Process Session Cache:
SSLSessionCache shmcb:logs/ssl_scache(512000)
SSLSessionCacheTimeout 300

#== SSL engine uses internally for inter-process synchronization.
SSLMutex default

#== Pseudo Random Number Generator (PRNG):
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin

############ SSL Virtual Host ############################

NameVirtualHost *:443
<VirtualHost _default_:443>

ServerName localhost
DocumentRoot C:/UniServer/ssl
ServerAdmin you@example.com

ErrorLog logs/error_ssl.log
TransferLog logs/access_ssl.log

#== SSL Engine Switch:
SSLEngine on
SSLOptions +StrictRequire
```

```
#== SSL Cipher Suite:
SSLProtocol -all +TLSv1 +SSLv3
SSLCipherSuite HIGH:MEDIUM:!aNULL:+SHA1:+MD5:+HIGH:+MEDIUM

#== Server Certificate:
SSLCertificateFile C:/UniServer/usr/local/apache2/server_certs/server.crt

#== Server Private Key:
SSLCertificateKeyFile C:/UniServer/usr/local/apache2/server_certs/server.key

#== StartSSL certificate chain for class 1 certificates
# Disable when using a self-signed certificate
# Enable remove # disable add #

#SSLCertificateChainFile C:/UniServer/usr/local/apache2/server_certs/sub.class1.server.ca.pem
#SSLCACertificateFile    C:/UniServer/usr/local/apache2/server_certs/ca.pem

SSLVerifyClient none
SSLProxyEngine off

#== Server Root folder:
<Directory "C:/UniServer/ssl">
  AllowOverride All
  Order allow,deny
  Allow from all
  SSLRequireSSL
</Directory>


#== Most problems of broken clients are related to the HTTP
# keep-alive facility. Disable keep-alive for those clients.
SetEnvIf User-Agent ".*MSIE.*" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0

</VirtualHost>
```

Certificates and signing request (CSR) location

The Uniform Server uses OpenSSL to generate:

- A self-signed certificate and server key.
- Or a server key and certificate signing request.

OpenSSL and supporting files are located in folder UniServer**openssl**.
Note: To view installed server certificate details, run UniServer\openssl**View_cert_details.bat**

Server certificates are located or copied to folder UniServer\usr\local\apache2**server_certs** A default installation pre-installs the required intermediate certificate for StartSSL **ca.pem** and sub.class1.server.**ca.pem**

Apache - Server Certificate Self-Signed

This covers how to enable SSL on The Uniform Server 8.0-Coral.

The Uniform Server does not include a test server certificate/key pair, so a default installation has SSL disabled. The reason is one of security. A certificate/key pair must be unique to each server. After creating a new server certificate/key pair, SSL is automatically enabled in Apache's configuration file.

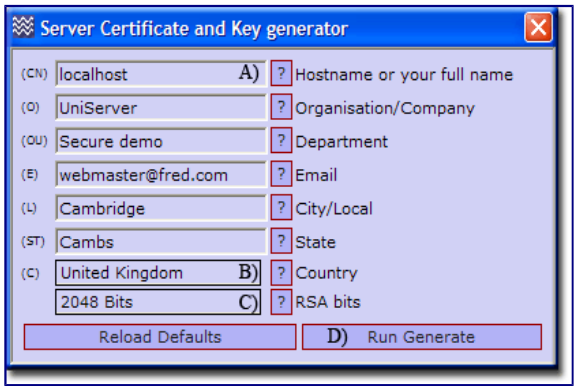
Contents

Creating a self-signed certificate

Generating a self-signed certificate requires only a few mouse clicks. The “Server Certificate and Key generator” form has been pre-configured for a self-signed certificate there is no need to change these values just click “Run Generate” However if you wish you can change any of the defaults.

UniController: Server Configuration > Apache > Generate Certificate

- This opens **Server Certificate and Key generator** menu shown on right
- D) Click **Run Generate** After a short time a confirmation pop-up is displayed.
- For the new configuration to become effective, **restart Apache server**.



Note 1: A) If you have changed the server name, that will be displayed instead of localhost.

Note 2: B) C) Are dropdown menus.

Note 3: C) 2048 Bits provide high-grade encryption. It's recommended not to change this.

Alternative Scripts

The Uniform Server provides three scripts to generate a server certificate and key pair.

1. UniServer\openssl\Generate_server_cert_and_key.bat
2. UniServer\openssl\Generate_server_cert_and_key.vbs
3. UniServer\coral_con\scripts\Key_cert_gen.hta

Generate_server_cert_and_key.bat

This script generates a self-signed server certificate and key pair. It assumes you have not changed the server name from its default of localhost. This allows the certificate and key to be automatically generated and installed without any user input.

Note: The certificate signing request is not required and is deleted.

Generate_server_cert_and_key.vbs

This script also generates a self-signed server certificate and key pair, but it assumes you have changed the server name from its default of localhost. A popup displays the current server name setting. You can either accept this displayed value or change it as required. Pressing OK in

either case will generate and install the certificate and key.

Note: The certificate signing request is not required and is deleted.

Key_cert_gen.hta

***Self-signed certificate*:**
This script is similar to the above in that it generates a self-signed server certificate and key pair. Several certificate defaults are displayed including server-name. For a self-signed certificate you need only change the server-name or accept its default. Pressing "Run Generate" generates and installs the certificate and key.

***Signed certificate*:**
If you are intending to purchase a signed certificate, fill in all appropriate form fields. Pressing “Run Generate” generates and installs the (self-signed) certificate and key. Unlike the previous two key-cert generation scripts, this script does not delete the certificate-signing request (server.csr). It is located in folder UniServer\openssl. You will open this file and post the contents for signing by the certificate authority. When you receive the signed certificate, you will replace the self-signed certificate with it.

Note 1: Copy both the server key, UniServer\usr\local\apache2\server_certs\ssl.key\server.key, and the returned signed certificate to a writable CD or USB memory stick for safekeeping.

Note 2: For a free signed certificate check out the following page: Free server certificate

Apache - Free Server Certificate

A free server certificate using the service from http://www.startssl.com/ is a better choice than a self-signed certificate. Please note that The Uniform Server developers have no relation or affiliation with this site. The material is included here only to provide a better alternative to a self-signed certificate. Other choices likely are available.

Contents

StartSSL (StartCom) is a public certification authority providing digital certificates for free! They have their root certificate in all major browsers, which means that certificates issued by StartSSL are trusted by default. Major browsers include Firefox, Internet Explorer, Gogole Chrome, Safari and Opera. The term free means just that, no hidden catches or trials, so The Uniform Server support now includes support for StartCom.

Obtaining a free server certificate requires a number of steps. First you need to have a registered domain name, then you create a certificate-signing request and obtain a personal ID certificate by registering with StartCom. With this in place, you can login to StartCom and have your certificate signed. Finally you install this certificate on your server. This page covers each step in more detail.

Register a domain name

Register a domain name with a registrar of your choice. The information you supply pertains to the domain you are registering. Of importance for obtaining a free certificate is the contact e-mail address you supply. I am referring to the address where a registrar can contact you. This e-mail address will be used by StartCom to verify who you are (domain name). Alternatively, if your domain name is active (site has a mail-server) the site email address can be used for verification.

Note: Your registrar ***contact e-mail address*** allows you to obtain a signed certificate before making your site live. This allows you to test your site locally with a signed certificate and confirm that Apache's SSL Vhost server verification chain works.

Create an account at StartSSL

Creating an account at StartSSL can be done in advance. You are asked for certain details that identify you. This information is used for creating a personal identification (ID) certificate that is installed into your browser. One purpose of this certificate is to automatically log you into your StartSSL account.

1. Go to [https://www.startssl.com/?app=12/] website
2. Click Sign up
3. Fill in all fields in the sign up form and click Continue
4. An authentication code will be e-mailed to you.
5. Copy and paste it into the Complete Registration form. Click Continue
6. Choose the level of security required and click Continue
7. Allow the pop up to install security certificate into your browser
8. Click Install
9. If prompted by your firewall, allow outside access. (Certificate authority)
10. Click Finish

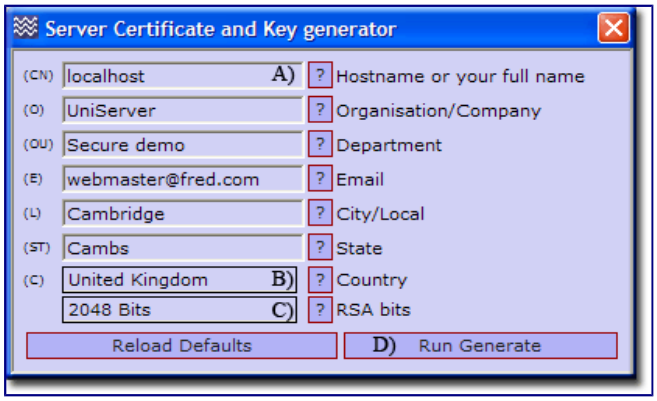
Note: Backup your client certificates!

Creating a certificate signing request (CSR)

Generating a certificate-signing request is similar to generating a self-signed certificate. The only difference is to change Hostname from localhost to your registered domain name. For a free StartCom certificate, this field is the only one that is part of the certificate. If you change any of the other fields, the class of the certificate changes as well as its cost, and will require further validation. These instructions cover only the free certificate.

UniController: Server Configuration > Apache > Generate Certificate

- This opens the **Server Certificate and Key generator** menu shown at right
- A) Change Common Name (CN) to your registered domain name; for example, fred.com
- D) Click **Run Generate**. After a short time, a confirmation pop-up is displayed.



This generates your new server key (server.key) and CSR (server.csr)

It's important to save these files to a safe and secure location (example: UniServer\certs) as follows:

- Save UniServer\openssl**server.csr** to folder UniServer\certs
- Save UniServer\usr\local\apache2\server_certs\server.key to folder UniServer\certs
- Create a new text document named **server.crt** in folder UniServer\certs

Note 1: The empty file server.crt is where you will copy your signed certificate. Folder UniServer\certs is used only to store your original certificates. It keeps them as a separate group.

Note 2: Folder UniServer**certs** is just an example. Ideally you would save its final content to a pen drive, writable CD or another PC for safekeeping.

Sign certificate at StartSSL

Login to StartSSL

- Go to https://www.startssl.com/?app=12/ website
- Click Authenticate
- Click Control Panel

Validation Wizard

1. Click **Validations Wizard** tab (top menu)
2. From drop-down menu choose **Domain Name Validation** . -- Click continue

3. Enter the **URL of your domain name** to be validated
4. Select an e-mail from the list (select last entry). Note: The last entry is your registrar contact e-mail address. -- Click continue button.
5. An authentication code is sent to the selected e-mail address.
6. Copy this authentication code and paste it into **Verification Code**: Box. -- Click continue button.
7. Click Finish

Certificates Wizard

1. Now click the **Certificates Wizard** tab
2. From the Certificate Target dropdown menu select **Webserver SSL/TLS Certificate** -- Click continue button.
3. **Generate Private** Key form displayed. Click Skip - We are using our own key.
4. **Submit Certificate Request (CSR)** form opens
    1. Open file UniServer\certs**server.csr**
    2. Copy and past its content into the box displayed.
    3. Click continue button.
5. Message displayed Certificate Request Received -- Click continue button.
6. Select your domain name (from dropdown list) and use www as the subdomain.
    1. Your signed certificate is generated and displayed.
    2. Copy the displayed content and save it to file
    3. UniServer\certs**server.crt**

Download StartCom CA Certificates - Information

Note: This section is for information only. The required certificates are preinstalled. A signed certificate requires a verification chain. To implement this on Apache, two additional certificates are required. Download these from StartSSL

- At StartSSL, log into your control panel.
- Download ca.pem Save to folder UniServer\certs.
- Download sub.class1.server.ca.pem Save to folder UniServer\certs.

Note: If you have a higher validated server certificate you need to use the class2 or class3 intermediate CA certificate.

Install certificates

Copy the two files server.key and server.crt to Apache folder server_certs

- Copy UniServer\certs\server.key to folder UniServer\usr\local\apache2\server_certs
- Copy UniServer\certs\server.crt to folder UniServer\usr\local\apache2\server_certs

Edit SSL Configuration file

Two lines require enabling in Apache's SSL configuration file as follows:

Edit file: UniServer\usr\local\apache2\conf\extra**ssl.conf**

Locate these two lines:

```
#SSLCertificateChainFile C:/UniServer/usr/local/apache2/server_certs/sub.class1.server.ca.pem
#SSLCACertificateFile    C:/UniServer/usr/local/apache2/server_certs/ca.pem
```

Remove the # to enable as shown below:

```
SSLCertificateChainFile C:/UniServer/usr/local/apache2/server_certs/sub.class1.server.ca.pem
SSLCACertificateFile    C:/UniServer/usr/local/apache2/server_certs/ca.pem
```

Note: Your path to folder UniServer may be different.

Local Test

Before going live with your secure server you can test it locally. To do this your domain name (example fred.com) must resolve to an IP address. This is done through an entry in your **hosts** file.

- Edit file: C:\WINDOWS\system32\drivers\etc\hosts
- Add this line: 127.0.0.1 fred.com
- Start the Apache server
- Click **View ssl** button
- The secure test page should be displayed with no alerts.

Note: Replace fred.com with the hostname you used when creating your CSR. Be sure to remove this line when you are ready to go live.

MySQL - Introduction

MySQL is an open source database server. When used in conjunction with PHP scripts, powerful and dynamic server-side applications can be created. MySQL server administration is performed using phpMyAdmin and The Uniform Server's integrated utilities. This page covers the configuration sub-menu.

Contents

Change Password

Change or set a new MySQL root user password.
From UniController, open the menu:

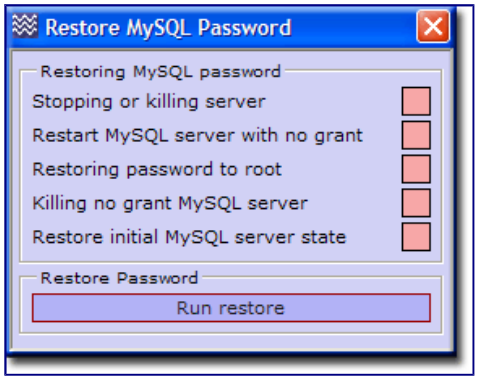
Server Configuration > MySQL > Change Password

- The Set New MySQL Password menu opens (See image on right)
- **A)** Enter a new password
- **B)** Click the **Change Password** button.

The change process is automatic; the indicators show the actions taken.
Note: For detailed information, see MySQL - Change password

Restore Password

Restore MySQL root user password
From UniController, open the menu:



Server Configuration > MySQL > Restore Password > MySQL > Change Password

- The Restore MySQL Password menu opens (See image on right)
- Click the Run Restore button.

The restore process is automatic; the indicators show the actions taken.
Note: For detailed information, see MySQL - Restore password

Edit my.ini

Server Configuration > MySQL > Edit my.ini

Opens the configuration file in notepad, allowing you to directly edit the file, which is UniServer\usr\local\mysql**my.ini**

View Error log file mysql.err

Server Configuration > MySQL > View Error log file mysql.err

This menu option provides a convenient short cut, allowing you to view the MySQL server log file. If you are experiencing problems with the server, this should be your first port of call; its content may shed light on a particular issue.

The path to the log file is: UniServer\usr\local\mysql\data**mysql.err**
Note: Depending on where you extracted UniServer, the path may differ.

A short extract from the file:

```
110706 11:40:41 [Note] Plugin 'FEDERATED' is disabled.
110706 11:40:41 [Note] Plugin 'InnoDB' is disabled.
110706 11:40:43 [Note] Event Scheduler: Loaded 0 events
110706 11:40:43 [Note] C:\UniServer\usr\local\mysql\bin\mysqld1.exe: ready for connections.
Version: '5.5.14'  socket: ''  port: 3306  MySQL Community Server (GPL)
110706 11:55:35 [Note] C:\UniServer\usr\local\mysql\bin\mysqld1.exe: Normal shutdown
110706 11:55:35 [Note] Event Scheduler: Purging the queue. 0 events
110706 11:55:36 [Note] C:\UniServer\usr\local\mysql\bin\mysqld1.exe: Shutdown complete
```

Create Delete Database

There are three methods for creating or deleting a database: via command prompt, phpMyAdmin or UniController. UniController provides a convenient menu option described bellow. To use this option, the MySQL server must be running, otherwise a warning message is produced.

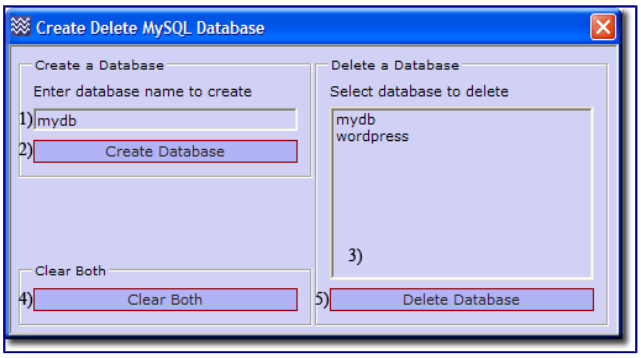
Server Configuration > MySQL > Create Delete Database

Create Database

- 1) Enter a database name, e.g mydb
- 2 Click the "Create Database" button

Delete Database

- 3) Select a database to delete
- 5) Click the "Delete Database" button



Note 1: Clear both (4) deletes the entered name and database selected. Note 2: For detailed information, see MySQL Create Delete Database

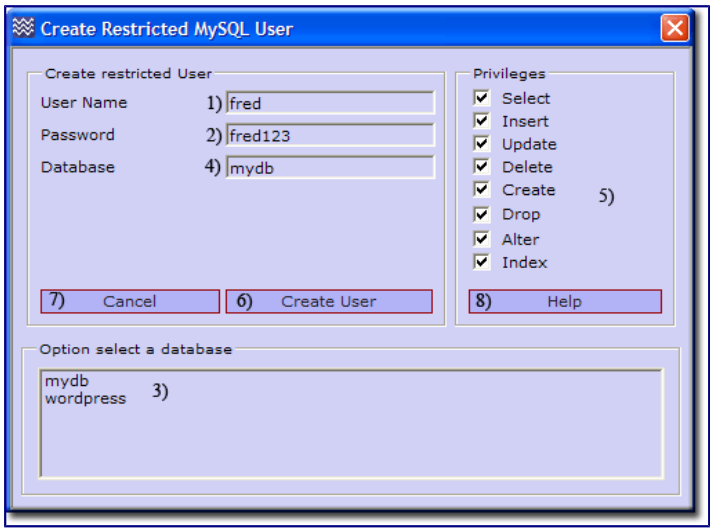
Create Restricted MySQL User

A restricted user has no root privileges and is constrained to a single database.

There are three methods for creating a restricted user: at a command prompt, with phpMyAdmin or through UniController, which provides a convenient menu option described below. To use this option, the MySQL server must be running, otherwise a warning message will show.

Server Configuration > MySQL > Create Restricted MySQL User

- 1) Enter a user name, e.g. fred
- 2) Enter the user password, e.g. fred123
- 3) Select a database to assign to the user. The database name (4) is automatically inserted.
- 5) Select or deselect primary user privileges as required.
- 6) Click “Create User”



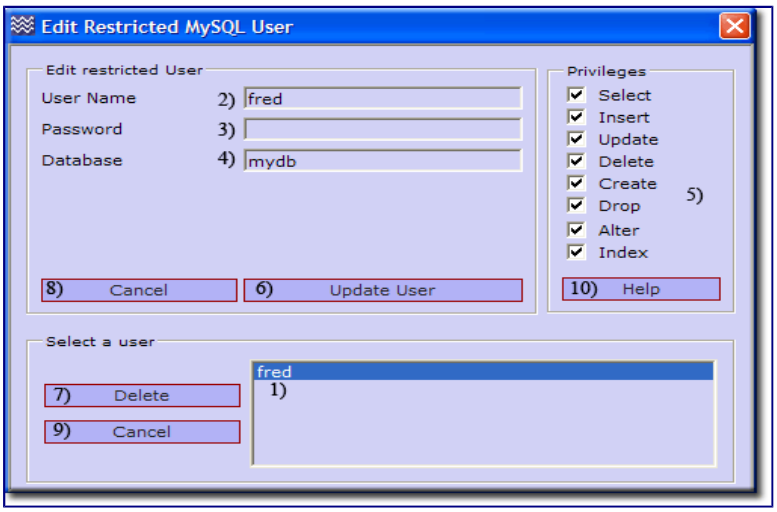
Note 1: Cancel (7) clears both the selected database and user input.

Note 2: For detailed information, see MySQL Create Restricted User

Edit Restricted MySQL User

The same aspects for creating a restricted user (above) apply to editing restricted users. The UniController menu is decribed below.

Server Configuration > MySQL > Edit Restricted MySQL User



Edit Restricted User

- 1) Select the user to edit from the list. This populates the form.
- 3) Enter the current or a new password for the user
- 5) Select or deselect user privileges as required
- 6) Click "Update User" button

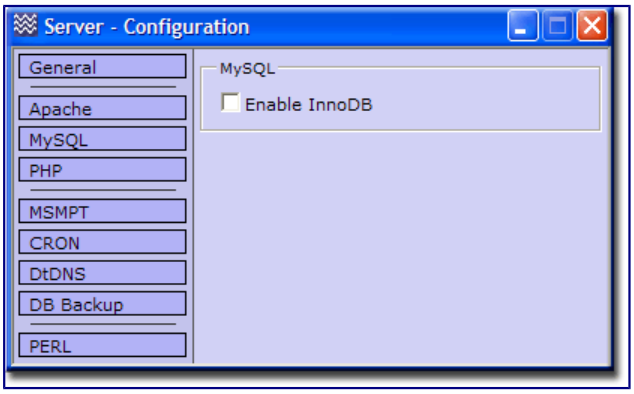
Delete Restricted User

- 1) Select the user to be deleted from list.
- 7) Click "Delete" button

Note 1: Cancel (8) clears only the edit fields.
Note 2: Cancel (9) clears only the selected user.
Note 3: For detailed information, see MySQL Edit Restricted User

InnoDB Enable Disable

Uniform Server default is InnoDB disabled. The following sub-menu allows you to enable or disable InnoDB:



Server Configuration > MySQL > InnoDB Enable/Disable

- Checkbox un-checked InnoDB disabled

- Checkbox checked InnoDB enabled

Note: After changing the state, you must restart the MySQL server to read the configuration.

Files modified: UniServer\usr\local\mysql**my.ini**, UniServer\usr\local\mysql**small_my.ini** and UniServer\usr\local\mysql**medium_my.ini**

General note: MySQL access and privileges

When connecting to a MySQL server, the **host** from which the connection is made, a **user name** and **user password** specified all determine that user's identity. This identity informs MySQL what access is allowed to the server and what privileges are assigned to this user.

```
C:\UniServer\usr\local\mysql\bin>mysql -uroot -proot
mysql> SELECT User, Host, Password FROM mysql.user;
+------+-----------+-------------------------------------------+
| User | Host      | Password                                  |
+------+-----------+-------------------------------------------+
| root | localhost | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| root | 127.0.0.1 | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| root | ::1       | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
| pma  | localhost | *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B |
+------+-----------+-------------------------------------------+
mysql>exit
Bye
```

There are three, root accounts that permit connections from the local host only. Connections can be made by specifying the host name localhost, the IP address 127.0.0.1, or the IPv6 address ::1.

An attempt to connect to the host 127.0.0.1 normally resolves to the localhost account. However, this fails if the server is run with the --skip-name-resolve option, so the 127.0.0.1 account is useful in that case. **The ::1 account is used for IPv6 connections**.

Ref: http://dev.mysql.com/doc/refman/5.5/en/default-privileges.html

MySQL - Change password

There are three methods of changing the MySQL root password: by a command window, phpMyAdmin or UniController. The easiest method is to use UniController, which is fully automated. The other methods require a certain amount of typing and editing, and are more error-prone.

Contents

Set New MySQL root user password using UniController

Change or set a new MySQL root user password.
From UniController, open the menu:

Server Configuration > MySQL > Change Password

- The Set New MySQL Password menu opens (See image on right)
- **A)** Enter a new password
- **B)** Click the **Change Password** button.

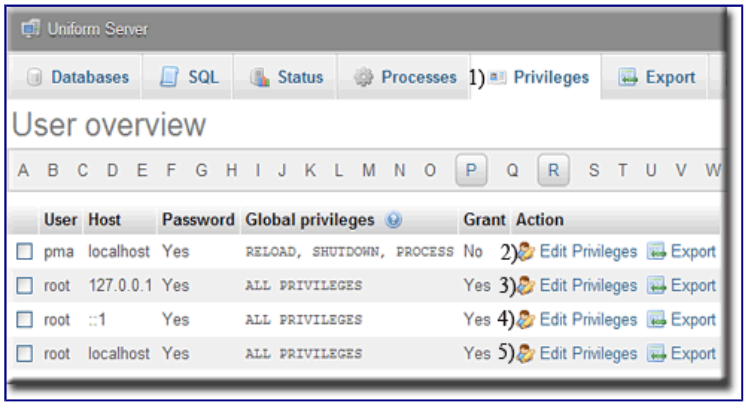
The change process is automatic; the indicators show the actions taken.

Set New MySQL root user password using phpMyAdmin

Setting a new MySQL root password using phpMyAdmin requires four root accounts to be changed along with two configuration files.
Note: The following assumes the current password is **root12** and the new password is **root123**; substitute for these as appropriate.

First start both servers using UniController and then start phpMyAdmin to change the password. Proceed as follows:

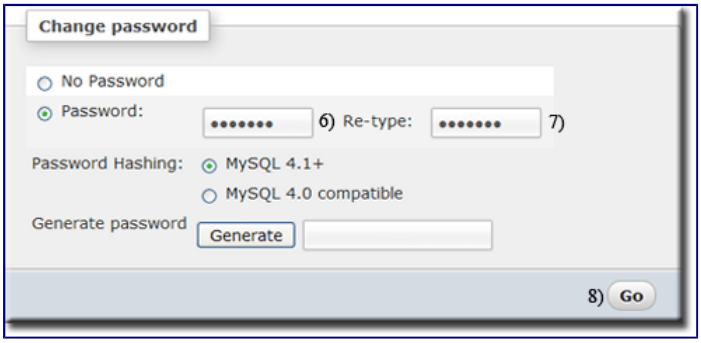
1. From the top menu bar of phpMyAdmin, click the **Privileges** (1) button. The user privileges page opens, as shown on right:
2. Click the first Edit Privileges (2) which opens a new page: "Edit Privileges: User"
    - Scroll down the page to **Change password**
    - Enter new root password for example root123 (6) repeat password (7)
    - Click **Go** (8) to unpdate the user's password
3. Repeat step 2) for each remaining user (3), (4) and (5)
4. Navigate to the end of the Privileges User Overview page



and click **reload the privileges** link.

Note: Do not stop the MySQL server at this point.

5. Edit file UniServer\uni_con\includes**mysql_password** and replace the existing password (root12) with the new root password (root123).
6. Edit file UniServer\uni_con**config_tracker.ini**
   - Navigate to section: **[MYSQL]**
   - Change line **password = root12** by replacing the existing password root12 with the new root password root123.



Note: Perform a quick test. Clear your browser's cache and restart the servers. Run phpMyAdmin and verify that the MySQL server is accessible.

Note: The above root user accounts all use the same root password. This is purely for convenience and one of consistency.

Set New MySQL root user password using MySQL Console

Setting a new MySQL root password using MySQL Console requires four root accounts to be changed along with two configuration files.
Note: The following assumes the current password is **root123** and the new password is **root1234**; substitute for these as appropriate.

First start both servers using UniController and then start MySQL Console to change the password. Proceed as follows:

1. Connect to MySQL server by typing the following at the command window:
   mysql -uroot -proot123
2. At the MySQL prompt type the following:

```
mysql> SET PASSWORD FOR 'root'@'localhost' =
PASSWORD('root1234');
mysql> SET PASSWORD FOR 'root'@'127.0.0.1' =
PASSWORD('root1234');
mysql> SET PASSWORD FOR 'root'@'::1'       =
PASSWORD('root1234');
mysql> SET PASSWORD FOR 'pma'@'localhost'  =
PASSWORD('root1234');
mysql> FLUSH PRIVILEGES;
mysql> exit;
```

Results for steps 1) and 2) are shown on right.

```
C:\UniServer\usr\local\mysql\bin>mysql -uroot -proot123
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('root1234');
Query OK, 0 rows affected (0.03 sec)
mysql> SET PASSWORD FOR 'root'@'127.0.0.1' = PASSWORD('root1234');
Query OK, 0 rows affected (0.00 sec)
mysql> SET PASSWORD FOR 'root'@'::1'       = PASSWORD('root1234');
Query OK, 0 rows affected (0.00 sec)
mysql> SET PASSWORD FOR 'pma'@'localhost'  = PASSWORD('root1234');
Query OK, 0 rows affected (0.00 sec)
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.06 sec)
mysql> exit;
Bye
C:\UniServer\usr\local\mysql\bin>
```

3. Edit file UniServer\uni_con\includes**mysql_password** and replace the existing password (root123) with the new root password (root1234).
4. Edit file UniServer\uni_con**config_tracker.ini**
   Navigate to section: **[MYSQL]**
   Change line **password = root123** by replacing the existing password root123 with the new root password root1234.

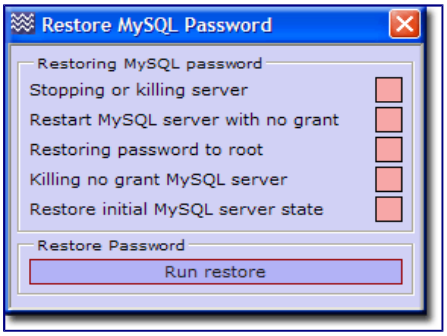
Note: Perform a quick test. Clear your browser's cache and restart the servers. Run phpMyAdmin and verify that the MySQL server is accessible.

MySQL - Restore password

There are two methods for restoring the MySQL root password: by a command window or UniController. The better method is to use UniController which is fully automated. The other method requires running two command windows and a certain amount of typing and editing.

Contents

Restore MySQL root user password using UniController



Restore MySQL root user password
From UniController, open the menu:

Server Configuration > MySQL > Restore Password > MySQL > Change Password

- The Restore MySQL Password menu opens (See image on right)
- Click the Run Restore button.

The restore process is automatic; the indicators show the actions taken.

Restore MySQL root user password using command window

To restore the MySQL root user password, the MySQL server requires restarting with "skip grant tables" option set. This bypasses any security, allowing access to the server. A new password is set and the grant tables updated. The following method uses two command windows and task manger.

Kill MySQL process

If the MySQL server is running, the process must first be stopped. Open windows task manager by pressing **ctrl**, **alt** and **del** keys together. In the window that opens, click on the **Process tab** (default); scroll up or down and locate the file **mysqld1.exe**. Right click on this and select **End Process**; click **Yes** to confirm kill process.

Note: If you have run multi-servers, the process name will differ. For example, mysqldX.exe, where X is a digit. Kill this process

Restart MySQL server skip grant

The MySQL server requires restarting from a command window with "skip grant tables" option set as follows:

Note: The path you installed The Uniform Server may be different; substitute your path as appropriate.

1. Start a command window: (**Start** > click on **Run** > type cmd click **OK**)
2. Type the following lines into the command window:
   - **cd **
   - **cd C:\UniServer\usr\local\mysql\bin**
   - **mysqld1 --skip-grant-tables --user=root**
     Note: Do not copy and paste! You must type it in.
3. A flashing cursor indicates the server is running. Minimise this window (do not close it).

The dialogue in the command window looks similar to this:<r />

```
C:\Documents and Settings\mike>cd \
C:\> cd C:\UniServer\usr\local\mysql\bin
C:\UniServer\usr\local\mysql\bin>mysqld1 --skip-grant-tables --user=root
_
```

Set Password and Update grant tables

Start a second command window to set a new password (root) and update grant tables as follows:

1. Start a command window: (Start > click on Run > type cmd and click OK)
2. Type following lines into the command window:
   - **cd **
   - **cd C:\UniServer\usr\local\mysql\bin**
   - **mysql**
   - **USE mysql;**
   - **UPDATE user SET password=password("root") WHERE user="root";**
   - **UPDATE user SET password=password("root") WHERE user="pma";**
   - **flush privileges;**
   - **exit**
3. Finished.

The dialogue in the command window looks similar to that on the right.

Note: All three user **root** accounts have their passwords changed.

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\mike>cd \
C:\>cd C:\UniServer\usr\local\mysql\bin
C:\UniServer\usr\local\mysql\bin>mysql

mysql> USE mysql;
Database changed
mysql> UPDATE user SET password=password("root") WHERE user="root";
Query OK, 3 rows affected (0.50 sec)
Rows matched: 3  Changed: 3  Warnings: 0
mysql> UPDATE user SET password=password("root") WHERE user="pma";
Query OK, 3 rows affected (0.50 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> flush privileges;
Query OK, 0 rows affected (0.86 sec)

mysql> exit;
Bye

C:\UniServer\usr\local\mysql\bin>
```

Kill process and restart

1. Kill the process **mysqld1** (see the section above)
2. Close both command windows.
3. Edit the file UniServer\uni_con\includes**mysql_password** and replace the existing password with the new root password **root**.

4. Edit the file UniServer\uni_con**config_tracker.ini**
    - Navigate to section: **[MYSQL]**
    - Change line **password = xxx** by replacing the existing password xxx with the new root password **root**.
5. Start the servers using **UniController**
6. Run phpMyAdmin. If the **phpMyAdmin page is displayed**, it confirms the update was **successful**.

MySQL Console

The MySQL server can be administered from a command window. Uniform Server provides a short cut named MySQL console whichs opens a command window with the correct path to the MySQL utilities folder already set. Apart from cosmetics, it is identical to a standard command window.

This page focuses on using MySQL console. It is not intended as a definitive guide but just an introduction to show you how to access and use MySQL client.

Contents

Command window

Using a command window to administer the MySQL server is not specific to The Uniform Server. Running a standard command window is the same for any Windows MySQL installation. The Uniform Server's MySQL console sets the paths and cosmetics transparently. With a command window open you can access the MySQL server utilities mysql client and mysqladmin. MySQL client allows you to run SQL commands, for example to create or delete databases. The following covers opening a standard command window and then The Uniform Server's preferred alternative MySQL console.

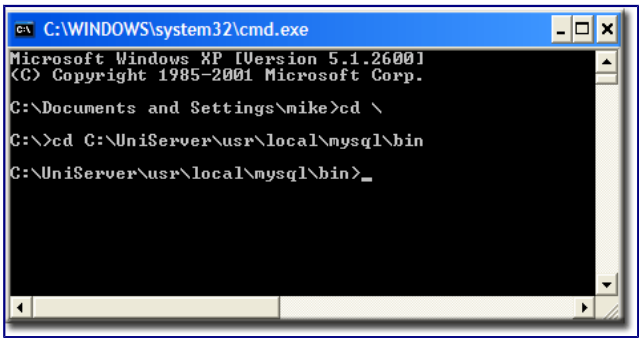
How to run a standard command window

To open a command window, run the command **cmd** from the start menu. Running a MySQL utility program requires navigating to its binary folder.

Note: The path where you installed The Uniform Server may be different; substitute your path as appropriate.

1. Start a command window: (**Start** > click on **Run** > type cmd click **OK**)
2. Alternatively: Click **Start** > **All Programs** > **Accessories** > click **Command Prompt**
3. Type following lines into the command window:
   - **cd **
   - **cd C:\UniServer\usr\local\mysql\bin**

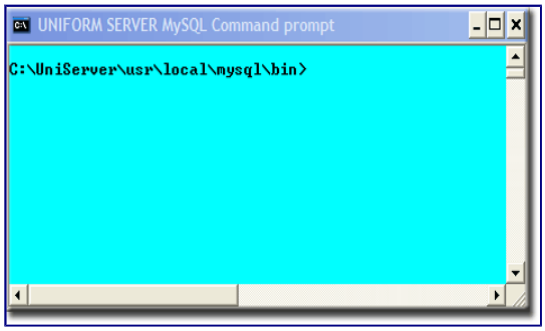
The dialogue in the command window looks similar to that on the right.



Note: To use the command window you must have the MySQL server running.

MySQL console command window short cut

Running MySQL console from UniController has a number of advantages. The button is enabled only when the MySQL server is running. Server location tracking allows the correct folder to be automatically selected when the console window is open. Opening a console window requires only a single button click.



1. Start UniController
2. Start MySQL server

3. Click **MySQL console** button. Opens window shown on right

Quick MySQL binary folder check

When running MySQL utilities, the working directory is changed to the MySQL folder: UniServer\usr\local\mysql\bin.

To check the working directory, type the **dir** command into either of the above windows. The result should be as on the right.

- mysql.exe - MySQL client runs SQL commands
- mysqladmin.exe - MySQL admin
- mysqldump.exe - Database dump

```
C:\UniServer\usr\local\mysql\bin>dir
Directory of C:\UniServer\usr\local\mysql\bin
21/06/2011 19:55    4,001,792 libmysql.dll
21/06/2011 19:54    4,106,240 mysql.exe
21/06/2011 19:54    4,037,632 mysqladmin.exe
21/06/2011 19:54    8,153,600 mysqld1.exe
21/06/2011 19:54    4,094,976 mysqldump.exe
C:\UniServer\usr\local\mysql\bin>
```

Start MySQL Client

To start MySQL Client, type the following at the command window:

- mysql -uroot -proot

Note 1: Do not copy and paste! Type the line in as shown.

- -u stands for user name. Immediately follow that with the user name (no spaces)
- -p stands for user password. Immediately follow that with the password (no spaces)

Note 2: The **mysql**> prompt indicates you are in **MySQL monitor** and it is ready to receive your commands.

Note 3: Quit MySQL monitor by typing the command **exit**

On successful connection, you will receive a message similar to this:

```
C:\UniServer\usr\local\mysql\bin>mysql -uroot -proot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 53
Server version: 5.5.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

mysql> exit
Bye
```

MySQL Create delete Database

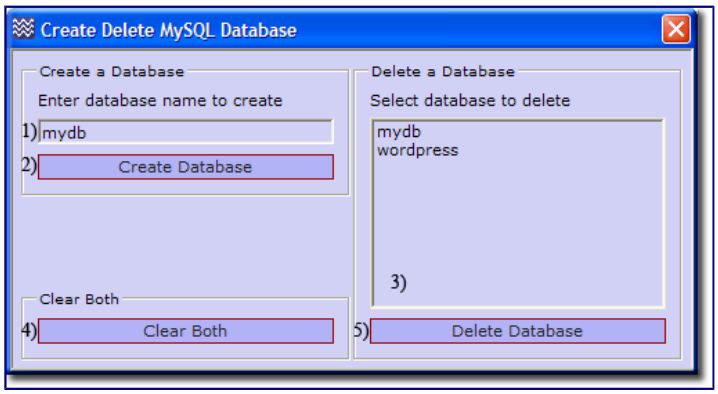
There are three methods for creating or deleting a database; you can use a command window, phpMyAdmin or UniController. UniController provides a convenient menu option described bellow. To use this option ensure the MySQL server is running otherwise a warning message is produced.

Contents

Create Delete Database using UniController

Server Configuration > MySQL > Create delete Database

Create Database



- 1) Enter a databse name e.g mydb
- 2 Click "Create Database" button

Delete Database

- 3) Select a database to delete
- 5) Click "Delete Database" button

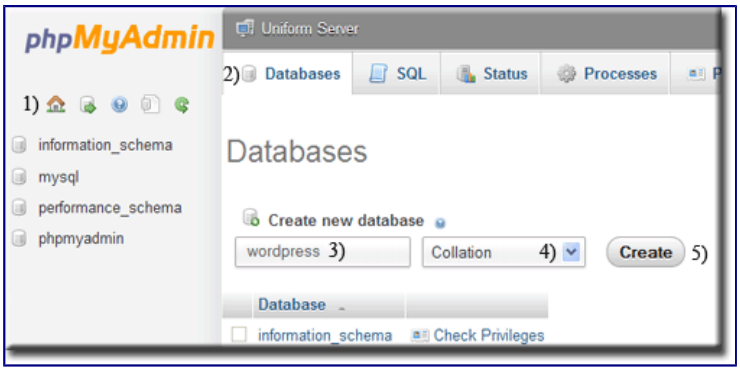
Note 1: Clear both (4) deletes entered name and database selected.

Create Delete Database using phpMyAdmin

Start UniController and click phpMyAmin button. To create or delete a database proceed as follows:

Create Database

When first started, the phpMyAdmin home page is displayed. You can always return to this page by clicking the home icon (1)

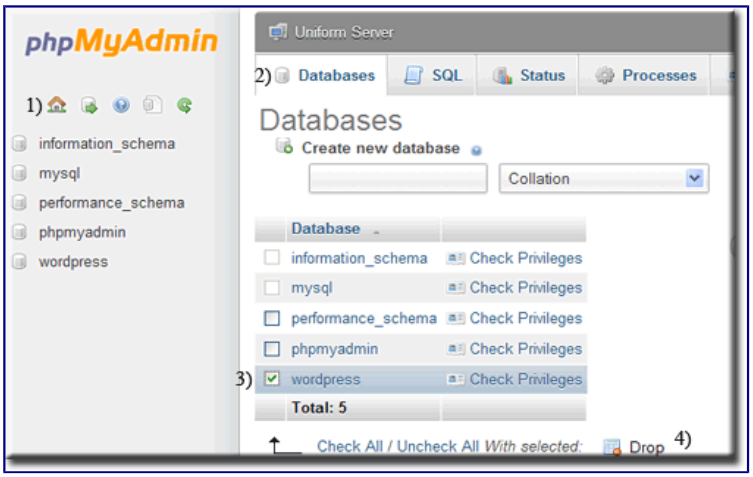


- 1) If not at the home page, click **home icon**
- 2) From the top menu bar select **Databases**
- 3) Enter name of database to create; for example, "wordpress"
- 4) Optionally select a collation from the dropdown. No selection default is used (see note).
- 5) Click **Create** button, a created confirmation is displayed.

Note: On the home page the default collection is set to **utf8_general_ci**

Delete Database

When first started, the phpMyAdmin home page is displayed. You can always return to this page by clicking the home icon (1)



- 1) If not at the home page, click **home icon**
- 2) From top menu bar select **Databases**
- 3) Select database to delete (drop); for example, "wordpress"
- 4) Click **Drop** button; an alert box is displayed.
- 5) Click **Yes** on alert box "You are about to DESTROY a complete database!"
- 6) A confirmation is displayed "1 databases have been dropped successfully."

Create Delete Database using command window

Click MySQL console, which opens a command window. You can use either of the MySQL utilities, Client or Admin, to create and delete a database as follows:

Command window - mysqladmin

Create a database

To create a database named **joomla**, enter the following command into the window:

- mysqladmin.exe --user=root --password=root create joomla

Delete a database

To delete a database named **joomla**, enter the following command into the window:

- mysqladmin.exe --user=root --password=root --force drop joomla

Note: You can specify the MySQL port to use:

- mysqladmin.exe --port=3306 --user=root --password=root create joomla
- mysqladmin.exe --port=3306 --user=root --password=root --force drop joomla

Command window - mysql Client

Create a database

C:\UniServer\usr\local\mysql\bin>mysql -uroot -proot

To create a database named **wordpress**, enter the following commands into the window:

- mysql -uroot -proot
- At the mysql prompt type: CREATE DATABASE wordpress;
- At the mysql prompt type: exit

```
mysql> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.03 sec)

mysql> exit
Bye
```

Delete a database

To delete a database named **wordpress**, enter the following commands into the window:

- mysql -uroot -proot
- At the mysql prompt type: DROP DATABASE wordpress;
- At the mysql prompt type: exit

```
C:\UniServer\usr\local\mysql\bin>mysql -uroot -proot

mysql> DROP DATABASE wordpress;
Query OK, 0 rows affected (0.17 sec)

mysql> exit
Bye
```

Note: If you have changed the MySQL root password, remember to substitute (-p**root**) **root** with your password in the above.

MySQL Create Restricted User

MySQL privileges allow each of the users to access and utilize only the areas they need to perform their tasks. This prevents a user from accidentally accessing an area where he or she should not have access this adds to the security of the MySQL server.

When you connect to a MySQL server, the host from which you connect and the user name you specify determines your identity. With this information, the server then grants privileges based upon this identity. This page focuses on creating a user with restricted privileges.

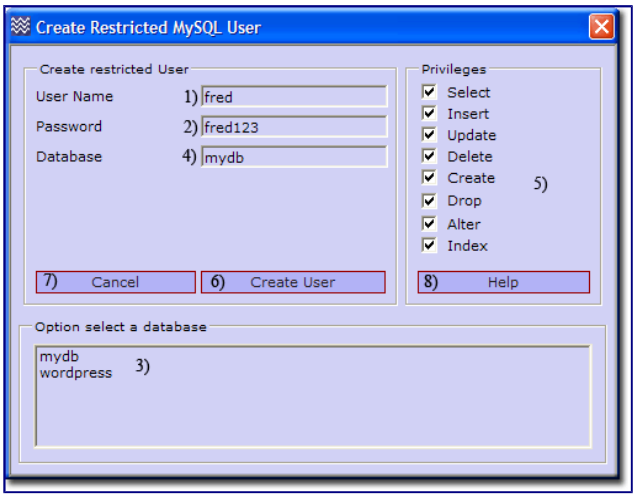
There are three methods for creating a restricted user; you can use a command window, phpMyAdmin or UniController. UniController provides a convenient menu option described bellow. To use this option ensure the MySQL server is running otherwise a warning message is produced.

Contents

Create Restricted MySQL User using UniController

Server Configuration > MySQL > Create Restricted MySQL User

- 1) Enter a user name, e.g. fred
- 2) Enter the user password, e.g. fred123
- 3) Select a database to assign to the user. The database name (4) is automatically inserted.
- 5) Select or deselect primary user privileges as required.
- 6) Click “Create User”



Note 1: Cancel (7) clears both the selected database and user input.

Create Restricted MySQL User using phpMyAdmin

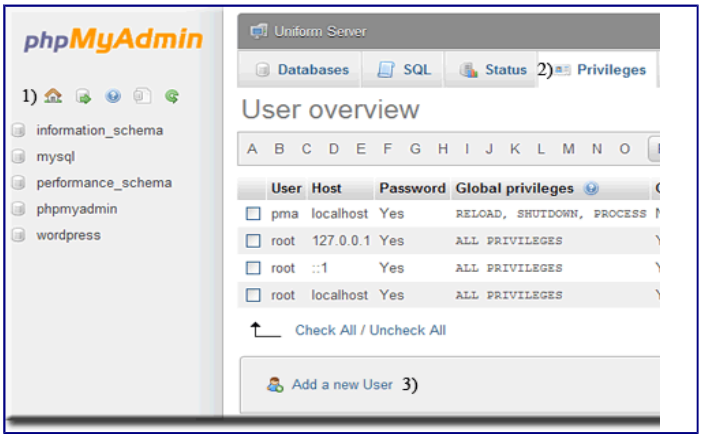
Start UniController and start both servers, then click phpMyAmin button. To create a restricted user, proceed as follows:

Note: Assume the user to create is **mike123** and password **pass123**. The database **wordpress** that the user will be assigned has aleady been created.

Open Add new user page

When first started the phpMyAdmin home page is displayed; you can always return to this page by clicking the home icon (1)

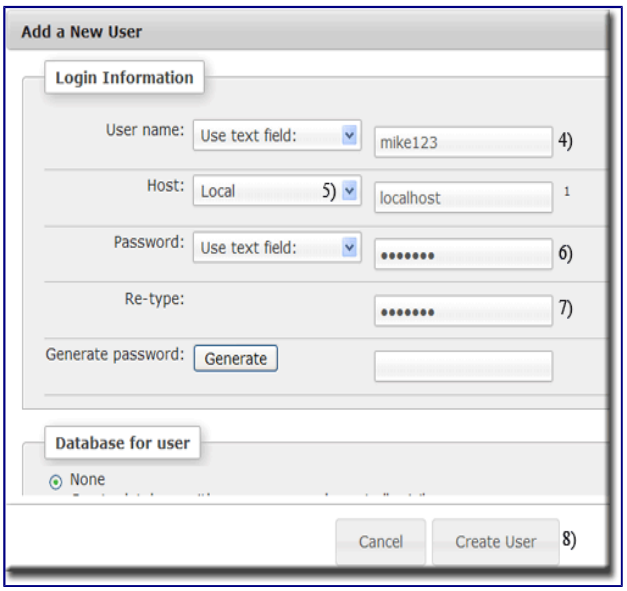
- 1) If not at the home page, click **home icon**
- 2) From the top menu bar, select **Privileges**
- 3) Click **Add a new User**



The **Add a new User** page is displayed.

Create new user

Creating a new user requires only the login information; the other fields are not required. We will assign this user to a database and set appropriate privileges as a second step.



From new user page:

- 4) Enter user name; example mike123
- 5) From the drop down menu, select Local. This user is restricted to localhost
- 6) Enter password; example pass123
- 7) Re-enter password as entered above.
- 8) Click Create User; confirmation produced.

Note: Leave all other fields set to their defaults:

- Database for user: **None** radio button selected
- Privileges: all boxes **Uncheck**
- Resource limits: all values set to **0**

Assign user to a database

The new user created has now been added to the Privileges User overview page. This new entry allows a user to be assigned to a database as follows:

After creating a new use you will be at the privileges page.

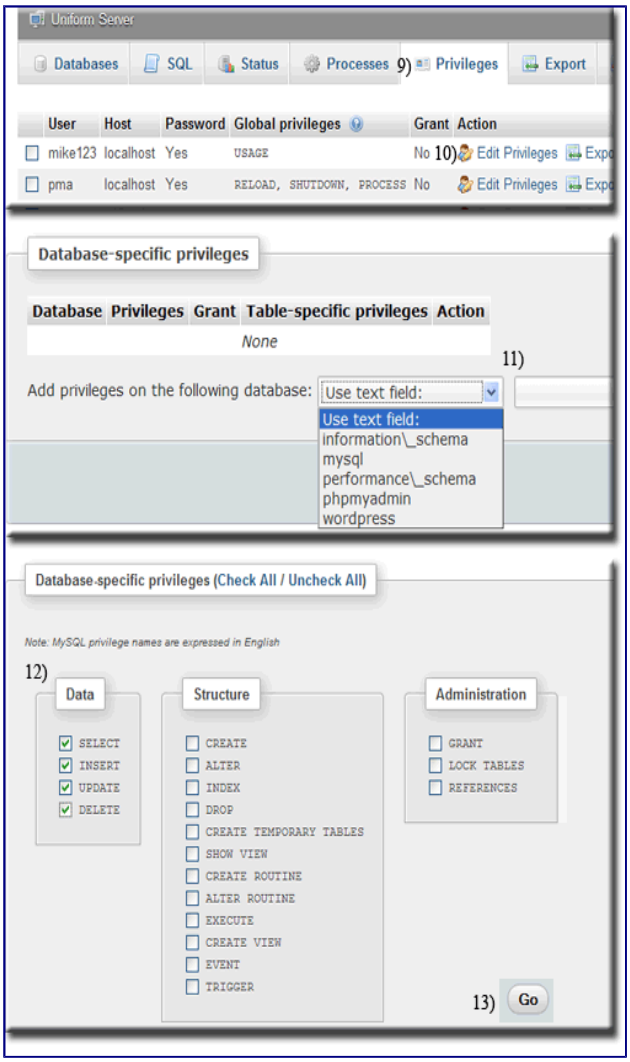
- 9) If not in Privileges page, from home menu select **Privileges**
- 10) Click **Edit Privileges** for user mike123

A new page opens displaying privileges.

- 11) In the Database-specific privileges section, click the drop down menu and select the database **wordpress**

This directs you to the Database-specific privileges section

- 12) Select the privileges required for your restricted user.
- 13) Click Go; this assigns the user to the database with the privileges selected.



Create Restricted MySQL User using command window

Creating a user with restricted privileges and assigning that user to an existing database can also be performed using a single SQL line.

Example user
User name to create: **fred123** User password: **fredpass** Existing database: **wordpress**

Click MySQL console, which opens a command window. Use the MySQL Client utility from here to create our example user with restricted privileges as follows:

- mysql -uroot -proot
- GRANT SELECT, INSERT, UPDATE, DELETE ON wordpress.* TO 'fred123'@'localhost' IDENTIFIED BY 'fredpass';
- exit

The result is shown below:

```
C:\UniServer\usr\local\mysql\bin>mysql -uroot -proot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.5.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> GRANT SELECT, INSERT, UPDATE, DELETE ON wordpress.* TO 'fred123'@'localhost' IDENTIFIED BY 'fredpass';
Query OK, 0 rows affected (0.09 sec)

mysql> exit
Bye

C:\UniServer\usr\local\mysql\bin>
```

Note: If you have changed the MySQL root password, remember to substitute (-proot) root with your password in the above.

Delete Restricted MySQL User using command window

Delete a user (example fred123) with restricted privileges as follows:

Click MySQL console which opens a command window. Use the MySQL Client utility from here to delete our example user fred123:

Enter the following in a command window:

- mysql -uroot -proot
- DROP USER 'fred123'@'localhost';
- exit

```
C:\UniServer\usr\local\mysql\bin>mysql -uroot -proot

mysql> DROP USER 'fred123'@'localhost';
Query OK, 0 rows affected (0.08 sec)

mysql> exit
Bye
```

The result is shown to the right:

Note: If you have changed the ySQL root password, remember to substitute (-p**root**) **root** with your password in the above.

MySQL Edit Restricted User

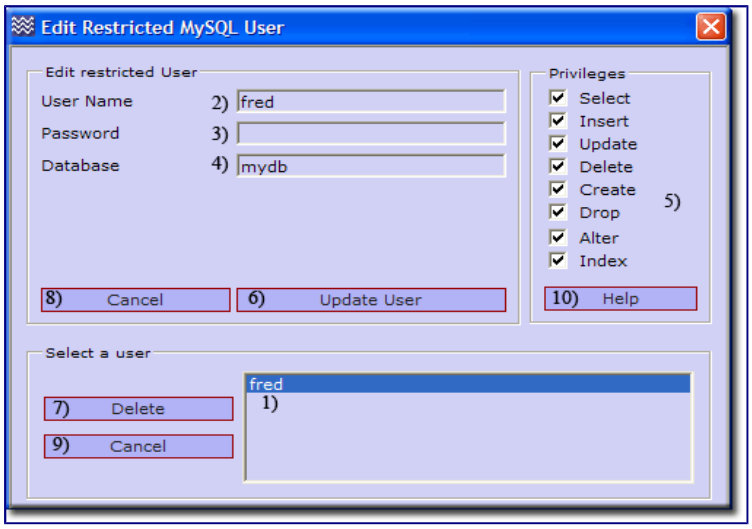
There are three methods for editing a restricted user; you can use a command window, phpMyAdmin or UniController. UniController provides a convenient menu option described bellow. To use this option ensure the MySQL server is running otherwise a warning message is produced.

Contents

Edit Restricted MySQL User using UniController

Server Configuration > MySQL > Edit Restricted MySQL User

Edit Restricted User



- 1) Select user to edit from the list, which populates the upper form
- 3) Enter the old or a new password for the user
- 5) Select or deselect user privileges as required
- 6) Click "Update User" button

Delete Restricted User

- 1) Select user to be deleted from the list.
- 7) Click "Delete" button

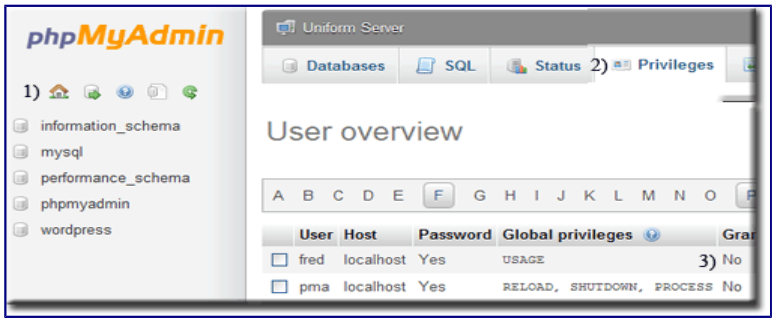
Note 1: Cancel (8) and (9) clear both the edit fields and the user selection.

Edit Restricted MySQL User using phpMyAdmin

Start UniController and start both servers, then click phpMyAmin button. To edit a restricted user, proceed as follows:

Note: Assume the user **fred** has already been created.

When first started the phpMyAdmin home page is displayed; you can always return to this page by clicking the home icon (1)



- 1) If not at the home page, click **home icon**
- 2) From the top menu bar, select **Privileges**
- 3) Under "Action", Click **Edit Privileges**

The **Edit Privileges: User 'fred'@'localhost'** page is displayed.

From this page you can change the user privileges, the database(s) the user is assigned to and privileges, password and login information.

Edit Restricted MySQL User using command window

Editing an existing user with restricted privileges can be performed using the MySQL Client.

You can use REVOKE to remove some or all privileges or alternatively use GRANT to add additional privileges. This example assumes a user fred has been created with the privileges GRANT SELECT, INSERT, UPDATE, DELETE assigned on database wordpress. You can revoke (remove) privileges; for example, the following command removes INSERT, UPDATE and DELETE.

- REVOKE INSERT, UPDATE, DELETE ON wordpress.* FROM 'fred'@'localhost' ;

You can grant (add) privileges; for example, the following command adds INSERT and UPDATE.

- GRANT INSERT, UPDATE ON wordpress.* TO 'fred'@'localhost' ;

You can display the grants assigned with the following command:

- SHOW GRANTS FOR 'fred'@'localhost';

Example for the above commands. Click MySQL console, which opens a command window. Then we run the MySQL Client with the following commands:

- mysql -uroot -proot
- SHOW GRANTS FOR 'fred'@'localhost';
- REVOKE INSERT, UPDATE, DELETE ON wordpress.* FROM 'fred'@'localhost' ;
- SHOW GRANTS FOR 'fred'@'localhost';
- GRANT INSERT, UPDATE ON wordpress.* TO 'fred'@'localhost' ;
- SHOW GRANTS FOR 'fred'@'localhost';
- exit

The results are shown below.

```
C:\UniServer\usr\local\mysql\bin>mysql -uroot -proot

mysql> SHOW GRANTS FOR 'fred'@'localhost';
+-----------------------------------------------------------------------------------------------------------+
| Grants for fred@localhost                                                                                 |
+-----------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'fred'@'localhost' IDENTIFIED BY PASSWORD
'*F5F0B28BD93FCF0C77FD96BB97BBC745ED8EA6BC' |
| GRANT SELECT, INSERT, UPDATE, DELETE ON `wordpress`.* TO 'fred'@'localhost'                               |
+-----------------------------------------------------------------------------------------------------------+
2 rows in set (0.00 sec)
```

```
mysql> REVOKE INSERT, UPDATE, DELETE ON wordpress.* FROM 'fred'@'localhost' ;
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW GRANTS FOR 'fred'@'localhost';
+-----------------------------------------------------------------------------------------------------------+
| Grants for fred@localhost                                                                                 |
+-----------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'fred'@'localhost' IDENTIFIED BY PASSWORD
'*F5F0B28BD93FCF0C77FD96BB97BBC745ED8EA6BC' |
| GRANT SELECT ON `wordpress`.* TO 'fred'@'localhost'                                                       |
+-----------------------------------------------------------------------------------------------------------+
2 rows in set (0.00 sec)

mysql> GRANT INSERT, UPDATE ON wordpress.* TO 'fred'@'localhost' ;
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW GRANTS FOR 'fred'@'localhost';
+-----------------------------------------------------------------------------------------------------------+
| Grants for fred@localhost                                                                                 |
+-----------------------------------------------------------------------------------------------------------+
| GRANT USAGE ON *.* TO 'fred'@'localhost' IDENTIFIED BY PASSWORD
'*F5F0B28BD93FCF0C77FD96BB97BBC745ED8EA6BC' |
| GRANT SELECT, INSERT, UPDATE ON `wordpress`.* TO 'fred'@'localhost'                                       |
+-----------------------------------------------------------------------------------------------------------+
2 rows in set (0.00 sec)
mysql> exit
Bye
```

PHP - Introduction

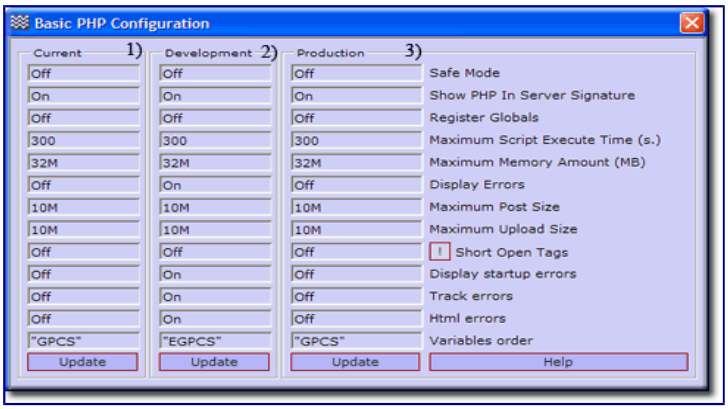
PHP is a general-purpose scripting language designed for creating dynamic web pages. The Apache server uses a PHP processor module to generate the final web page being served. It interprets code embedded into HTML documents. PHP has been pre-configured for production and development. One of Unicontroller's sub-menus allows you to easily switch between the two configurations. This page covers the configuration sub-menu and its additional features for maintaining PHP.

Contents

Edit Basic Configuration

Server Configuration > PHP > Edit Basic Configuration

This menu item allows you to change commonly configured PHP options.



1. **Current**: This file is the active file used by PHP for configuration. Switching to development or production overwrites this file. Consider any changes you make to this file as temporary.
2. **Development**: Configuration file used for development. Errors will be written to the screen. You can enable other parameters for testing.
3. **Production**: Configuration file for production ini file. Prevents errors being written to screen. The initial settings tighten security. For example, globals are off by default. Any changes you wish to be permanent should be made in this file.

Note 1: After making changes, click the corresponding update button. This writes the modified data to the associated configuration file.

Note 2: Changes to the current configuration are used next time the servers are started.

Note 3: For development and production configuration files to become effective, you need to switch over the appropriate file (see switching.) and restart Apache.

Note 4: Do not enable **short open tags**. Doing so will conflict with other languages such as XML's open tag. For detailed information, and the exception, see Short open tags

Directly edit configuration files

The following menu items allow you to edit the specific PHP configuration file in notepad.

Edit Current Configuration file: php.ini

Server Configuration > PHP > Edit Current Configuration file: php.ini

- File path: UniServer\usr\local\php**php.ini**
- This configuration file is the active configuration. When Apache is started, the contents of

this file will configure PHP.
- **Note 1**: When either production or development configuration file is selected, as explained in the next section on switching, any edits in this file will be overwritten. Consider changes to this file as temporary. This is ideal for testing without making permanent changes.
- **Note 2**: The initial version of the current configuration file is a copy of the production version.

Edit Production Configuration file: php.ini_production

Server Configuration > PHP > Edit Production Configuration file: php.ini_production

Server Configuration > PHP > Edit Production Configuration file: php.ini_production

- File path: UniServer\usr\local\php**php.ini_production**
- This configuration file is pre-configured for production use.
- **Note**: In a default installation of The Uniform Server, the current configuration is a copy of this file (php.ini = php.ini_production).

Edit Development Configuration file: php.ini_development

Server Configuration > PHP > Edit Development Configuration file: php.ini_development

- File path: UniServer\usr\local\php**php.ini_development**
- This configuration file is pre-configured for development use.
- **Note**: Errors will be displayed in a user's browser.

Edit Command Line Configuration file: php-cli.ini

Server Configuration > PHP > Edit Command Line Configuration file: php-cli.ini

- File path: UniServer\usr\local\php**php-cli.ini**
- This configuration file is pre-configured for command-line operation.
- **Note**: Do not change the existing directives. These are required for correct operation of UniController's command-line scripts. You can add any directives you want that are not included.

Configuration file switching

The following two menu items allow you to easily switch between PHP production and development configuration files.

Switch to Production Configuration file

Server Configuration > PHP > Switch to Production Configuration file

- File path: UniServer\usr\local\php**php.ini_production**

- This configuration file is pre-configured for production use.
- **Note 1**: In a default installation of The Uniform Server, the current configuration is a copy of this file (php.ini = php.ini_production).
- **Note 2**: Errors are not displayed; these are directed to Apache's log file.
- **Note 3**: Switching **overwrites** the php.ini configuration file.
- **Note 4**: After switching you must **restart** the servers or the new configuration will not be recognised.

Switch to Development Configuration file

Server Configuration > PHP > Switch to Development Configuration file

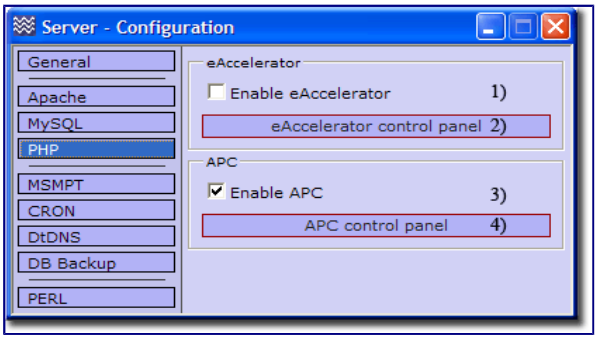
- File path: UniServer\usr\local\php**php.ini_development**
- This configuration file is pre-configured for development.
- **Note 1**: Errors are displayed in a user's browser and also directed to log file
- **Note 2**: Switching **overwrites** the php.ini configuration file. The php.ini file is replaced by a copy of php.ini_development.
- **Note 3**: After switching you must **restart** the servers or the new configuration will not be recognised.

PHP Accelerators

The Uniform Server provides two PHP accelerators: APC and eAccelerator. The following UniController sub-menu allows you to enable or disable these accelerators and open their corresponding controller in your default browser.

Server Configuration > PHP > PHP Accelerators

eAccelerator



- Check box 1) enables or disables eAccelerator. With the box checked eAccelerator is enabled.
- Control panel button 2) opens eAccelerator's control panel in the default browser.

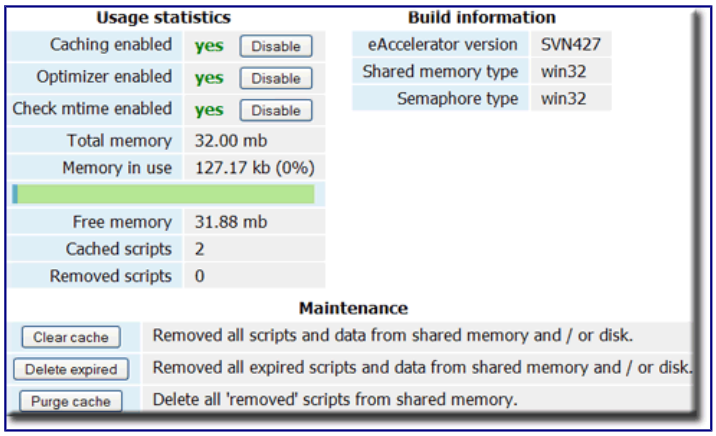
APC

- Check box 3) enables or disables APC. With the box checked APC is enabled.
- Control panel button 2) opens APC's control panel in the default browser.

Note: After enabling or disabling either accelerator, the Apache server must be restarted in order for the new configuration to become effective.

eAccelerator control panel

eAccelerator is a free open-source PHP accelerator and optimizer. It increases the performance of PHP scripts by caching them in their compiled state, so that the overhead of compiling is almost completely eliminated.

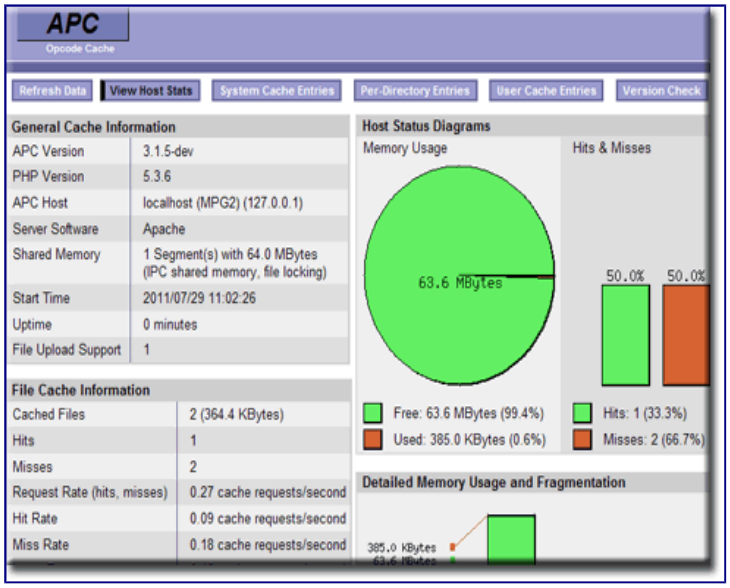


The Uniform Server's default installation has eAccelerator **enabled**.

The eAccelerator control panel is shown on the right.

APC control panel

The Alternative PHP Cache (APC) is a free and open opcode cache for PHP. It is a robust framework for caching and optimizing PHP intermediate code.

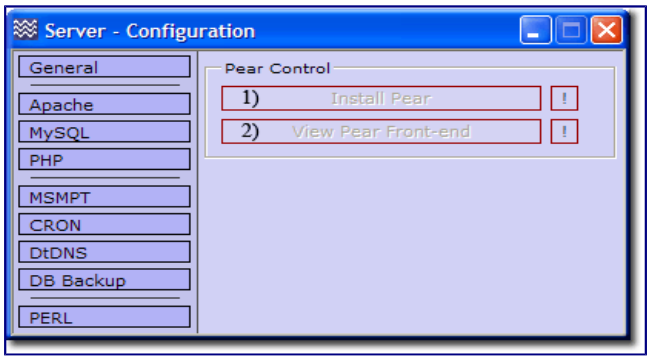


The APC control panel is shown on the right.

For detailed information, see PHP APC

PEAR control panel

Server Configuration > PHP > PEAR control panel



The PHP Extension and Application Repository, or PEAR for short, contains re-usable code. This menu option opens the PEAR Controller. It consists of two buttons, operating as follows:

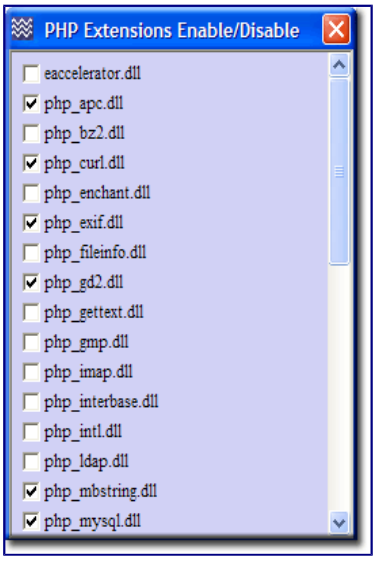
- 1) Install PEAR Click to install PEAR. For detailed information see PEAR Auto Install
- 2) View PEAR front-end Opens the PEAR front-end in the default browser.

Note 1: Apache must be running, else the buttons remain greyed out.

Note 2: For a short tutorial on how to use PEAR, see Using PEAR Frontend

Note 3: You can also view the PEAR Front-end by entering the following into your browser: '"http://localhost/us_pear/index.php"'

PHP Extensions Enable Disable



The following sub-menu allows you to enable or disable PHP extensions:

Server Configuration > PHP > PHP Extensions Enable/Disable

- Checkbox un-checked Extension disabled
- Checkbox checked Extension enabled

Note: After changing state you must **restart** the Apache server for the new configuration to be recognised.

Files modified:

UniServer\usr\local\php**php.ini**
UniServer\usr\local\php**php.ini_development**
UniServer\usr\local\php**php.ini_production**

Note: PHP mail() function

Uniform Server has been pre-configured for sending mail using the open source **msmtp** client. For reference the configuration section contained in confiuration files php.ini, php.ini_development and php.ini_production is shown below:

```
; For Unix only.  You may supply arguments as well (default: "sendmail -t -i").
;sendmail_path = "/usr/bin/sendmail.exe -t"

sendmail_path = "**path**/UniServer/msmtp/msmtp.exe --file=**path**/UniServer/msmtp/msmtprc.ini  -t"

; Force the addition of the specified parameters to be passed as extra parameters
; to the sendmail binary. These parameters will always replace the value of
; the 5th parameter to mail(), even in safe mode.
;mail.force_extra_parameters =

; Add X-PHP-Originating-Script: that will include uid of the script followed by the filename
mail.add_x_header = On
```

Note: **path** is automatically set to where Uniform Server is located; for example: C:/Coral_4

Before using the PHP mail function, the msmtp client must be set up. See the MSMTP page for details.

PHP - Short open tags

PHP short open tags allow you to write slightly less code! These were most beneficial when inserting pieces of PHP code into HTML however:

Warning

- Do not use the short form (**<? ?>**) of PHP open tag.
- Use the long form of PHP open tag (**<?php ?>**).
- **Always** switch "short form open tags" to "off" in the PHP configuration file.

Contents

Why

Using short tags is discouraged because they are in conflict with XML's open tag - '<?xml'. If the file is actually XML, the PHP interpreter will process everything after the '<?' as PHP code, resulting in a parsing error.

Another reason to always use full PHP opening tags is to avoid the unknown. For example, a hosting provider that doesn't allow short open tags (<? ?>) on their servers.

Testing older scripts

For testing older scripts, you can switch short open tags "on" temporarily. After testing, be sure to switch short form open tags "off" and retest the scripts

Convert older scripts as follows:

Replace all occurrences of (**<? ?>**) with (**<?php ?>**)

Replace all occurrences of (**<?=**) with (<**?php echo**)

Replace all occurrences of (**<% %>**) with (**<?php ?>**)

There's a similar issue with ASP-like **<% %>** tags. The Uniform Server default is asp_tags = Off
Replace all occurrences of (**<% %>**) with (**<?php ?>**)

PHP - PEAR Auto Install

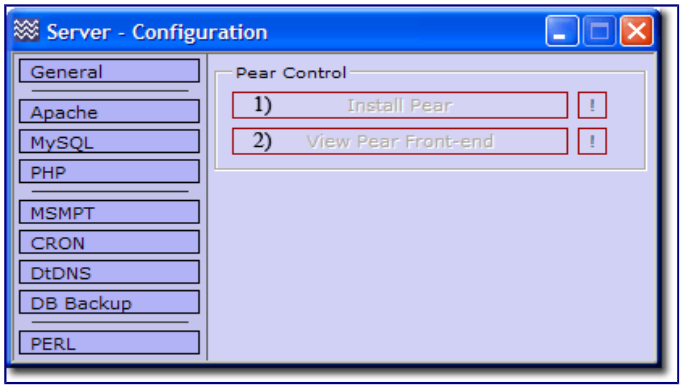
The PHP Extension and Application Repository, or PEAR for short contains re-usable code. This page covers PEAR installation using go-pear, which is integrated into The Uniform Server's architecture.

Contents

Install PEAR core package

Open Pear control panel as follows:

Server Configuration > PHP > Pear Control Panel



- Click **Install Pear** button (1)
- PEAR intaller opens in your browser. There is nothing to do other than to click OK at each prompt as follows:
  - Welcome to go-pear - Click **Next >>**
  - At the end of Configuration section - **Click Install**
- Installation process starts; this will take a while to complete.
- Expected results are shown on right.

Note: Your Installation path may be different!

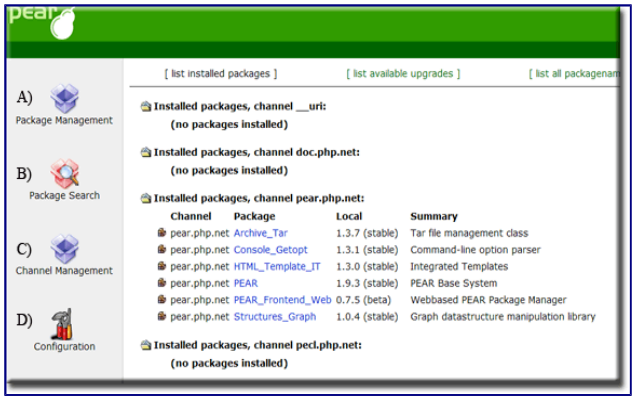
```
Starting installation ...
Loading zlib: ok

Bootstrapping Installer.....................
Bootstrapping PEAR.php.............(remote) ok
Bootstrapping Archive/Tar.php.............(remote) ok
Bootstrapping Console/Getopt.php.............(remote) ok

Extracting installer.....................
Downloading package: PEAR.............ok
Downloading package: Structures_Graph....ok

Preparing installer.....................
Updating channel "doc.php.net"
Update of Channel "doc.php.net" succeeded
Updating channel "pear.php.net"
Update of Channel "pear.php.net" succeeded
Updating channel "pecl.php.net"
Update of Channel "pecl.php.net" succeeded

Installing selected packages.....................
Downloading and installing package: PEAR.............ok
Installing bootstrap package: Structures_Graph.......ok
Downloading and installing package: Archive_Tar-stable.......ok
Downloading and installing package: Console_Getopt-stable.......ok
Downloading and installing package: PEAR_Frontend_Web-beta.......ok

Writing WebFrontend file ... ok

Installation Completed !
      Note: To use PEAR without any problems you need to add your
PEAR Installation path (C:\UniServer\home\us_pear\PEAR)
to your include_path.

Using a .htaccess file or directly edit httpd.conf would be working
solutions for Apache running servers, too.

Note: Above path already pre-configured in The Uniform Server hence
     you are ready to go.
```

For more information about PEAR, see:
PEAR FAQ
PEAR Manual

Thanks for using go-pear!

Start Web Frontend of the PEAR Installer >>

- Scroll to end of installation section.
- Click the link **Start Web Frontend of the PEAR Installer**
- Web Front End opens in your default browser as shown on right.



Open PEAR Web Frontend

After installing PEAR, you can access the web-front-end at any time using either of the following methods:

Method 1
Run the web-front-end from UniController menu as follows;

Server Configuration > PHP > Pear Control Panel

Click **View Pear Front-end** button

Method 2
Alternatively enter the following into your browser:

http://localhost/us_pear/index.php

Using PEAR Frontend

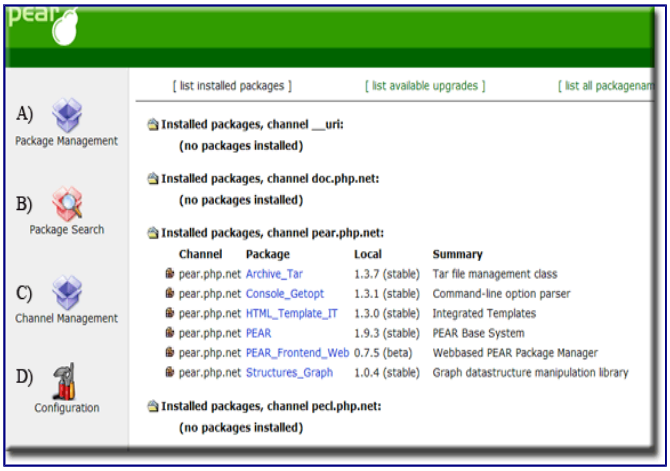
Open the PEAR front-end using one of the methods explaind above.

Installing PEAR packages is very easy. To confirm our PEAR installation is working and to demonstrate how to install packages, we will be using a package named **HTML_QuickForm**

Update the Channel Management system

Before downloading any package, **update** the Channel Management system

- C) Click on **Channel Management** - Opens Registered Channels page.
- From this page, click on **Update All Channels**



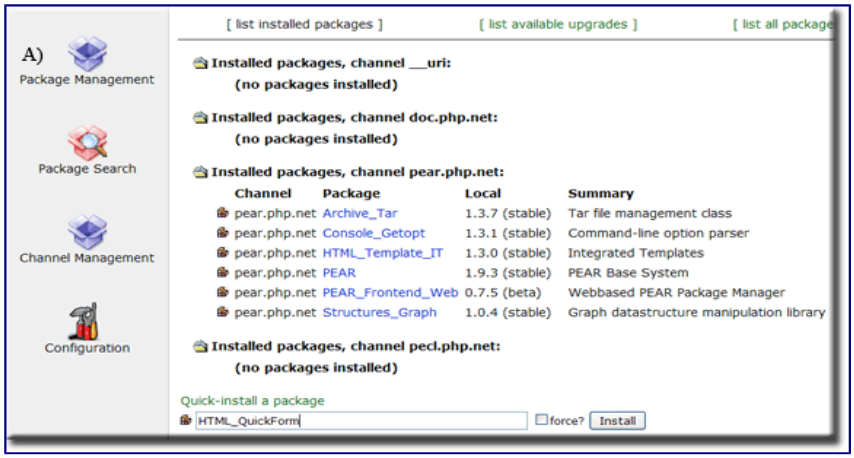
You will receive something similar this:

```
Updating channel "doc.php.net"
Update of Channel "doc.php.net" succeeded
Updating channel "pear.php.net"
Channel "pear.php.net" is up to date
Updating channel "pecl.php.net"
Channel "pecl.php.net" is up to date
```

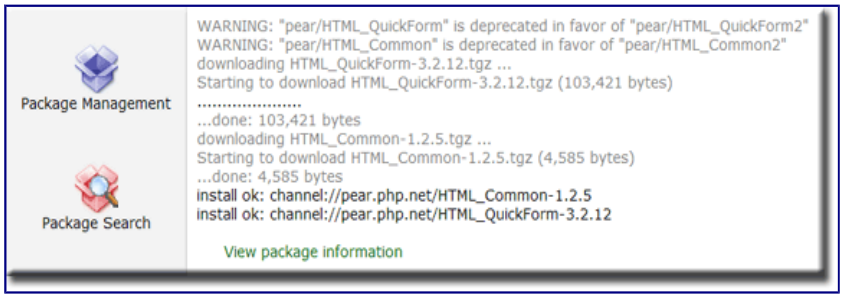
Downloading modules

Downloading a known package (example HTML_QuickForm)

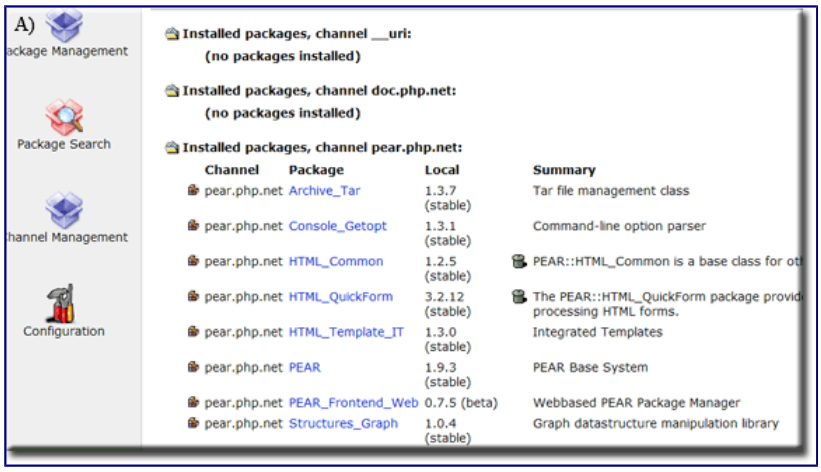
- A) Click on **Package Management** - Opens management page.
- Scroll down page to **Quick-install a package**
- Enter package name **HTML_QuickForm**
- **Click Install**



- After a short time the package is installed.
- Note the Warnings! We know this is an older package but want to use it anyway.



- A) Click on **Package Management** - Opens management page.
- New Installed packages are listed.



Although we intended to only download and install a single package, two were installed. The package HTML_QuickForm will not function without package HTML_Common; it depends on this being in place. In other words, when installing a package, all dependencies that do not exist are also installed.

Form - Test Script

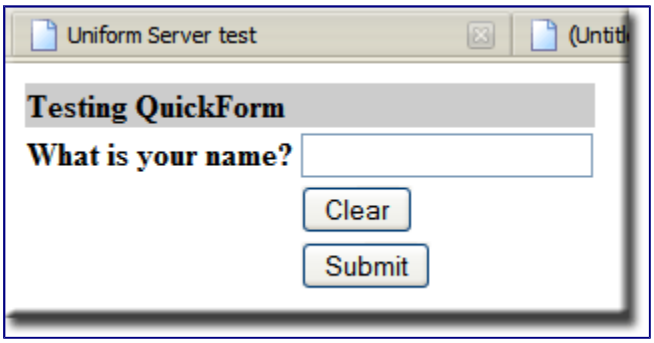
Create a new text file named **form.php** with the following content and save to folder UniServer\www

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
 <html>
 <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Pear HTML_QuickForm test</title>
 </head>
 <body>
 <?php
    require_once "HTML/QuickForm.php";

    $form = new HTML_QuickForm('frmTest', 'get');
    $form->addElement('header', 'MyHeader', 'Testing QuickForm');
    $form->addElement('text', 'MyTextBox', 'What is your name?');
    $form->addElement('reset', 'btnClear', 'Clear');
    $form->addElement('submit', 'btnSubmit', 'Submit');
    $form->display();

 ?>
 </body>
 </html>
```

Uniform Server – Coral

***Test*:**

- Start The Uniform Server
- Type the following into a browser: http://localhost/form.php
- Expected result is shown on right.



PHP - PEAR Manual Install

This page covers an alternative to automatically installing PEAR using a script!

Using a script to automatically install PEAR makes the process very easy. However there is a drawback that this process masks what is going on behind the scenes. Manually installing PEAR provides greater control, flexibility and exposes this hidden detail. The following covers manually installing PEAR, along with one of its components (classes) for testing.

Contents

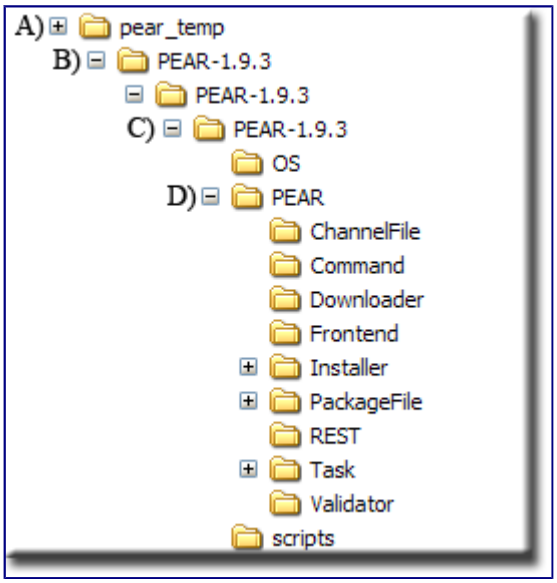
Install PEAR core package

The first requirement is to install the PEAR core package. This core package is used by all other packages to extend their functionality.

Note: All PEAR files are distributed using double compression (Unix standard). These have a file extension of tgz. To uncompress these files you can install the file archiver 7-Zip. Alternatively, you can use the portable version from Portable Apps.

Download and Extract PEAR core package

- A) Create a new folder. It can be any name, for example **pear_temp**
- Download the latest PEAR core package (PEAR-1.9.4.tgz - see top right "For manual installation only") - Save to folder **pear_temp**
- A) Start 7-Zip and navigate to folder **pear_temp**.
- A) Highlight file PEAR-1.9.4.tgz (left mouse click), click Extract (top menu button) and click OK in pop-up
- B) A new folder (PEAR-1.9.4) is created. Navigate into this folder.
- B) Highlight file PEAR-1.9.4.tar (left mouse click), click Extract and click OK
- You can now close 7-Zip.



That completes file extraction. You should have a folder structure as shown on the right.

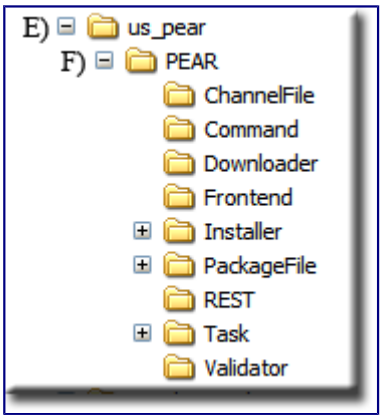
- C) Using explorer, navigate into folder PEAR-1.9.4 (last one in tree with this name).
- From this folder, copy files **PEAR.php** and **PEAR5.php** to folder PEAR D)

The PEAR core package is now ready to install on The Uniform Server.

Install core package on The Uniform Server

The Uniform Server has been pre-configured to run PEAR from a specific location UniServer\home\us_pear. This folder contains files appropriate for automatic installation; with the exception of file .htaccess its contents requires deleting. Install PEAR as follows:

- E) Navigate into folder UniServer\home**us_pear**.
  With the exception of file .htaccess, delete its content.
- D) Copy folder pear_temp\PEAR-1.9.4\PEAR-1.9.4\PEAR-1.9.4**PEAR**
  E) to folder UniServer\home\us_pear
- Folder structure as show on right



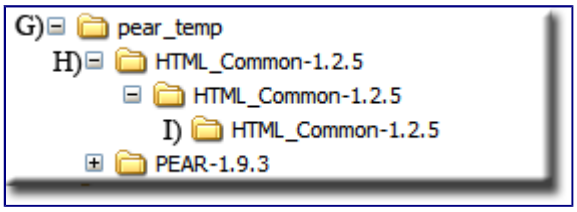
Installing PEAR packages

Installing PEAR packages is relatively easy. Download and extract the package and install it to the PEAR folder. To confirm our PEAR installation is working and to demonstrate how to install packages we will be using a package named QuickForm.

A Quick tour of PEAR repository

- First port of call is List Packages All packages have been grouped into sections. The number to the right of each section title is the total packages in this group.
- We are interested in the QuickForm package which comes under section HTML This page contains a list of all packages in this section.
- Scrolling down the page you will see enteries HTML_QuickForm, HTML_QuickForm2, HTML_QuickForm_advmultiselect etc. To the right is displayed the current status.
- Scroll to package HTML_QuickForm and click link.
- Scroll down the page and you will find an important section named **Dependencies**. For HTML_QuickForm you will see it requires HTML_Common. We do require this because our current installation contains no additional installed packages. The dependencies section provides a quick link; clicking the link HTML_Common will take you to the download page.
- At the top of each package page are the following links, Main, **Download,** **Documentation**, Bugs, Trackbacks. Generally you can find examples of how to use a package in the documentation. Alternatively, you can search the Internet. The download link opens a download page where you can download files for manual installation.

Installing Dependencies

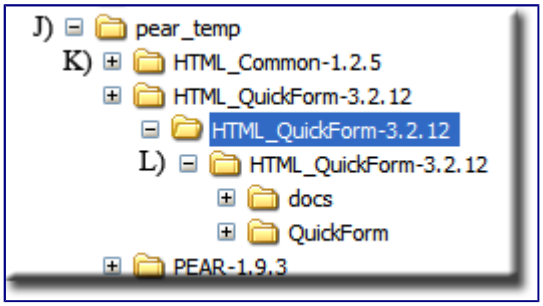


- G) Create a new folder or use existing one; for example **pear_temp**
- Download latest HTML_Common package (HTML_Common-1.2.5.tgz top right manual installation only) - Save to folder **pear_temp**
- G) Start 7-Zip and navigate to folder **pear_temp**.
- G) Highlight file HTML_Common-1.2.5.tgz (left mouse click), click Extract (top menu button) and click OK in the pop-up
- H) A new folder (HTML_Common-1.2.5) is created. Navigate into this folder.
- H) Highlight file HTML_Common-1.2.5.tar (left mouse click), click Extract and click OK
- You can now close 7-Zip.
- I) That completes file extraction. Installation requires installing a single file contained in folder HTML_Common-1.2.5

Where to install this file? Each distribution contains a file package.xml. Inside this file look for baseinstalldir which tells you the folder to install to. For this package it is HTML (baseinstalldir="HTML")

- F) Using explorer, navigate into folder UniServer\home\us_pear\PEAR
- Create a new folder HTML
- Using explorer navigate into folder HTML_Common-1.2.5 (last one in tree with this name).
- From this folder, copy file Common.php to folder UniServer\home\us_pear\PEAR\HTML

Installing PEAR Package - QuickForm



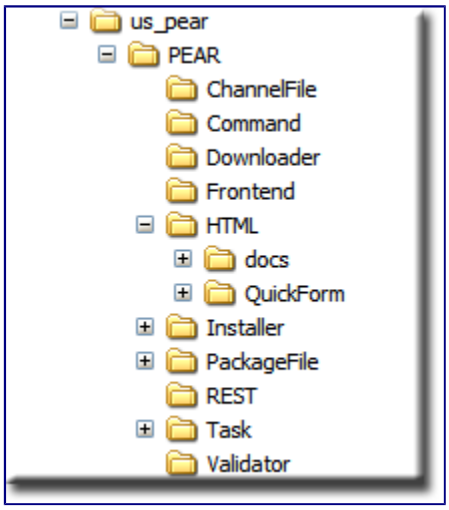
- J) Create a new folder or use existing one, for example **pear_temp**
- Download latest QuickForm package (HTML_QuickForm-3.2.12.tgz top right manual installation only) - Save to folder **pear_temp**
- J) Start 7-Zip and navigate to folder pear_temp.
- J) Heighlight file HTML_QuickForm-3.2.12.tgz (left mouse click), click Extract (top menu button) and click OK in pop-up
- K) A new folder (HTML_QuickForm-3.2.12) is created. Navigate into this folder.

- K) Heighlight file HTML_QuickForm-3.2.12.tar (left mouse click), click Extract and click OK
- You can now close 7-Zip.
- L) That completes file extraction. Installation requires installing content of folder HTML_QuickForm-3.2.12

Where to install this file? Each distribution contains a file package.xml inside this file look for baseinstalldir which tell you the folder to install to. For this package it is HTML (baseinstalldir="HTML" name="/")

- We have already create the base folder HTML in folder UniServer\home\us_pear\PEAR
- L) Using explorer navigate into folder HTML_QuickForm-3.2.12 (last one in tree with this name).
- From this folder copy its content to folder UniServer\home\us_pear\PEAR\HTML

The Uniform Server's PEAR folder will now look like that shown on the right.



Form - Test Script

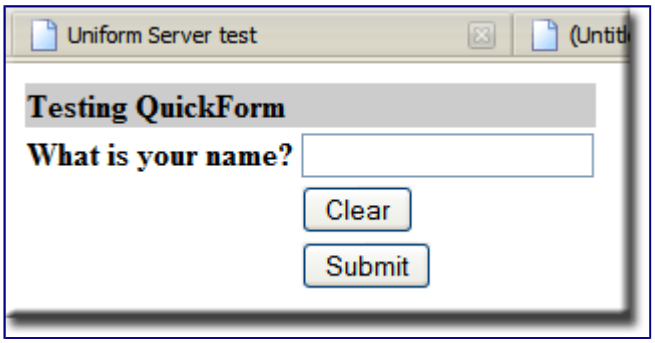
Create a new text file named **form.php** with the following content and save to folder UniServer**www**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   <title>Pear HTML_QuickForm test</title>
</head>
<body>
<?php
   require_once "HTML/QuickForm.php";

   $form = new HTML_QuickForm('frmTest', 'get');
   $form->addElement('header', 'MyHeader', 'Testing QuickForm');
   $form->addElement('text', 'MyTextBox', 'What is your name?');
   $form->addElement('reset', 'btnClear', 'Clear');
   $form->addElement('submit', 'btnSubmit', 'Submit');
   $form->display();

?>
```

```
</body>
</html>
```

Test:

- Start The Uniform Server
- Type follwong into browser: http://localhost/form.php
- Expected result is shown on right.



PEAR Alternate location

Although The Uniform Server uses a specific PEAR location, you can install to any location that suits your requirements. For example, you can copy PEAR folder to folder UniServer\usr\local**php** or perform a manual installation directely to this folder. Adding packages to the PEAR folder has been covered in the above sections.

Change include path for PEAR

The include path in PHP's configuration file php.ini requires changing to new PEAR folder location. There are three files to edit as follows:

- UniServer\usr\local\php**php.ini**
- UniServer\usr\local\php**php.ini_development**
- UniServer\usr\local\php**php.ini_production**

In each of the above files locate the following line:

```
include_path = ".;C:/UniServer/home/us_pear/PEAR"
```

And change to:

```
include_path = ".;UniServer/usr/local/php/PEAR"
```

Note 1:The path C:/UniServer may be different depending on where you extracted The Uniform Server, so change the paths accordingly.

Note 2: PHP always uses configuration file php.ini; the content of this file is selectable by a user. For example, when a user switches to development mode, contents of file php.ini_development are copied to php.ini

PHP - APC

The APC administrator interface is a PHP script named **apc.php** located in folder UniServer\home\us_extra. It can be accessed using UniController

Server Configuration > PHP > PHP Accelerators

Alternatively, you can type the following into your browser: http://localhost/us_extra/apc.php

This page covers the administrator interface menu bar and fine-tuning the cache.

Contents

APC administrator menu

The APC administrator interface has the following menu options allowing you to view usage or inspect cached variables.

Refresh Data
This captures the current cache status and updates the statistics.

View Host Stats
This option allows you to view the current cache status, general cache information, and usage and statistics on hits and misses. Information displayed covers both the system cache (which handles opcodes) and the user cache (which handles user variables). Various other statistics are displayed, such as the number of cache requests per second, hit rate and miss rate.

This information allows you identify areas that are under-optimized. For example, the cache full count value indicates how often the cache has filled up. A high number indicates high cache churn. To reduce this, assign more memory to the cache.

System Cache Entries
The System Cache Entries menu lists the PHP scripts that are currently being cached, together with their filename, size and number of hits. APC automatically caches each script's opcodes.

Note: You can clear the opcode cache at any time with the **Clear cache** button top right of the page.

User Cache Entries
The User Cache Entries menu lists user variables that have been stored in the cache, together with their identifier, size, and creation and modification times. You can select any of these entries to look inside the cache entry and see what it contains. User cache entries are not automatically created by APC. They are the result of specific instructions in a PHP script.

Note: You can clear the user cache at any time with the **Clear cache** button top right of the page.

Tuning the cache

The Uniform Server default **apc.shm_size** memory size is **64M**, which should be adequate for most small applications. The setting is application specific. It may require adjusting to avoid

fragmentation and cache churning. To determine if changes are required, open the APC control panel and check the following:

- Under **File Cache Information**, check Cache Full Count. If it's greater than zero, the cache is filling up and churning. This is because there is not enough memory allocated. Resolve this by increasing the memory allocation (apc.shm_size).
- Under **Detailed Memory Usage and Fragmentation** the Fragmentation value should be 0% (this may occasionally vary). A non-zero value is probably due to cache churning. Solve this the same as above.

Fragmentation occurs when cached items expire and new items fill their vacated memory space. The new item maybe slightly smaller than the old item and the remaining memory space may be too small for a new cache item. This unused memory space is referred to as fragmentation.

To avoid fragmentation, ensure you run your servers during high traffic conditions and note the value for **Memory Usage**. Double this value and add another Meg to it. Update the PHP configuration files with this new value for **apc.shm_size**

Additional Information

Read more about configuration directives at http://in3.php.net/manual/en/apc.configuration.php

MSMTP - Introduction

The majority of SMTP clients use the Windows registry. These are not suitable for portability. The Uniform Server uses the open source msmtp client. It's flexible, relative easy to set-up, and more importantly it's portable. This page covers the configuration sub-menu which allows you to configure and test the **msmtp** client.

Contents

Features

- MSMTP Integrated client.
- Pre-configured configuration template file.
- Controller to set default account - quick switching between accounts.
- Controller includes simple e-mail form for testing account configuration.

Overview

MSMTP has been pre-configured and enabled in the PHP configuration files. MSMTP itself uses a separate configuration file which has been pre-configured with example accounts. Before you can use PHP e-mail functions you need to set-up at least one account. Generally this account is your ISP email account, but for portability, consider setting up a free email account such as Hotmail or Gmail.

Only one account (the default account) is active at any one time. The Uniform Server allows you to easily switch between the accounts you have configured and set one of these as the default account. After selecting an account, you can test it using the builtin email tester.

Edit MSMTP Configuration

MSMTP is configured via the msmtprc.ini file. This has been pre-configured with three account skeletons (MyISP, Hotmail and Gmail). You can add as many accounts as you like, but each must have a unique account name.

Open this file either by using UniController or by directly using a text editor.

Server Configuration > MSMTP > Edit MSMTP Configuration

There are three pre-configured account skeletons (MyISP, Hotmail and Gmail). You can add as many as you like, but each must have a unique account name.

The pre-configured accounts are templates. You must change the following for at least one account and assign it as the default.

1) MyISP

- Substitute smpt.tiscali.co.uk with your ISP's smpt server
- Substitute

```
# 1 === Your ISP account ========================

account MyISP
host smpt.tiscali.co.uk
from john.doe@tiscali.co.uk
auth off

# 2 === A freemail account at Hotmail ===========

account Hotmail
tls on
tls_certcheck off
host smtp.live.com
from john.doe123@hotmail.co.uk
auth on
user john.doe123@hotmail.co.uk
password fred123

# 3 === A freemail account at Google ============
```

john.doe@tiscali.co.uk with your real email address

2) Hotmail

- Substitute john.doe123@hotmail.co.uk with your login email address
- Substitute fred123 with your login password

Note: To relay via Hotmail you will need to regularly sign into your account. [??]

3) Gmail

- Substitute john.doe777@gmail.com with your login email address
- Substitute fred999 with your login password

Default Account

- Change "account default : **Hotmail**" with the account name you want to use

```
account Gmail
tls on
port 587
tls_certcheck off
host smtp.gmail.com
from john.doe777@gmail.com
auth on
user john.doe777@gmail.com
password fred999

#==== Set a default account ====================

account default : Hotmail
```

Note 1: PHP will use whichever account you have assigned as default, which is currently set to Hotmail.
Note 2: With more than one account configured you can easily select a default.
Note 3: The drop down menu for selecting a default account lists all accounts in the configuration, so you probably will want to delete any accounts you have not configured.

Default Account

You can manually edit the configuration file **msmtprc.ini** and set the default account, or alternatively use UniController's sub-menu:

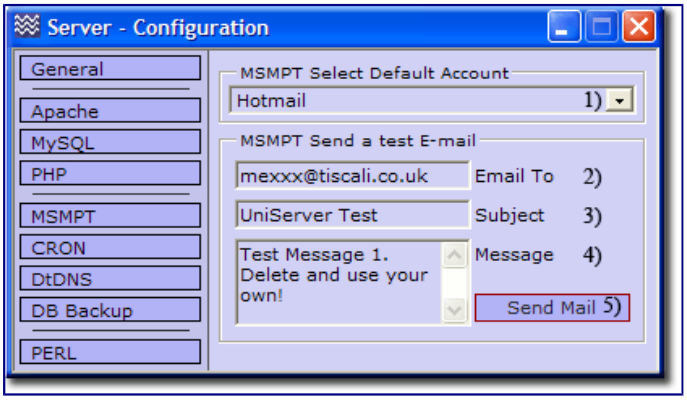
Server Configuration > MSMTP > Default Account

Uniform Server – Coral

This is a dual-purpose menu allowing you to set a default account and send a test email using this account.

Set Default Account

- Click the drop-down menu 1). This displays all accounts contained in the configuration file.
- To set a default account, click an account name.



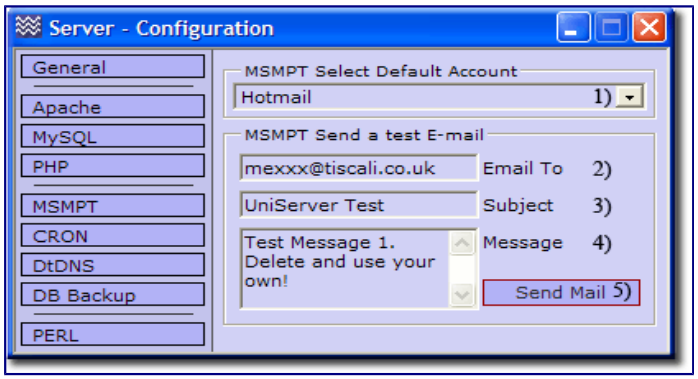
Send Test E-Mail

Server Configuration > MSMTP > Send Test E-Mail

This is a dual-purpose menu allowing you to set a default account and send a test email using this account.

Send Test Email

- Enter recipient's email address 2)
- Optionally enter a subject 3) or use default.
- Optionally enter a message 4) or use default.
- Click **Send Email** button 5)



View Log

Server Configuration > MSMTP > View Log

This menu option provides a convenient short cut allowing you to view the MSMTP log file. If you are experiencing problems sending e-mails, this should be your first port of call;

Feb 04 14:07:10 host=smtp.gmail.com tls=on auth=on user=xxx@gmail.com from=xxx@gmail.com recipients=my_mail@operamail.com mailsize=117 smtpstatus=250 smtpmsg='250 2.0.0 OK 1265292428 7sm2124401eyg.17' exitcode=EX_OK

Sep 22 14:08:53 host=smpt.tiscali.co.uk tls=off auth=off from=xxx@tiscali.co.uk recipients=my_mail@operamail.com mailsize=108 smtpstatus=250 smtpmsg='250 ok: Message 497320898 accepted' exitcode=EX_OK

Jan 24 11:12:14 host=smtp.live.com tls=on auth=on user=xxx@hotmail.co.uk from=xxx@hotmail.co.uk recipients=yyy@tiscali.co.uk mailsize=100 smtpstatus=250 smtpmsg='250 2.6.0 <BLU0-SMTP9@BLU0-

it may shed light on a particular issue.

An example of the log content is shown on right. There are three lines which have been split to fit this web page.

Log file:
UniServer\msmtp**msmtp.log**

```
SMTP9.blu0.hotmail.com>
Queued mail for delivery' exitcode=EX_OK
```

MSMTP - Detail

The majority of SMTP clients use the Windows registry. These are not suitable for portability. The Uniform Server uses the open source msmtp client. It's flexible, relative easy to set-up, and more importantly it's portable. This page covers the configuration sub-menu which allows you to configure and test the **msmtp** client.

While msmtp can be configured to use your ISP's SMTP server, that again restricts portability. The answer is to create a free account such as Google Mail (**gmail**) or **Hotmail** and configure msmtp to use that.

This combination allows any PHP scripts to send e-mail to your account transparently. MSMTP has been integrated into The Uniform Server. All you need to do is configure the account as explained below.

Contents

Background

SMTP (simple mail transport protocol) was originally designed to be an open relay where an SMTP server would accept any e-mail for forwarding. This quickly became abused by spammers. In retaliation, ISP's restricted open relaying. This means you cannot use the PHP function to directly send e-mail to a user. You either require your own mail server with all the complication that is associated with it, or you use your ISP's SMTP server. In either situation you are restricted to a local server on a dedicated line.

Free e-mail accounts such as Google Mail remove these chains by allowing you to relay through their servers. However you must login to their servers before this privilege is granted.

Upgrade

This section is useful for users who wish to upgrade when a newer version of MSMTP is released.

Download the latest version (currently msmtp-1.4.24-w32.zip) from
http://sourceforge.net/projects/msmtp/files/

- Unzip to any folder.
- Copy file: msmtp-1.4.24-w32\msmtp-1.4.24-w32**msmtp.exe**
- To folder: UniServer**msmtp**

That is all there is to an upgrade.

How MSMTP client was integrated

If you are interested in how msmtp integrates into The Uniform Server's architecture, read on.

There are three requirements for integration:

- Inform PHP where to find the msmtp executable.
- Inform msmtp where to find its configuration file.
- Finally inform msmtp where its log file is to be located.

msmtp configuration file:	log file location:
Path and name UniServer\msmtp\msmtprc.ini Note: File name could have been different	The configuration file msmtprc.ini contains the log file path. This path is specified as an absolute path. For example: "C:/some_folder/UniServer/msmtp/msmtp.log" Note: When the servers are moved, this path is automatically updated.

php.ini configuration

A single line placed in php.ini configuration file resolves the first two requirements. The line has the following format:

```
sendmail_path = "C:/some_folder/UniServer/msmtp/msmtp.exe --file=C:/some_folder/UniServer/msmtp/msmtprc.ini  -t"
```

Note: Absolute paths with forward slashes are used. The first part instructs PHP where to find the msmtp executable, and the second part instructs msmtp where to find its configuration file.

Extract from PHP configuration file php.ini

```
[mail function]
; For Win32 only.
; http://php.net/smtp
;SMTP = localhost
; http://php.net/smtp-port
;smtp_port = 25

; For Win32 only.
; http://php.net/sendmail-from
;sendmail_from = me@localhost.com

; For Unix only.  You may supply arguments as well (default: "sendmail -t -i").
;sendmail_path = "/usr/bin/sendmail.exe -t"

sendmail_path = "C:/UniServer/msmtp/msmtp.exe --file=C:/UniServer/msmtp/msmtprc.ini  -t"
```

CRON - Introduction

Cron originated in the Unix environment. It is a job scheduler, allowing scripts to be run automatically at a certain time or date. The name was based on the Greek word for time, "chronos" .

The Uniform Server’s portable Cron plugin has been integrated into 8.0-Coral. Cron's configuration file has been pre-populated with various scripts to be run. These are essentially four-line templates that require enabling to run the appropriate applications from the Cron script.

The Uniform Server provides a simple user control interface to start and stop Cron as a standard program. Alternatively, you can install Cron as a service.

Contents

Features

- Configuration file pre-configured for common applications.
- Run Cron as a standard program
- Run Cron as a service
- Run scripts with the following file extensions: .bat, .vbs and .php
- Pre-configured to run DtDNS, DBbackup and Drupal cron jobs

Configuration file

Each script to be run requires a configuration entry in the configuration file cron.ini. Open this file in the default editor using UniController as follows:

UniController: Server Configuration > Cron > Edit Cron Configuration

- Each script to be run requires a configuration block. Two examples are shown at right.
- Comand-line scripts (.bat, .php, .vbs) require an absolute path to be specified.
- Web-applications require a URL path. This URL path is what you would type into a browser to run that script.
- Each script running a Command-line or Web-application is defined in a separate block with the following format:

```
[db_backup]
start  = 2009-09-21 2:56:52
period = hourly
path   =
C:\UniServer\uni_con\db_backup\Run_db_backup.bat
ref =

[moodle]
start  = 2009-09-21 2:56:52
period = hourly
path   = http://localhost/moodle/admin/cron.php
ref =
```

Configuration block format

[dtdns]	--	Each block starts with a unique name enclosed in square brackets. Note that no spaces are allowed.
start =	--	Initial start time with the following format: Y-M-D H:M:S - Note: 24 hour clock.
period =	--	How often to run the script from the above referenced start time Values: hourly, daily, weekly, monthly or numeric in seconds
path =	--	a) For a web application, the full URL of the script. e.g. http://localhost/drupal/cron.php b) Command-line (CLI) uses an absolute path with back-slashes e.g. C:\UniServer\uni_con\cron\test_cron_1.bat
ref =	--	A timestamp updated by cron script. Set initial value to blank

Note 1: Cron automatically updates ref. It initially adds period as set above to Start time. For subsequent runs, it sets ref to current time + period.

Note 2: To change start time, first set a new value for start and delete the ref number, then save the file. The script will run at the new date and time set and thereafter at the rate you defined for period.

Note 3: To use pre-configured CLI blocks for dtdns and db_backup, uncomment to enable.

Note 4: To use pre-configured web blocks for drupal and moodle, uncomment to enable.

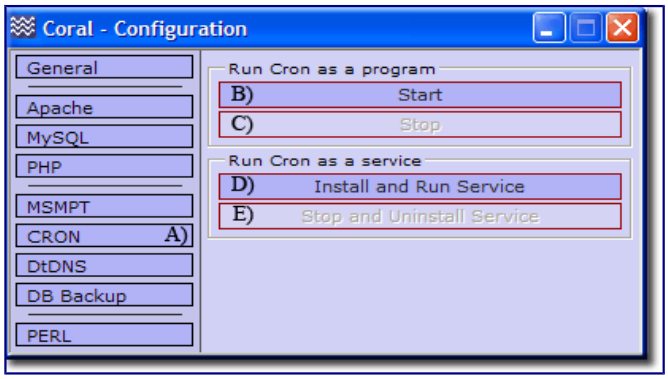
Note 5: Command-line paths to scripts contained in sub-folders below UniServer are automatically updated (portable). Paths outside UniServer require a manual change.

Cron Controller

Using Cron controller, portable Cron is run either as a standard program or as a service. Open the Cron controller as follows:

UniController: Server Configuration > Cron > Cron Controller

The Cron Controller consists of four buttons (B,C,D and E) as shown on right



- Initially, both program B) and service D) buttons are enabled. When Cron is running, one of these buttons is disabled; for example, if it is running as a service the program button is disabled.

- **Run as a program:** To run Cron, click the start button B). With Cron running, this button is disabled and the Stop button C) becomes enabled, allowing you to stop Cron.

- **Run as a service:** To install and run Cron, click the start button D). With Cron running as a service, this button is disabled and the Stop and Uninstall Service button E) becomes enabled, allowing you to stop and uninstall Cron service.

Logging

Cron logs the following information:

- Time Cron was started
- Start time and path of a script that is run.
- Time Cron was stopped. Note: if this is missing from the log, it means the servers were not shutdown before the PC was turned off

Note: Cron logging is enabled by default. This can be disabled; see xxx for details.

Summary

The above has shown how easy Cron is to use and configure.

Note: If you enable either or both builtin CLI scripts (DtDNS updater, db_backup), remember to properly set their configuration files.

CRON - Configuration detail

This page adds more detail to the basic Cron page. It covers additional user configuration and paths to appropriate files.

Contents

User Cron script-configuration

Generally there is no need to edit the Cron script! However there is a user configuration section where you can disable logging or change the basic tick rate.

Open file: UniServer\uni_con\cron**run_cron.vbs**
Locate this section:

```
'*** User configuration ****************************************************

 logging   = true          'true = Enable logging false = disable logging
 cron_time = 60             'Set cron time (tick) in seconds. Default 60=1 Min
 cron_loop = cron_time      'Set equal allows immediate first time run

'************************************************** End User configuration ***
```

Changing cron_time also changes the resolution time. Sixty seconds is a good compromise and ideal for testing scripts and running a production server.

Test Scripts

Three test files have been included and pre-configured in Cron's configuration file. Each is a simple command-line script creating a text file when run.

- UniServer\uni_con\cron\cron_test**test_cron_1.bat** File created:
  UniServer\uni_con\cron\cron_test**test_cron_1_result.txt**
- UniServer\uni_con\cron\cron_test**test_cron_2.vbs** File created:
  UniServer\uni_con\cron\cron_test**test_cron_2_result.txt**
- UniServer\uni_con\cron\cron_test**test_cron_3.php** File created:
  UniServer\uni_con\cron\cron_test**test_cron_3_result.txt**

CRON - Portable Design

The Uniform Server has integrated Nick Rozanski's srvstart utility into Coral’s control architecture, allowing portable Cron to run as a service. For anyone wanting to modify it or understand how it has been implemented, the following describes its operation and design.

Contents

Portable Cron

Portable Cron consists of an infinite program loop. It periodically checks an external file to determine when to exit. This concept is important when running the script as a service.

Service files

Cron service files are contained in folder UniServer\uni_con**cron_service** and perform the following function:

Utility program files:

- logger.dll - Service utility
- srvstart.dll - Service utility program functions
- srvstart.exe - Service utility program
- srvstart.ini - Service utility configuration file

Individual service control:

- z_install.bat - Install a named service
  - start_automatic.vbs - Starts service and changes service run status to Automatic
- z_service_start.bat - Starts service independently of changing run status.
- z_service_stop.bat - Stop service
- z_uninstall_service.bat - Remove service

Combined service control:

- install_start_service.bat - Uses the above to install and start cron service
- stop_uninstall_service.bat - Uses the above to stop and remove service.

The following provides details for each control file.

File description and function

For clarity only pertinent information is included (files title block has been removed)

srvstart.ini

This file allows you to specify any number of services to run. Each service is specified in a block starting with a unique name enclosed in square brackets. The Uniform Server Cron requires a single block as shown below:

```
[US_CronS1]
startup=C:\UniServer\uni_con\cron\start_cron.bat
startup_dir=C:\UniServer\uni_con\cron
shutdown=C:\UniServer\uni_con\cron\stop_cron.bat
auto_restart=y
```

- **[US_CronS1]** specifies a unique service name **US_CronS1** When running multi-servers digit is incremented to retain uniquness.
- **startup** Full path to the script to run.

restart_interval=30

Service can run only .exe, .com or .bat files hence start_cron.bat This in turn runs run_cron.vbs (portable cron)

- **startup_dir** Initially look in this folder for the file to run
- **auto_restart** If portable Cron fails attempt to restart it after a delay of 30 seconds.
- **restart_interval** Define restart delay time, attempt to re-start script every 30 seconds

shutdown When the service control manger (SCM) receives a **net stop** message it runs the script (stop_cron.bat) defined by **shutdown**.

Script **stop_cron.bat** runs script **stop_cron.vbs** which sets cron status to **stop** in configuration file **config_tracker.ini** this signals portable Cron to stop. The batch file **start_cron.bat** then closes informing SCM service has stopped.

z_install.bat

The service utility installs a named service (US_CronS1) using the **install** command. It uses a configuration "**-c**" file defined with an absolute path to prevent any ambiguity. Configuration file used is srvstart.ini

```
rem ### working directory current folder
pushd %~dp0
srvstart.exe install US_CronS1 -c C:\UniServer\uni_con\cron_service\srvstart.ini
: pause

rem ### restore original working directory
popd
```

Note 1: The above installs the service as a manual service. After re-starting your PC you must manually start the service next script changes service state to automatic.

Note 2:

- pushd %~dp0 - Saves current working directory and change working directory to batch file location.
- popd - Restores original working directory

This allows a batch file to be run from any location without worrying about current working directory

Note 3: The pause command is used to pause a batch file. It allows any errors to be seen, waits for a user to press any key. It is used only for testing hence is disabled by commenting it out using a colon ":"

start_automatic.vbs

The service utility installs a service as a manual service this state requires changing to automatic allowing the service script to run when the PC is restarted. The following script starts the service and changes state to automatic.

```
Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")

Set colRunningServices = objWMIService.ExecQuery ("Select * from
Win32_Service")
 For Each objService in colRunningServices
   If objService.DisplayName = "US_CronS1" Then
    'If objService.State = "Running" Then
    '   MsgBox "Is Running"
    'End If
    If objService.State= "Stopped" Then
       errReturn = objService.StartService()              'Start service
       errReturn = objService.Change( , , , , "Automatic") 'Change state
    End If
   End If
 Next
```

Create a Windows Management Instrumentation (WMI)object.

Executes a query on the win32 service object

Checks for named service running. Starts it if not running.

Changes service state to Automatic

The above script is run after installing a service.

z_service_start.bat

This script is provided for testing. It starts a service that has been installed with a manual status.

```
rem ### working directory current folder
pushd %~dp0
net start US_CronS1
:pause
rem ### restore original working directory
popd
```

The script uses the standard **net start** command to start a named (US_CronS1) service.

z_service_stop.bat

This script stops a named service.

```
rem ### working directory current folder
pushd %~dp0
net stop US_CronS1
:pause
rem ### restore original working directory
popd
```

The script uses the standard **net stop** command to stop a named (US_CronS1) service.

z_uninstall_service.bat

The service utility uninstalls a named service (US_CronS1) using the uninstall command.

```
rem ### working directory current folder
pushd %~dp0
```

To use this script service must first be

```
srvstart.exe remove US_CronS1
:pause
rem ### restore original working directory
popd
```

stopped.

install_start_service.bat

This script combines above scripts (z_install.bat and start_automatic.vbs) into a single script, which is used by Coral's control interface.

```
rem ### working directory current folder
pushd %~dp0
:Install srvice
call z_install.bat
:Change to automatic and start service
call start_automatic.vbs
: pause
rem ### restore original working directory
popd
```

Call Runs and waits for a script to complete before continuing

stop_uninstall_service.bat

This script combines above scripts (z_service_stop.bat and z_uninstall_service.bat) into a single script, which is used by Coral's control interface.

```
rem ### working directory current folder
pushd %~dp0
:Stop service
call z_service_stop.bat
:Uninstall service
call z_uninstall_service.bat
: pause
rem ### restore original working directory
popd
```

Server relocation

Relocating the servers all absolute paths are re-written the following files are changed accordingly:

- srvstart.ini
- z_install.bat

Multi-Servers

Running multi-servers requires unique service names hence these files are changed accordingly:

- srvstart.ini
- z_install.bat
- start_automatic.vbs
- z_service_start.bat
- z_service_stop.bat
- z_uninstall_service.bat

CRON - SrvStart utility tutorial

This short tutorial covers using the SrvStart utility written by Nick Rozanski. It allows you to run a script as a service!

What is a Service?
A Windows Service is automatically started when your PC starts. Services automatically recover from program crashes and Standby or Hibernation modes. Installing software as a service requires Administrator permissions.

Complexity
An application program designed to run as a service contains code that interfaces to the Windows service manager. Running a script as a service requires additional code that allows it to access the Windows service manager. This code registers your script, allowing it to be run as a service. In addition it must provide responses to command requests from the service manager and have the ability to provide your script's status.

It's not a trivial exercise, but this complexity is hidden when using SrvStart.

Contents

Download SrvStart

Download SRVStart.exe from http://www.rozanski.org.uk/software

- Navigate to section SRVSTART.EXE
- Click on the image named "Executable"
- This starts the download.
- Save file srvstart_run.v110.zip to any folder.
- Extract contents of this file to where you saved it.

Install SRVStart

The folder srvstart_run.v110 contains the following files:

- logger.dll (28KB)
- srvstart.dll (140KB)
- srvstart.exe (size 36KB)
- svc.exe

We are only interested in files **srvstart.exe**, **srvstart.dll** and **logger.dll**
For this tutorial, create a new folder c:**service_test** and copy the above three files into it.

Note 1:
The documentation says to copy files srvstart.exe, srvstart.dll, logger.dll and msvcrt.dll to C:\WINDOWS\system32 (system path). For XP onwards msvcrt.dll is already installed on a system path. The other three files do not need to be installed on a system path; see below. We are going to run srvstart.exe from its installation folder. This allows it to be portable by not installing anything on a host PC (assumes running from memory stick).

Note 2:
Notwithstanding the above, when installing a service, it does write to the registry, so you must remember to stop and uninstall the service you create if it resides on an external drive or a memory stick.

Script to run

For this tutorial we require a script to run. Create a new text document named **my_script.vbs** with the following content:

```
Set oWS = WScript.CreateObject("WScript.Shell")
oWS.Run "%comspec% /c echo " & Chr(07), 0, True

Set oWS = WScript.CreateObject("WScript.Shell")
oWS.Run "%comspec% /c echo " & Chr(07), 0, True
```

This is an annoying VBScript which just produces beeps when it runs.

Double click on my_script.vbs and confirm that it produces two beeps.

Limitation

An application to be run as a service must have one of the following file extensions: .exe, .com or .bat. The script we want to run has an extension of .vbs so it cannot be run directly.

At first this looks very restrictive, however a batch file will run any application. To run our script as a service, create a batch file named my_script.bat with the following content:

```
start my_script.vbs
```

start Runs our vb script and closes. It does not wait for our vb script to finish.
Double click on my_script.bat and confirm it produces two beeps.

Note 1: Running my_script.bat a command window opens for a short time. When run as a service this is hidden.

Note 2: Removing the "start" keyword, our vb script runs and the command window remains open until vb script finishes. Again, when run as a service, this is hidden.

SrvStart configuration file

To create our application (script) service SrvStart requires a configuration file.

- Each service to be run is contained in a separate block.
- Every block starts with a service name enclosed in square brackets.
- Each block must contain a **startup** command, which specifies the program (application/script) to launch.

Create a file named **srvstart.ini** with the following content:

```
[z_test]
startup=C:\service_test\my_script.bat
startup_dir=C:\service_test
```

Service name is **z_test**
startup Full path to script
startup_dir Folder to start looking in

Control_batch_files

We are now ready to install and run our service. This can be done using a command prompt and issuing the appropriate commands. I prefer to put these commands into separate batch files as follows:

z_install.bat

Create a file named **z_install.bat** with the following content:

```
srvstart.exe install z_test -c C:\service_test\srvstart.ini
pause
```

- srvstart.exe - Run srvstart program
- install - Command to run is install

- z_test - Name of service to install
- -c Configuration file to use

z_uninstall_service.bat

Create a file named **z_uninstall_service.bat** with the following content:

```
srvstart.exe remove z_test
pause
```

- srvstart.exe - Run srvstart program
- remove - Command to run is install
- z_test - Name of service to uninstall

z_service_start.bat

Create a file named **z_service_start.bat** with the following content:

```
net start z_test
pause
```

- net - Command is used to manage services
- start - Start the named service
- z_test - Name of service to start

z_service_stop.bat

Create a file named **z_service_stop.bat** with the following content:

```
net stop z_test
pause
```

- net - Command is used to manage services
- stop - Stops the named service
- z_test - Name of service to start

Test - Perceived problems

Run **z_install.bat**	You will see "Successfully created non-desktop service 'z_test'"
Run **z_service_start.bat**	Confirm two beeps are produced. Note the following messages: • The z_test service is starting. • The z_test service could not be started. • The service did not report an error. • More help is available by typing NET HELPMSG 3534.

	With the confirmation of two beeps, our script was successfully run. The reason for **could not be started message**; the batch file closes immediately when run. Srvstart regularly checks the running status of an application it started, it immediately sees the script is no longer running and reports back to the service control that it could not be started. The service did not report an error because it's not an error.
Run **z_service_stop.bat**	Note the following messages • C:\service_test>net stop z_test • The z_test service is not started. • More help is available by typing NET HELPMSG 3521. The batch file is not running, hence the service is reported as not running; see above. The above is not a problem; it's the way we want to run our script.
Run **z_uninstall_service.bat**	Note the following message • Deletion of service 'z_test' succeeded Confirms correct operation

Note: The above results may be a little confusing. Rremember our VBScript does not run continuously; it's a run-once program. Its purpose is to familiarise yourself with the SrvStart program.

Service Status

With the service created, it will now appear in the Windows Services list and can be configured like any other Windows Service. The service which is installed has the following characteristics.

- Its short name and display name are both z_test.
- It is set to Manual startup.
- It starts using LocalSystem and has no dependencies.
- It cannot interact with the desktop.

Manual startup

The service start type specifies the service's behavior after reboot:

- Auto - Automatically start after a re-boot
- Manual - Need to manually start service once PC started
- Disabled - Deactivated

We want our script to start when the PC is powered up. Either change the Manual state to Auto

using service manger, or use a script to set the service state. Create a new file named **start_automatic.vbs** with the following content:

```
Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")

Set colRunningServices = objWMIService.ExecQuery ("Select * from Win32_Service")
 For Each objService in colRunningServices
  If objService.DisplayName = "z_test" Then
      errReturn = objService.Change( , , , , "Automatic")
  End If
 Next
```

The above script will change the service state to Automatic. After you have installed your service it only requires running once.

For testing, this script starts our service and changes it to Automatic.

```
Set objWMIService = GetObject("winmgmts:\\.\root\cimv2")

Set colRunningServices = objWMIService.ExecQuery ("Select * from Win32_Service")
 For Each objService in colRunningServices
  If objService.DisplayName = "z_test" Then
    If objService.State = "Running" Then
      MsgBox "Is Running"
    End If
    If objService.State= "Stopped" Then
      errReturn = objService.StartService()
      errReturn = objService.Change( , , , , "Automatic")
    End If
  End If
 Next
```

Simulating service program crash

Our batch file is ideal for simulating a service program crash and demonstrating recovery. The batch file only runs once (simulating a program crash), srvstat can be set to automatically restart the program using directive **auto_restart=y**, setting **restart_interval=seconds** will pause before starting.

This can be put to good use; it allows a one time run script to be periodically called. Modify the configuration file as shown below:

```
[z_test]
startup=C:\service_test\my_script.bat
startup_dir=C:\service_test
auto_restart=y
restart_interval=10
```

Our test script will now be run every 10 seconds

Note 1: auto_restart will restart the service program if it exits for any reason. If restart_interval is defined, then before restarting, SRVSTART will wait the specified seconds.

Note 2: auto_restart does not restart the service program if it is stopped by request (eg NET STOP or Control Panel|Services).

Note 3: auto_restart does not restart the service program if it thinks that Windows NT is shutting down. If you are irritated by services restarting during Windows NT shutdown, then increase the value of restart_interval to, say, a minute.

Conclusion

The above has shown how to run a simple VBScipt as a service. When The Uniform Server is installed as a permanent installation, the concepts outlined above can be used for kicking portable cron into life. Portable cron is intended to be run from a memory stick and still requires manually starting.

DtDNS - Introduction

The majority of domestic Internet connections are via a dynamic IP address, one that can change every time you connect to the Internet. A web server requires a static IP address. A free dynamic DNS service such as DtDNS provides accounts and tracking software to automatically update your IP address when it changes, creating the effect of a static IP address. The Uniform Server has a built-in utility which supports the DtDNS service. This page covers the configuration sub-menu, which allows you to configure and test the DtDNS service.

Contents

Edit DtDNS Accounts

Server Configuration > DtDNS > Edit DtDNS Accounts

For each account you wish to update at DtDNS add a block with the following format:

[account_1]	-	A unique name enclosed in square brackets. Can be any name you wish do not include spaces.
hostname =	-	Full host name as configured at DtDNS
password =	-	Your DtDNS account password

An extract from the configuration file (dtdns.ini) is shown on the right.

```
;###################################

;[account_1]
;hostname = books.effers.com
;password = fred123

;[account_2]
;hostname = books.effers.net
;password = fred123

;[account_3]
;hostname = books.dtdns.net
;password = fred123

;=== END Config ==================
```

- Add as many blocks as required.
- To use the examples uncomment (remove the ;) and substitute your data as appropriate.

Force DtDNS UpDATE

Server Configuration > DtDNS > Force DtDNS UpDATE

Runs the Uniform Server DtDNS update script. A conformation pop-up is produced "DtDNS was manually updated View log for details."

Enable Log

Logging is enabled by default you can turn logging off using this menu option. The DtDNS script is open in Notepad.

Server Configuration > DtDNS > Enable Log

To disable logging change the following line as shown:

```
'*** User configuration ****************************************

logging = true   'true  = Enable logging false = disable logging
test    = false  'true  = display IP address and host names
                 'false = no display
```

- logging = false

'************************************ End User configuration ***

Extract from script (dtdns_updater.vbs) is shown on the right.

View Log

Server Configuration > DtDNS > View Log

```
18/07/2011 22:42:18 ### Log Started ======================================
18/07/2011 22:42:25 Changed IP from: xx.x10.165.92 To xx.x10.160.158 books.dtdns.net
18/07/2011 22:42:31 Changed IP from: xx.x10.165.92 To xx.x10.160.158 books.effers.com
18/07/2011 22:42:36 Changed IP from: xx.x10.165.92 To xx.x10.160.158 fredxx.dtdns.net
18/07/2011 22:42:36 ### Log Ended ========================================
```

This menu option displays the DtDNS log file in Notepad. Each updater run produces an entry in the log file. An example of one run is shown on the right.

Enable in CRON

Every time you start your PC and reconnect to the Internet (given a new IP address) you need to manually run the DtDNS updater.

Remembering to manually run scripts like **DtDNS** becomes a chore. Uniform Server centralises running this type of script using portable Cron. Cron is started either manually (portable) or run as a service. It periodically executes a list of scripts; the DtDNS script is included and requires enabling as follows:

Server Configuration > DtDNS > Enable in CRON

```
;[dtdns]
;start  = 2011-04-7 13:20:00
;period = 600                ; 10 Mins as required by DtDNS
;path   = C:\UniServer\uni_con\dtdns_updater\Run_dtdns_updater.bat
;ref    =
```

Opens Cron

configuration file in Notepad

- Locate section shown on right

Enable pre-configured DtDNS section in Cron as follows:

```
;[dtdns]
start  = 2011-04-7 13:20:00
period = 600                ; 10 Mins as required by DtDNS
path   = C:\UniServer\uni_con\dtdns_updater\Run_dtdns_updater.bat
ref    =
```

- Un-comment four lines as shown on right.

Note: You can change the start time and period to suite your own requirements see Cron Configuration file for details.

DtDNS - Detail

How to simulate static IP address for running a web server when your service provider allocates IP address dynamically.

Contents

Overview

The majority of domestic Internet connections are via a dynamic IP address, one that can change every time you connect to the Internet. A web server requires a static IP address. A free dynamic DNS service such as DtDNS provides accounts and tracking software to automatically update your IP address when it changes, creating the effect of a static IP address. This is an alternative to purchasing a static IP address from your ISP (Internet service provider). Even with a static IP, you require an entry in a DNS server that converts your domain name into this IP address, allowing other users to access your server.

DtDNS provide accounts that allow you to obtain a domain name, and provide tracking software to automatically update your IP address when it changes. A big advantage of their free service is that they automatically enable wild cards by default. This means that names like <whatever>.fred.com will be mapped to your IP address.

The Uniform Server supports the DtDNS service by integrating a simple script to automatically update hostname IP address at DtDNS. The script is configurable, allowing you to update all five free hostnames. The script can optionally be run by Cron, making the whole task transparent.

Note: The script is portable, so putting the servers on a USB memory stick means you can have them on-line and accessible within ten minutes (the minimum DNS propagation time). Of course this assumes the PC host machine has been configured to allow Internet server traffic.

Edit DtDNS Accounts

Uniform Server DtDNS is configured via a configuration file **dtdns.ini**. This has been pre-configured with three accounts you can add as many accounts as you like, each must have a **unique** account name.

You can open this configuration file in one of two ways either using UniController or directely.

UniController: Open file as follows:

Server Configuration > DtDNS > Edit DtDNS Accounts

Directly: Open the following file in notepad or other suitable text editor:
UniServer\uni_con\dtdns_updater**dtdns.ini**

Configuration

For each account you wish to update at DtDNS add a block with the following format:

[**account_1**]	-	A unique name enclosed in square brackets. Can be any name you wish do not include spaces.
hostname =	-	Full host name as configured at DtDNS
password =	-	Your DtDNS account password

An extract from the configuration file (dtdns.ini) is shown on the right.

```
;###################################

;[account_1]
;hostname = books.effers.com
;password = fred123

;[account_2]
;hostname = books.effers.net
;password = fred123

;[account_3]
;hostname = books.dtdns.net
;password = fred123

;=== END Config ==================
```

- Add as many blocks as required.
- To use the examples uncomment (remove the ;) and substitute your data as appropriate.

Force DtDNS UpDATE

You can use either one of two methods to force a DtDNS update.

Method 1:

Easiest way to force a DtDNS update is to use UniController:

Server Configuration > DtDNS > Force DtDNS UpDATE

Method 2:

Alternately you can directely run the script using:

UniServer\uni_con\dtdns_updater**Run_dtdns_updater.bat**

Either method runs the Uniform Server DtDNS update script. A conformation pop-up is produced "DtDNS was manually updated View log for details."

Enable Log

Logging is enabled by default you can turn logging off using this menu option. The DtDNS script is open in Notepad.

Server Configuration > DtDNS > Enable Log

To disable logging change the following line as shown:

```
'*** User configuration ****************************************

logging = true   'true  = Enable logging false = disable logging
test    = false  'true  = display IP address and host names
                 'false = no display
```

- logging = false

'************************************ End User configuration ***

Extract from script (dtdns_updater.vbs) is shown on the right.

Note: Alternative is to directely edit the script UniServer\uni_con\dtdns_updater**dtdns_updater.vbs**

View Log

Server Configuration > DtDNS > View Log

This menu option displays the DtDNS log file in Notepad. Each updater run produces an entry in the log file. An example of one run is shown on the right.

```
18/07/2011 22:42:18 ### Log Started ======================================
18/07/2011 22:42:25 Changed IP from: xx.x10.165.92 To xx.x10.160.158 books.dtdns.net
18/07/2011 22:42:31 Changed IP from: xx.x10.165.92 To xx.x10.160.158 books.effers.com
18/07/2011 22:42:36 Changed IP from: xx.x10.165.92 To xx.x10.160.158 fredxx.dtdns.net
18/07/2011 22:42:36 ### Log Ended ========================================
```

Note: Log file location UniServer\uni_con\dtdns_updater**dtdns.log**

Enable in CRON

Every time you start your PC and reconnect to the Internet (given a new IP address) you need to manually run the DtDNS updater.

Remembering to manually run scripts like DtDNS becomes a chore. Uniform Server centralises running this type of script using portable Cron. Cron is started either manually (portable) or run as a service. It periodically executes a list of scripts; the DtDNS script is included and requires enabling as follows:

You can open Cron configuration file in one of two ways either using UniController or directely.

UniController: Open file as follows:

Server Configuration > DtDNS > Enable in CRON

Directly: Open the following file in notepad or other suitable text editor: UniServer\uni_con\cron**cron.ini**

Configuration:

- Locate section shown on right

```
;[dtdns]
;start  = 2011-04-7 13:20:00
;period = 600                ; 10 Mins as required by DtDNS
;path   =
C:\UniServer\uni_con\dtdns_updater\Run_dtdns_updater.bat
;ref    =
```

- Un-comment four lines as shown on right.

```
;[dtdns]
start  = 2011-04-7 13:20:00
period = 600                ; 10 Mins as required by DtDNS
path   =
C:\UniServer\uni_con\dtdns_updater\Run_dtdns_updater.bat
```

Enables pre-configured DtDNS section.

ref =

Create DtDNS Account

This page covers creating an account at DtDNS. Using this account allows you to configure and run the servers from a dynamically allocated IP address.

Contents

Creating a DtDNS account

Creating an account at DtDNS is extremely easy.

- Go to DtDNS home page.
- To the right of **log in** click **Create an Account**

There are two pages to fill in. Enter the following details:

Page 1:

1)	**Desired Username**	This name is used for logging into your **account**
2)	**Your Name**	Your real name (Internal use)
3)	**Desired Password**	A password for logging into your account
4)	**Confirm Password**	As above
5)	**E-mail Address**	A real active e-mail is required. An activation code is sent to this e-mail address
6)	**Confirm E-mail**	As above
7)	**Optional**	Skip the Optional Information section if you don't wish fill it in.
8)	**Two CAPCHA words**	Enter the two CAPCHA words (if these are difficult to read, click the new challenge button)

Note 1: After step 8) you will be redirected to page 2:
Note 2: A confirmation e-mail is sent to your e-mail account set in step 5)

Page 2:

9)	**Username**	Enter your Username as entered in 1).
10)	**Enter code**	Enter code sent to your email.

Note 3: The email sent to you contains a link returning you to page 2. After step 8) you can close the page and wait for this email.

Creating a Hostname overview

General information

- A DtDNS account is limited to **five free** hostnames.
- A hostname is concatenated with a **domain** name.
- I have listed available domains on the right.

- Your full URL, for example
  - If you chose **books** as hostname
  - And select from the drop down menu

Available Free Domain Names

3d-game.com	etowns.net
4irc.com	etowns.org
b0ne.com	flnet.org
bbsindex.com	gotgeeks.com
chatnook.com	scieron.com

effers.com

Will look like this: **http://books.effers.com**

darktech.org
deaftone.com
dtdns.net
effers.com
slyip.com
slyip.net
suroot.com

Wild Cards

After creating a hostname, wild cards are enabled by default. This allows you to use a URL such as: http://**www**.books.effers.com Note: The **www**. is the wild card part. This can be anything you like; for example:

- http://**www**.books.effers.com
- http://**www-internation**.books.effers.com
- http://**local**.books.effers.com
- http://**www.home**.books.effers.com

The host name **books.effers.com** resolves to your IP address. The **wild card** can be resolved on your server using Vhosts to create different sites.

Create Hostname

To create a new hostname at DtDNS

Login

- Go to Login page
- Enter Username: as set in step **1)**
- Enter Password: as set in step **3)**
- **My Services** page displayed
- Click on **Hostnames**. This opens the **Hostname Manager**

Create hostname

- Enter a desired **Hostname**
- From the drop down menu select a Domain
- Click **Add Hostname**
- Enter the two **CAPCHA** words (if these are difficult to read, click the new challenge button)
- Click **Add Hostname** button
- This is added to your list of host names

Manage existing hostnames

To manage existing hostnames, simply click the entry in the list; for example:

Hostname	Domain	IP Address	Type
uni23	dtdns.net	89.119.35.45	Active Dynamic
books	effers.com	87.129.30.174	Active Dynamic

Click a hostname link to open its corresponding configuration page. Here you can change various settings, including the IP address.

Note 1: After reconnecting to your service provider you need to login to DtDNS and update the IP address accordingly. Alternatively add the new account to The Uniform Server's builtin updater. See Edit DtDNS Accounts.

Note 2: With the updater configured you can enable Cron to perform automatic updates.

Summary

For a home web server the free account at DtDNS is ideal. At least it gives you an opportunity to test their service. Perhaps it will encourage you to purchase a real domain and use some of their paid services.

Db Backup

The Uniform Server's database backup allows for either manual or periodic backups. This page covers the configuration sub-menu, which allows you to configure the backup system and view backup logs.

Contents

Features

- Simple configuration file. Sets FIFO depth and enables or disables logging
- Force backup
- Enable in Cron - Auto backups
- Create database backup list. Add to list or delete from list
- Restore a selected database

Overview

The Uniform Server's database backup allows for either manual or periodic backups. Each database is extracted from the MySQL server in SQL format. The archive files consist of a database name with a unix time stamp appended. For example fred_1306863938_.sql. These are saved to folder UniServer\db_backup (which is created automatically).

Archive files can become large (they eat disk space). To minimise this, the archive system implements a FIFO (first in first out) deletion. The total number of files for each database is user specifiable (default 3). On reaching this value the eldest file is deleted to make way for a new archive file.

The above applies to both manual and automatic updates (Using Cron).

Edit DB Backup Config

The configuration file sets the FIFO depth to 3 and enables logging by default.

UniController: Open file as follows:

Server Configuration > DB Backup > Edit DB Backup Config

Alternatively open the file directly.
Directly: Open the following file in notepad or other suitable text editor:
UniServer\uni_con\db_backup**db_backup.ini**

Configuration

Fifo_depth = 3	This defines total number of archives to save. When the limit is reached, the eldest archive is deleted to make way for a new archive. FIFO stands for first-in, first-out (deleted).	;####################### [FIFO] Fifo_depth = 3 [LOG] Logging = true ;=== END Config ========
Logging = True	True enable logging False Disable logging	

Note: FIFO size is a compromise between database size, disk space and how often backups are required. Three is just a test value and should be changed to suit your particular requirements.

Force DB Backup

Using the menu option, you can force all selected databases to be backed up. It can also be done directly by running the batch file Run_db_backup.bat.
Either method assumes you have entered databases to the file list. See section Select DBs to backup.

UniController:

Server Configuration > DB Backup > Force DB Backup

Directely:

UniServer\uni_con\db_backup**Run_db_backup.bat**

Enable log

This menu option is identical to "Edit DB Backup Config". It allows you to enable or disable logging.

Server Configuration > DB Backup > Enable log

- Logging = **True** - Enable logging
- Logging = **False** - Disable logging

```
;#########################

[FIFO]
Fifo_depth = 3

[LOG]
Logging = true

;=== END Config ========
```

View Log

Server Configuration > DB Backup > View Log

The log file opens in Notepad; an extract from this file is shown on the right.

```
19/07/2011 22:18:07 ### Log Started ==========================
19/07/2011 22:18:07  File backed up - wordpress_1311081487_.sql
19/07/2011 22:18:07  File backed up - joomla_1311081487_.sql
19/07/2011 22:18:07 ### Log Ended ============================

19/07/2011 22:22:37 ### Log Started ==========================
19/07/2011 22:22:37  File backed up - wordpress_1311081757_.sql
19/07/2011 22:22:37  File backed up - joomla_1311081757_.sql
19/07/2011 22:22:37 ### Log Ended ============================
```

Log file displays date and time a database was backed up. Each file has a time stamp appended to its file name.

Note: The log file provides a quick way to check databases are being backed up. This is useful when setting up configuration files such as Cron.

```
19/07/2011 22:25:30 ### Log Started ==========================
19/07/2011 22:25:30  File backed up - wordpress_1311081930_.sql
19/07/2011 22:25:30  File backed up - joomla_1311081930_.sql
19/07/2011 22:25:30 ### Log Ended ============================
```

Note: The Log file location UniServer\uni_con\db_backup\db_backup.log

Enable In Cron

Every time you want to create a backup you need to manually run **Force DB Backup**.

Remembering to manually run scripts like **Force DB Backup** becomes a chore. The Uniform Server automates running this type of script using Cron, which is started either manually (portable) or run as a service. It periodically executes a list of scripts; the "Force DB Backup" script is included and requires enabling as follows:

You can open Cron configuration file in one of two ways either using UniController or directely.

UniController: To open the configuration file in Notepad

Server Configuration > DB Backup > Enable In Cron

Directly: Open the following file in notepad or other suitable text editor: UniServer\uni_con\cron\cron.ini

Configuration:

- Locate the section shown on right

```
;[db_backup]
;start  = 2009-09-21 2:56:52
;period = hourly
;path   = C:\UniServer\uni_con\db_backup\Run_db_backup.bat
;ref =
```

Enable the pre-configured **db_backup** section in Cron as follows:

```
;[db_backup]
start  = 2009-09-21 2:56:52
period = hourly
path   = C:\UniServer\uni_con\db_backup\Run_db_backup.bat
```

- Un-comment the four lines as shown on right.

```
ref =
```

Note: You can change the start time and period to suit your own requirements. See Cron Configuration file for details.

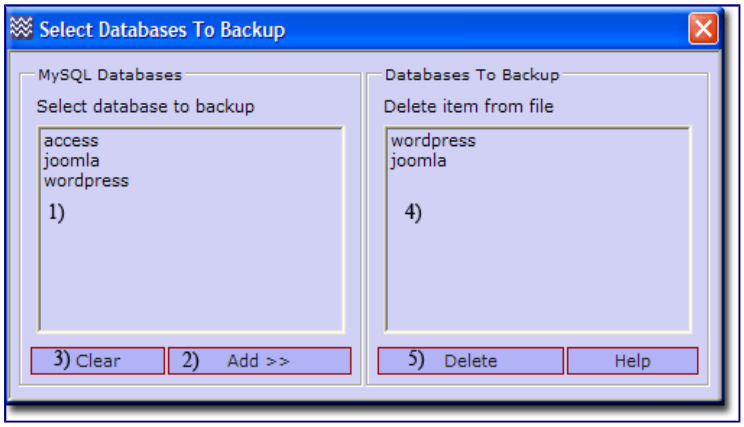
Select DBs to backup

In order to backup a database it must be added to the backup list contained in a file. This menu option provides an easy to use interface for adding or removing databases to and from this list. All users MySQL databases are listed on the left (1). The list of databases to be backed up is shown on the right (4) these are the current entries saved in a file.

Server Configuration > DB Backup > Select DBs to backup

Add to file

- From the databases listed on the left select a database to backup (1).
- Click the **Add** button (2). Selected database appears in right window (4) and added to file.



Note: Clear button (3) deselects a selection

Remove from file

- To delete an entry select a database from the list (4)
- Click **Delete** button (5). Removes entry from list and file.

Direct Alternative:

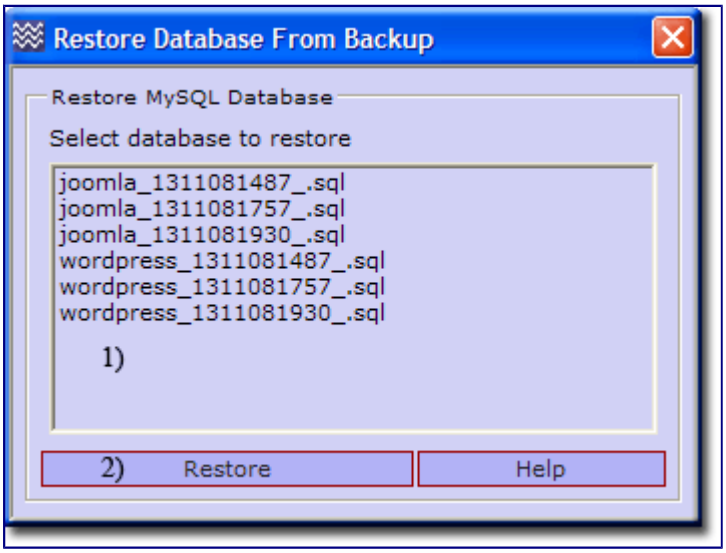
An alternative to using the above menu item is to directly edit the database list file. Open the following file in a text editor:

UniServer\uni_con\db_backup**dbs_to_backup.txt**

Add or delete database names as required.

Note: The following files should not be included in the backup list: "information_schema", "mysql", "performance_schema" or "phpmyadmin".

Restore DBs from backup



This menu option allows you to restore a database from a backup file.
All files backed up are displayed in section (1) each file listed has a time stamp appended.

Note: Smaller number indicates older backup.

Server Configuration > DB Backup > Restore DBs from backup

Restore database

- From the list (1)Select a database to restore.
- Click **Restore** button (2)

Note: Backup files are saved to folder UniServer**db_backup**.

Db Backup - Detail

The Uniform Server's database backup allows for either manual or periodic backups. Each database is extracted from the MySQL server in SQL format. The archive files consist of a database name with a unix time stamp appended. For example fred_1306863938_.sql. These are saved to folder UniServer\db_backup (created automatically).

Archive files can become large (they eat disk space). To minimise this, the archive folder implements a FIFO (first in first out) deletion. The total number of files for each database is user specifiable (default 3). On reaching this value the eldest file is deleted to make way for a new archive file.

The above applies to both manual and automatic updates (Using Cron).

Contents

Edit DB Backup Config

Uniform Server **DB Backup** is configured via a configuration file **db_backup.ini**. This has been pre-configured sets the FIFO depth to 3 and enables logging by default.

You can open this configuration file in one of two ways either using UniController or directely.

UniController: Open file as follows:

Server Configuration > DB Backup > Edit DB Backup Config

Alternatively open the file directly.
Directly: Open the following file in notepad or other suitable text editor:
UniServer\uni_con\db_backup**db_backup.ini**

Configuration

Fifo_depth = 3	This defines total number of archives to save. When the limit is reached, the eldest archive is deleted to make way for a new archive. FIFO stands for first-in, first-out (deleted).
Logging = True	True enable logging False Disable logging

```
;########################

[FIFO]
Fifo_depth = 3

[LOG]
Logging = true

;=== END Config ========
```

Note: FIFO size is a compromise between database size, disk space and how often backups are required. Three is just a test value and should be changed to suit your particular requirements.

Force DB Backup

You can force a database backup either using UniController or directly by running the batch file Run_db_backup.bat. Either method assumes you have entered databases to the file list see section to be backed up.

UniController:

Server Configuration > DB Backup > Force DB Backup

Directely:

UniServer\uni_con\db_backup**Run_db_backup.bat**

Enable log

This menu option is identical to "Edit DB Backup Config". It allows you to enable or disable logging.

Server Configuration > DB Backup > Enable log

- Logging = **True** - Enable logging
- Logging = **False** - Disable logging

```
;########################

[FIFO]
Fifo_depth = 3

[LOG]
Logging = true

;=== END Config ========
```

View Log

Server Configuration > DB Backup > View Log

The log file opens in Notepad; an extract from this file is shown on the right.

Log file displays date and time a database was backed up. Each file has a time stamp appended to its file name.

Note: The log file provides a quick way to check databases are being backed up. This is useful when setting up configuration files such as Cron.

```
19/07/2011 22:18:07 ### Log Started ==========================
19/07/2011 22:18:07  File backed up - wordpress_1311081487_.sql
19/07/2011 22:18:07  File backed up - joomla_1311081487_.sql
19/07/2011 22:18:07 ### Log Ended ============================

19/07/2011 22:22:37 ### Log Started ==========================
19/07/2011 22:22:37  File backed up - wordpress_1311081757_.sql
19/07/2011 22:22:37  File backed up - joomla_1311081757_.sql
19/07/2011 22:22:37 ### Log Ended ============================

19/07/2011 22:25:30 ### Log Started ==========================
19/07/2011 22:25:30  File backed up - wordpress_1311081930_.sql
19/07/2011 22:25:30  File backed up - joomla_1311081930_.sql
19/07/2011 22:25:30 ### Log Ended ============================
```

Note: The Log file location UniServer\uni_con\db_backup**db_backup.log**

Enable In Cron

Every time you want to create a backup you need to manually run **Force DB Backup**.

Remembering to manually run scripts like **Force DB Backup** becomes a chore. Uniform Server centralises running this type of script using portable Cron. Cron is started either manually (portable) or run as a service. It periodically executes a list of scripts; the "Force DB Backup" script is included and requires enabling as follows:

You can open Cron configuration file in one of two ways either using UniController or directely.

UniController: Open file as follows:

Server Configuration > DB Backup > Enable In Cron

Directly: Open the following file in notepad or other suitable text editor:
UniServer\uni_con\cron\cron.ini

Configuration:

- Locate the section shown on right

```
;[db_backup]
;start  = 2009-09-21 2:56:52
;period = hourly
;path   = C:\UniServer\uni_con\db_backup\Run_db_backup.bat
;ref =
```

Enable the pre-configured **db_backup** section in Cron as follows:

- Un-comment the four lines as shown on right.

```
;[db_backup]
start  = 2009-09-21 2:56:52
period = hourly
path   = C:\UniServer\uni_con\db_backup\Run_db_backup.bat
ref =
```

Note: You can change the start time and period to suit your own requirements. See Cron Configuration file for details.

Select DBs to backup

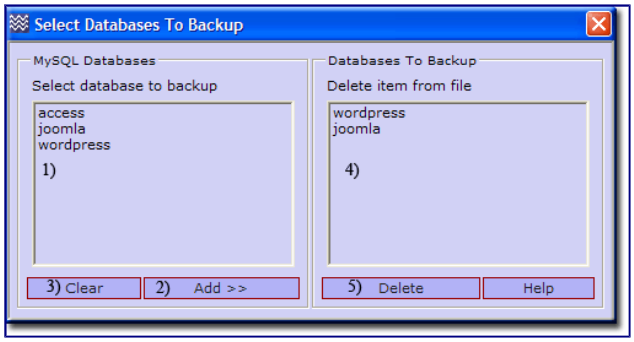
In order to backup a database it must be added to the backup list contained in a file. This menu option provides an easy to use interface for adding or removing databases to and from this list. All users MySQL databases are listed on the left (1). The list of databases to be backed up is shown on the right (4) these are the current entries saved in a file.

Server Configuration > DB Backup > Select DBs to backup

Add to file

- From the databases listed on the left select a database to backup (1).
- Click the **Add** button (2). Selected database appears in right window (4) and added to file.

Note: Clear button (3) deselects a selection



Remove from file

- To delete an entry select a database from the list (4)

- Click **Delete** button (5). Removes entry from list and file.

Direct Alternative:

An alternative to using the above menu item is to directly edit the database list file. Open the following file in a text editor:

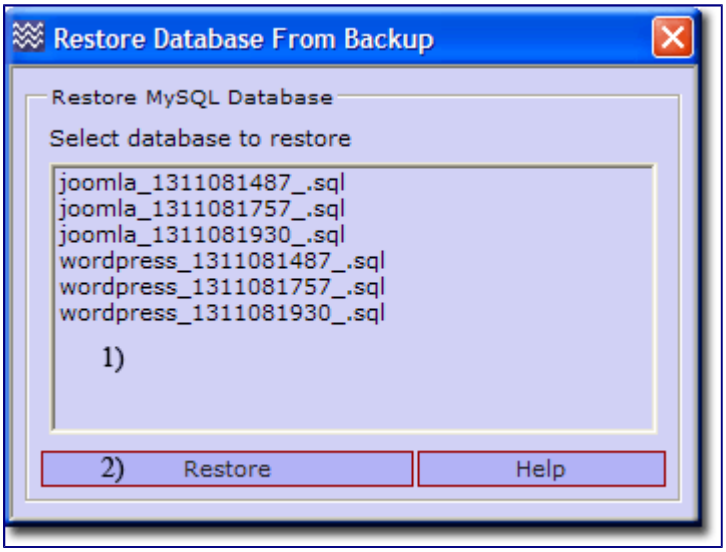
UniServer\uni_con\db_backup**dbs_to_backup.txt**

Add or delete database names as required.

Note: The following files should not be included in the backup list: "information_schema", "mysql", "performance_schema" or "phpmyadmin".

Restore DBs from backup

This menu option allows you to restore a database from a backup file.
All files backed up are displayed in section (1) each file listed has a time stamp appended.



Note: Smaller number indicates older backup.

Server Configuration > DB Backup > Restore DBs from backup

Restore database

- From the list (1)Select a database to restore.
- Click **Restore** button (2)

Note: Backup files are saved to folder UniServer**db_backup**.

Perl - Introduction Guide

Perl is a very powerful scripting language and is fully supported by The Uniform Server. Since the ActivePerl Community License precludes distribution with The Uniform Server, you will have to download it yourself. However, Perl is easy to install when using The Uniform Server's builtin script. For details, see Perl - Installing ActivePerl. This page covers UniController's menu option that displays the Perl control panel.

Contents

Overview

The Uniform Server automatically detects the presence of Perl and enables the functionality to support it in UniController. If Perl is not installed, the Perl control panel buttons are greyed out.

Auto tracking

Place your Perl scripts in folder UniServer**cgi-bin** and any sub-folders as appropriate. Each script requires a Shebang. This must be an absolute path to the Perl executable. For example:

#!C:/UniServer/usr/bin/perl.exe

Moving UniServer to another location automatically updates this Shebang. UniController, on detecting a server move, writes a new Shebang to all files with a **.pl** or **.cgi** extension. An automatic update is initiated **only** when the servers are **moved**.

***Important*:** When you place a third party Perl script in cgi-bin, you must either move the servers or **force a Shebang update** using UniController. Alternatively, you can manually edit each file.

UniController support

Force Shebang update

Perl scripts contain a Shebang (the first line) which informs a server where to find the Perl program. The Uniform Server is dynamic and tracks any path changes, automatically updating the Shebang.

However if the server remains static and you add new scripts, the Shebang is never updated. You must therefore run "Force Shebang update" from the Perl control panel. It updates the Shebang to the current Perl location in all files.

Tip: When writing Perl scripts you can use the following for the Shebang:

#!

From the Perl control panel, run "Perl Shebang Update" and all pages will be updated with the correct Shebang.

Convert to Unix format

Perl scripts developed on Windows will not run on a Unix machine. They require conversion to Unix format. You can run the conversion script from the Perl control panel.

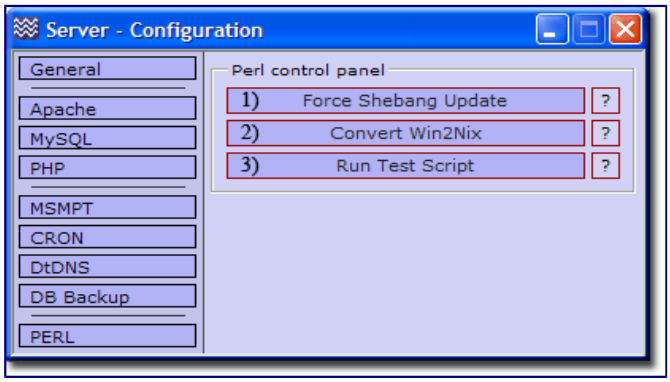
This script copies all files in cgi-bin to a new folder **cgi-bin-unix**
Scripts in this new folder are converted from Windows to Unix format.

- Converts end of line: Dec(#13#10=>#10) Hex 0D0A to 0A
- Replaces Windows Shebang with Unix Shebang

Perl control panel

This sub-menu option opens the Perl control panel.

Server Configuration > Perl > Perl control panel



1. **Force Shebang Update**
   Rewrites the Shebang in all .pl and .cgi files infolder UniServer\cgi-bin and its sub-folders.
2. **Convert Win2Nix**
   Copies the folders and files in cgi-bin to folder \cgi-bin-unix\ and converts from Windows to Unix format.
3. **Run Test Script**
   Runs the script UniServer\cgi-bin\test.pl

Perl - Install ActivePerl

ActivePerl is considered to be the de facto standard. However the ActivePerl Community License restricts the way the package can be distributed. Since this is inconsistent with The Uniform Server's license, ActivePerl is not included with The Uniform Server, nor is it available as a plug-in. On the other hand, you are free to download, install and use a personal copy of ActivePerl Community Edition. If you need to make it portable, it must be installed first and then integrated into The Uniform Server’s file structure.

This process is a bit tedious because ActivePerl is distributed only in msi format. Although the files are extractable without actually performing an installation, they do require relocating. The Uniform Server automates much of the installation process as explained below.

Contents

Download ActivePerl

First download the latest version of ActivePerl Community Edition from ActiveState (http://www.activestate.com/activeperl/downloads)

- As of August 2011, the current version is 5.12.4.1205 file (ActivePerl-5.12.4.1205-MSWin32-x86-294981.msi)
- Save the downloaded file to folder UniServer\alt_diag**install_perl**

Note: If you wish, save a copy of the downloaded file for archiving. After completing the installation, you can save space by deleting the folder UniServer\alt_diag\install_perl.

Install ActivePerl

To extract and install, double click on the batch file:

UniServer\alt_diag\install_perl**extract_install_perl.bat**

Note 1:
For a fresh install, this process is automatic and does not require any user input.

Note 2:
If ActivePerl is already installed, you will be prompted to delete the old version and install the new version. Press enter to delete and install. Entering anything other than "Yes" will terminate the installation.

Background

Files within a MSI installer file are easily extracted using a batch file.

Use the following batch command to perform extraction:

```
msiexec /a "Full path to MSI file" /qb TARGETDIR="Full path to target folder"

msiexec              - Batch command
/a                   - Forces all files to be reinstalled.
Full path to MSI file - Package name of the Windows Installer package file
/qb                  - Displays a basic user interface.
TARGETDIR=           - Folder where etracted files will be saved
```

Example:

```
msiexec /a F:\test\ActivePerl-5.12.3.1204-MSWin32-x86-294330.msi /qb TARGETDIR=F:\test\temp
```

These files can then be copied to the desired location.