

Technical Guide

November 2000
13T3-1100A-WWEN

Prepared by
dotCOM & Service Provider
Solutions

Compaq Computer Corporation

Contents

Solution Overview	5
Sizing Considerations	8
Installing and Verifying	
Components	11
Managing an Apache Server	24
Further Reading	27

Linux Web Solution with Apache, PHP, MySQL, and ht://Dig

Abstract

Of all the web servers on the market today, Apache is most popular because it supplies basic web server functionality right out of the box. Yet, many customers want a more sophisticated website, one with SQL database functionality, search capabilities, and server-side scripting. The complexity and interdependencies of these packages make it difficult for independent software vendors, solution developers, and website administrators to put together a solution.

Customizing the Apache server with additional functionality can be complex on the Linux platform. The solutions for extending functionality are just not obvious.

This technical guide demonstrates how to integrate PHP, MySQL, and ht://Dig with the Apache server. Independent software vendors, solution developers, programmers, and website administrators can use this guide to plan and deploy advanced Apache web servers on the Linux platform.

Send your comments and questions on this solution to activeanswers_support@compaq.com. Please include the Document Number (shown in the upper left corner of this page) in the Subject line of your message.

Notice

The information in this publication is subject to change without notice and is provided "AS IS" WITHOUT WARRANTY OF ANY KIND. THE ENTIRE RISK ARISING OUT OF THE USE OF THIS INFORMATION REMAINS WITH RECIPIENT. IN NO EVENT SHALL COMPAQ BE LIABLE FOR ANY DIRECT, CONSEQUENTIAL, INCIDENTAL, SPECIAL, PUNITIVE OR OTHER DAMAGES WHATSOEVER (INCLUDING WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION OR LOSS OF BUSINESS INFORMATION), EVEN IF COMPAQ HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The limited warranties for Compaq products are exclusively set forth in the documentation accompanying such products. Nothing herein should be construed as constituting a further or additional warranty.

This publication does not constitute an endorsement of the product or products that were tested. The configuration or configurations tested or described may or may not be the only available solution. This test is not a determination of product quality or correctness, nor does it ensure compliance with any federal state or local requirements.

Compaq, Deskpro, Compaq Insight Manager, Systempro, Systempro/LT, ProLiant, ROMPaq, QVision, SmartStart, NetFlex, QuickFind, PaqFax, and Prosignia are registered with the United States Patent and Trademark Office.

ActiveAnswers, Tru64, Netelligent, Systempro/XL, SoftPaq, Fastart, QuickBlank, QuickLock are trademarks and/or service marks of Compaq Computer Corporation.

Microsoft, Windows, and Windows NT are trademarks and/or registered trademarks of Microsoft Corporation.

Intel, Pentium, and Xeon are trademarks and/or registered trademarks of Intel Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Ltd.

Linux is a registered trademark of Linus Torvalds.

Red Hat, the Red Hat "Shadow Man" logo, RPM, Maximum RPM, the RPM logo, Linux Library, PowerTools, Linux Undercover, RHmember, RHmember More, Rough Cuts, Rawhide and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.

SuSE is a trademark of SuSE Inc.

Other product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

©2000 Compaq Computer Corporation.
All rights reserved. Printed in the U.S.A.

Linux Web Solution with Apache, PHP, MySQL, and ht://Dig Technical Guide
Prepared by dotCOM & Service Provider Solutions

First Edition (November 2000)
Document Number 13T3-1100A-WWEN

Table of Contents

1 Solution Overview	5
1.1 Apache Web Server.....	5
1.2 PHP3 Scripting Language	5
1.3 MySQL Database	6
1.4 ht://Dig Search Engine.....	7
2 Sizing Considerations	8
1.1 Small Configurations.....	8
1.2 Medium Configurations.....	9
1.3 Large Configurations	10
3 Installing and Verifying Components	11
3.1 Installing the Components	11
3.1.1 Installing MySQL	11
3.1.2 Installing Apache Server (Phase One)	12
3.1.3 Installing PHP3.....	13
3.1.4 Reconfiguring Apache Server (Phase Two)	14
3.1.5 Installing ht://Dig Search Engine	14
3.1.5.1 Changing the Location of Some Files.....	15
1.1.1.2 Changing the Starting Point for the Search Engine to Find Content.....	15
1.1.1.3 Changing Search Method from HTTP Connections to Entry through the File System	15
3.1.6 Building Search Engine Database.....	16
3.2 Verifying the Installation.....	16
1.2.1 Verifying Automatic Boot.....	16
1.2.2 Verifying PHP3 Operation	16
1.2.3 Verifying MySQL Support.....	18
1.2.4 Setting Up a MySQL Test Database	19
1.2.5 Verifying the PHP3 Connection to MySQL	20
1.2.6 Verifying that ht://Dig Is Working	22
4 Managing an Apache Server	24
4.1 Monitoring the Log Files	24
4.2 Monitoring Network Connections.....	24
4.3 Monitoring System Performance	24
4.3.1 Status Displays	24
4.3.2 Configuration Information.....	25
4.3.3 Monitoring Tools.....	25
4.4 Analyzing Log Files	26
5 Further Reading	27

1 Solution Overview

This document describes how to install, configure, and deploy a sophisticated Apache website on the Linux operating system. Such a website includes the powerful, server-side scripting language, PHP3, access to the full-featured SQL database, MySQL and the ht://Dig search engine. All of the software packages described in this document are open source applications.

What makes open source software so attractive? First of all, it is free. Secondly, you get copies of the software source code, which frees you to control the software to meet your short-term and long-term requirements.

The key points of open source software restrictions are that no one:

- Distributes the product without the source code (or, a list of differences between the modified version and the original) and a copy of all related documentation
- Charges for the software itself (only for its distribution)
- Can impose restriction on its distribution

For more information on licensing, refer to the regulations defined by the Open Source Initiative organization at: <http://www.opensource.org/osd.html>.

1.1 Apache Web Server

Apache is more widely used than all other web servers combined. A survey by Netcraft in September 2000 substantiates this claim. After polling over 21 million sites for web server software usage on Internet-connected computers, they found that over 61% of all Internet websites run on the Apache server. (For updates, see <http://www.netcraft.com/survey/>).

Clearly, the Apache Software Foundation is meeting its goal “to provide a secure, efficient, and extensible server that keeps its HTTP services in step with HTTP standards.”

The Apache Web Server Project homepage contains versions of the Apache server and corresponding documentation that you can download. The official website for the Apache Web server is <http://www.apache.org/>.

Getting Support on Apache

In addition to online documentation and a list of frequently asked questions, the Apache site maintains a number of mailing lists. To receive mail from other Apache developers about code releases, bug fixes, security fixes, and other information, join one, such as announce@apache.org. Locate the mailing list at <http://www.apache.org/announcelist.html>.

1.2 PHP3 Scripting Language

The PHP (originally Personal Home Page) scripting language supports the creation of dynamic, customized websites. With PHP, developers embed scripts within web pages. The web server calls PHP to perform the script operations, and then the server returns the page to the user. Functionally, PHP and the Microsoft Active Server Page (ASP) scripting language are quite similar.

As Table 1 demonstrates, PHP is rich and flexible software.

Table 1. PHP Features

PHP supports	Examples
Multiple platforms	Linux/Intel, Linux/Alpha, <i>Compaq Tru64</i> UNIX Microsoft Windows 95/98/NT and Windows NT
Multiple databases	MySQL, Oracle, Informix, Sysbase, and UNIX dbm.
E-mail: protocols:	IMAP (Internet Message Access Protocol) POP3 (Post Office Protocol 3) SMTP (Simple Mail Protocol)
USENET news protocol	NNTP (Network News Transfer Protocol)
Miscellaneous features	<ul style="list-style-type: none"> • Session-based authentication • Graphics library for dynamic image creation • Low-level network and socket interface

PHP popularity continues to grow. Many open-source and commercial projects choose PHP as an implementation language for web-based e-mail solutions, database access tools, and shopping carts for e-commerce sites, among others. The official PHP website at <http://www.php.net> publishes usage statistics it receives from NetCraft. At the current growth rate, as of September 2000, there will be in excess of 3.5 million (virtual) servers using PHP. To read a complete list of projects, select “Projects” on the PHP website.

Getting Support on PHP3

In addition to online documentation and a list of frequently asked questions, the PHP site maintains a set of mailing lists. As mailing list members learn about the PHP product, they can share their findings with others on the mailing list. If you prefer a bundled set of information, you can subscribe to the twice-daily digest. Multiple countries maintain mailing lists; at the time of this writing, there are versions in English, Italian, Portuguese, and Spanish. For general discussions subscribe to php3@lists.php.net.

To subscribe to this or another mailing list, see the PHP3 support page, <http://www.php.net/support.php>.

1.3 MySQL Database

MySQL (pronounced “my - S - Q - L”) is a multi-threaded SQL database server with a client/server implementation. Its speed, robustness, and ease of use make it a popular database for sites with dynamic content. PHP3, C, C++, Java, and Perl access MySQL directly.

The MySQL database source kits, plus pointers to manuals and SQL tutorials, are available at the official website for the MySQL database at <http://www.mysql.com>

MySQL Mailing Lists

The MySQL site maintains a set of mailing lists. As mailing list members learn about the MySQL product, they can share their finding with others on the MySQL site. Web designers can benefit from the general user mailing list and the announcements-only list. Information is also distributed for people using the MySQL-Java interface, the MySQL-ODBC interface, and the MySQL on Windows32 platform. Sign up for MySQL mail at: <http://www.mysql.com/documentation/index.html>.

1.4 ht://Dig Search Engine

The search engine package, ht://Dig, is a complete web indexing and searching system; it is one of the few open source software packages available for this purpose.

Unlike AltaVista, which can scale to Internet-wide searches, ht://Dig systems are designed for smaller configurations, such as small domains, single sites, or website subsections. If necessary, ht://Dig can index a number of web servers. With proper filters installed, ht://Dig can index and search through Adobe Acrobat (.pdf) files and Microsoft Word (.doc) files.

The source kit, manuals, and an archive of user mailing lists are located on the product's official website at <http://www.htdig.org/>.

Getting Support and Mail on ht://Dig

The ht://Dig site maintains a set of mailing lists. As mailing list members learn about the product, they can share their finding with others on the list. The two most useful mailing lists are as follows:

- For discussion of ht://Dig and related announcements: htdig@htdig.org
- For receiving notices of new releases: htdig3-announce@htdig.org

To subscribe to either, use the form on <http://www.htdig.org/mailarchive.html>.

2 Sizing Considerations

The performance of a web server system is dependent on a number of highly variable conditions that include:

- Popularity of the website
- The number of static pages (.html, .gif, .jpg) versus dynamically generated pages
- Complexity of the server side scripting code
- Size and design of the database tables
- Search engine usage
- Usage of encryption technology like SSL

Performance guidelines are based on the documentation provided with the Open Source software and on testing performed by Compaq.

Table 2 describes the typical Compaq *ProLiant* server systems that are used for the building blocks of interactive websites.

Table 2. Basic Configurations

Configuration	System	CPUs	Memory
Small	<i>ProLiant</i> 360 <i>ProLiant</i> 380	1 → 2	512 MB → 1 GB
Medium	<i>ProLiant</i> 380	1 → 2	512 MB → 2 GB
Large	Frontend: <ul style="list-style-type: none"> • <i>ProLiant</i> 360 • <i>ProLiant</i> 380 Backend: <ul style="list-style-type: none"> • <i>ProLiant</i> 380 • <i>ProLiant</i> 580 	See Distributed Internet Server Array (DISA) configuration documentation	

2.1 Small Configurations

The simplest configuration has all the components – the web server, MySQL database and search engine located on the same machine. For small configurations that have a relatively limited need for disk space, the rackmounted, thin, 1U form-factor, Compaq *ProLiant* DL360 is an appropriate choice. The Compaq *ProLiant* DL360 server has two internal Wide-Ultra2 or Wide-Ultra3 disks which means a minimum of 9 GB to a maximum of 36.4 GB on a high performance Integrated Smart Array.

If rack space density is not as high a priority as internal disk storage and PCI slots, the Compaq *ProLiant* DL380 is the best alternative because it supports up to six Wide-Ultra2 or Wide-Ultra3 internal disks.

A reasonable estimate for full Linux installation is around 3 GB, swap space is usually twice the amount of RAM, so for a maximum configuration that is an additional 8 GB of disk space. That leaves 25 GB available for the web server content, MySQL databases and search engine database.

As the web server becomes more active, be sure to increase the maximum number of concurrent Apache processes. In the Apache configuration file, `httpd.conf`, the entries for `MinSpareServers`

and `MaxSpareServers` will need to be increased to at least 100 and 20 respectively to allow for a quick response to a large number of requests being made at the same time. An Apache process without PHP scripting requires about 1 MB of memory but, if PHP is compiled in the process this can grow to 1.5 MB. Thus 100 concurrent Apache processes will require 150 MB of RAM.

Another major consumer of memory can be the MySQL processes. As with the Apache web server, for maximum performance there needs to be one process for each incoming request. For a simple database containing 500,000 entries, the RAM requirement for each MySQL process is between 0.5 and 0.75 MB.

Memory requirements will vary depending on website traffic, amount of static content versus dynamically created content, amount of dynamically created pages based on database queries. If a website expects to have 100 concurrent Apache processes running and one third are database driven that will require approximately 175 MB RAM. For optimal performance there should be enough free memory to allow for a large file buffer cache as well as the usual overhead of other processes running on the system. For a low volume website, Compaq recommends 512 MB of RAM is sufficient.

When the CPU load average consistently stays at or above 15, then it is time to upgrade to a second CPU. By default on most Linux distributions only a uniprocessor version of the kernel will be installed if there is only one CPU when it is first installed. After adding the second CPU, install the appropriate SMP kernel from the Linux distribution media or rebuild the kernel with SMP support enabled.

2.2 Medium Configurations

The Compaq *ProLiant* DL380 server is an excellent choice for larger sites that require a larger amount of disk space. The *ProLiant* DL380 supports up to 218 GB of storage by fully populating the Integrated Smart Array Controller with six 36 GB Wide Ultra-2 drives. As in the previous configuration, a *ProLiant* DL380 with 512 MB and one 933 MHz processor is suitable for moderately sized websites.

If your site's ht://Dig search engine is frequently used, adding a second CPU and increasing memory to 1 GB will dramatically improve performance.

The ht://Dig search engine's database is approximately the size of the web pages that it indexes. In the Linux kernel versions v2.2.x, there is a maximum file size of 2 GB, that means if you have a total web content is more than 2 GB ht://Dig will fail to index. At that point you need to segment the web content into separate indices for different content areas. In later Linux development kernels (v2.3.x), this maximum file limit has been increased to 4 GB. When v2.4.0 is available the limit will be raised into the terabyte range.

The MySQL database has a number of tuning parameters that need be considered when the number of Apache processes making database queries approaches 100. If the file `/etc/my.cnf` exists, the MySQL database server will use it for configuration on startup. The MySQL manual recommends the following `/etc/my.cnf` settings for large sites.

```
[mysqld]
set-variable      = max_connections=500
set-variable      = connect_timeout=10
set-variable      = key_buffer=384M
set-variable      = max_allowed_packet=1M
set-variable      = table_cache=512
set-variable      = sort_buffer=2M
set-variable      = record_buffer=2M
```

In addition, Compaq recommends the following parameters be increased as follows.

```
set-variable = max_connections=200
set-variable = connection_timeout=10
set-variable = back_log=200
```

Another area where a growing system may run into a problem is very large MySQL databases. As with ht://Dig, MySQL reaches the maximum file size problem if it has very large tables. If your site can not wait for v2.4.0 with the larger maximum file size, you may consider using another SQL database. PostgreSQL (postgresql.org) is another open source SQL database that also supports access from PHP and has the benefit of automatically splitting tables that would exceed an operating systems' maximum file size. MySQL performs better for 'mostly read' operations as found in web applications while PostgreSQL performs well for a more mixed read and write operations.

2.3 Large Configurations

For large website deployments, the way to scale up to meet the demands of a heavily loaded site is to expand from one system to a multi-system Distributed Internet Server Array (DISA) architecture. One backend server handles only the SQL database functions. Depending on your availability requirements, one of the Linux clustering packages should be considered such as Convolo (<http://www.missioncriticallinux.com>) or SteelEye (<http://www.steeleye.com/>) cluster solutions.

The Compaq ActiveAnswers site (<http://www.compaq.com/activeanswers>) contains a number of documents that describe in detail the DISA architecture.

3 Installing and Verifying Components

This guide assumes Linux is already installed on your system. Because most Linux distributions include the Apache software, the following procedures have you check and, if necessary, uninstall existing Apache software. All software in these procedures should be as current as possible.

3.1 Installing the Components

Note the order in which you install components is critical, so adhere to the following sequence:

1. Install MySQL
2. Install Apache Server (phase one)
3. Install PHP3
4. Reconfigure Apache Server (phase two)
5. Install ht://Dig search engine

3.1.1 Installing MySQL

Installing MySQL requires the following steps:

- Obtain the sources for the most up-to-date version from the official MySQL website database at <http://www.mysql.com>.

Once there, navigate to the "Downloads" page and identify the version number of the most recent release (at the time of this writing, Version 3.22.32). With that information, you can select a local, mirror site and verify it offers the most appropriate version. (Be aware that some mirror sites update their sources more often than other sites.) If the local mirror site lacks the correct version, use a different mirror site or download the source kit directly from <http://www.mysql.com/>.

Save the kit (for example, `mysql-3.22.32.tar.gz`) in a convenient location, such as `/usr/src/`, and unpack the kit using the following commands:

```
# cd /usr/src/  
# tar zxf mysql-3.22.32.tar.gz
```

- Uninstall any MySQL package on your system. Many Linux installations also install the Apache software. This is particularly common when the Red Hat Package Manager (RPM) utility downloads binary code from the Internet (as in RedHat, SuSE, and TurboLinux installations).

To check for an existing MySQL installation, enter the following command:

```
# rpm -qa | grep -i mysql
```

If MySQL software exists, remove it with commands similar to this:

```
# rpm -e mysql-3.22.27  
# rpm -e mysqldev-3.22.27
```

- Configure, build, and install the MySQL package with the following commands:

```
# cd /usr/src/mysql-3.22.32/  
# ./configure  
# make  
# make install
```
- Install the database that MySQL uses to manage itself with the following command:

```
# scripts/mysql_install_db
```
- Start the MySQL server manually with the following command:

```
# /usr/local/bin/safe_mysqld &
```

Also edit your system's startup script, /etc/rc.d/rc.local, to include the command:

```
/usr/local/bin/safe_mysqld &
```

Always include the ampersand (&) at the end of the line. Refer to the MySQL manual for additional switches to tune MySQL performance for your environment.
- Assign an administration password for root access to the database. Note that the single quotation marks enclose whatever password you assign. For example, enter a command such as the following:

```
# /usr/local/bin/mysqladmin -u root password 'newpassword'
```

3.1.2 Installing Apache Server (Phase One)

An Apache server installation takes place in two phases. In the first phase, you obtain the most recent sources and uninstall any pre-existing Apache server software. This phase puts files in place for the upcoming PHP compilation. After installing PHP, you complete the second phase of the Apache installation, which reconfigures the Apache server to use PHP3 libraries.

- Obtain the server sources from either the official Apache website at <http://www.apache.org> or from a closer, mirror site. As with the MySQL software, first check the most recent Apache software version and then refer to a list of mirror sites. Save the kit (for example `apache_1.3.14.tar.gz`) in a convenient location such as /usr/src and unpack the kit using these commands:

```
# cd /usr/src  
# tar zxvf apache_1.3.14.tar.gz
```
- Configure the Apache server. For this first phase, Apache must have the configuration script run. To do this, enter the following commands:

```
# cd apache_1.3.14/  
# ./configure --prefix=/usr/local/apache
```

- Uninstall any existing Apache server. (You probably have the Apache server if you are running one of the Linux distributions.) If you have made configuration changes, you may want to save the Apache file under a new name. Eliminate the existing version as follows:

```
# rpm -qa | grep apache
apache-1.3.3-1
apache-devel-1.3.3-1

# rpm -e apache-1.3.3-1 apache-devel-1.3.3-1
```

- In situations where other Apache components are installed, you will receive a warning about dependencies on other packages. They will have to be removed first. See the RPM man page ("man rpm") for information about using this tool. For example:

```
# rpm -e mod_perl-1.15-3 mod_php-2.0.1-5 mod_php3-3.0.5-2

# rpm -e apache-1.3.3-1 apache-devel-1.3.3-1
```

In the next step, you build PHP, and then you will complete Phase Two of the Apache installation.

3.1.3 Installing PHP3

Navigate to the official website for PHP3 at <http://www.php.net/>. This site provides the PHP3 source kit, manuals, code examples, and pointers to other open source packages that use PHP3.

- Obtain the sources. Go to the "Downloads" page and download the latest PHP3 source kit from the nearest mirror site. Save the kit (for example, php-3.0.17.tar.gz) in a convenient location such as /usr/src/ and unpack the kit using the commands:

```
# cd /usr/src/

# tar zxf php-3.0.17.tar.gz
```

Note: In the next step, use the same Apache src directory as in the previous commands for the Apache switch.

- Compile PHP3. PHP3 needs to be configured before it can be built as an Apache module and use MySQL.

There are a number of compile time options that you may want to enable. Popular options include IMAP mail support, GD graphics for image creation on the fly, Lightweight Directory Access Protocol (LDAP) support, WebDAV (HTTP Distributed Authoring and Versioning (DAV)) support, and zlib (the compressed library). These options require installation of additional kits and are beyond the scope of this document.

To compile PHP3, enter the following commands:

```
# cd /usr/src/php-3.0.17/
# ./configure --with-mysql --with-apache=../apache_1.3.14 \
    --enable-track-vars
```

- Install PHP3 with the following commands:

```
# make
# make install
```

3.1.4 Reconfiguring Apache Server (Phase Two)

At this point, the MySQL database is running, the Apache server is partially prepared, and the PHP libraries are installed. You now reconfigure the Apache server to use PHP.

- Return to the `apache_1.3.14` directory, as follows:

```
# cd /usr/src/apache_1.3.14/
```
- Build the Apache server with the PHP module as well as two modules named `mod-status` and `mod-info` that can provide web masters with configuration information and server status.

```
# ./configure --activate-module=src/modules/php3/libphp3.a \  
    --enable-module=info --enable-module=status  
# make  
# make install
```

- Edit the configuration files.

The default configuration for PHP is reasonable, but you can modify some resource limits. By default the maximum execution time for any PHP page (`max_execution_time`) is 30 seconds.

Move the configuration file for PHP into a system directory as follows:

```
# cp /usr/src/php-3.0.17/php3.ini-dist /usr/local/lib/php3.ini
```

- Edit the Apache configuration file, `/usr/local/apache/conf/httpd.conf`, so it processes files with the `php3` extension. Remove the `#` character from the following lines:

```
#AddType application/x-httpd-php3 .php3  
#AddType application/x-httpd-php3-source .phps
```

- Start the Apache Web Server with the following command:

```
# /usr/local/apache/bin/apachectl start
```

- Also edit your system's startup script, `/usr/rc.d/rc.local`, to include the command:

```
# /usr/local/apache/bin/apachectl start
```

3.1.5 Installing ht://Dig Search Engine

- Obtain the ht://Dig sources from the official website for ht://Dig at <http://www.htdig.org/>. Then go to the "Where to get it" page and download the latest source kit (at the time of this writing, 3.1.5).

Save the kit (for example, `htdig-3.1.5.tar.gz`) in a convenient location like `/usr/src` and unpack the kit using commands such as the following:

```
# cd /usr/src/  
# tar zxf htdig-3.1.5.tar.gz  
# cd /usr/src/htdig-3.1.5
```

Compile ht://Dig. By default, the search engine places files in the /opt/www/htdig directory. Building the kit is straightforward: configure, compile, and install it with the following commands. Notice the switches that set the default locations of the Apache directories.

```
# ./configure --with-cgi-bin-dir=/usr/local/apache/cgi-bin \
              --with-image-dir=/usr/local/apache/icons \
              --with-search-dir=/usr/local/apache/htdocs

# make
# make install
```

- Reconfigure the Apache server so it can find the images that ht://Dig uses. To do so, modify the file /usr/local/apache/conf/httpd.conf. Locate the section for aliases and insert this line at the end:

```
Alias /htdig/ "/usr/local/apache/icons/"
```

Since you have modified the Apache configuration file, you will need to restart the process to enable this change. The following command will restart the Apache server.

```
# /usr/local/apache/bin/apachectl restart
```

3.1.5.1 Changing the Location of Some Files

Since the ht://Dig search database can grow very large, it is a good idea to move the database to a partition with enough free space.

Note: Linux kernel v2.2.x running on an Intel platform has a maximum file size of 2 GB. Linux running on an *AlphaServer*TM platform has a maximum file size of 8 GB.

The storage requirements for the search database vary. To adequately plan for the search database, allocate sufficient storage for approximately twice the size of the expected content to be searched. The ht://Dig index file grows slightly larger than the amount of content that is indexed. This search product does not scale to large amounts of information.

To move the database, change the htdig.conf file (typically located in the /www/opt/htdig/conf/ directory) by modifying the following line:

```
database_dir: /opt/www/htdig/db
```

3.1.5.2 Changing the Starting Point for the Search Engine to Find Content

This is a list of URLs or file locations separated by spaces. By default, only links on the same system are followed. For example, the line changes the ht://Dig starting point to be the name of this web server. For example:

```
start_url: http://www.isp.com/
```

3.1.5.3 Changing Search Method from HTTP Connections to Entry through the File System

With this method ht://Dig traverses the content directly, a method which greatly reduces both the time needed to index a website and the amount of clutter in web logs.

Note: A search method change such as this works *only* if the web server and the search engine are on the same system or if they can access the same disks by NFS.

To demonstrate this, in the `htdig.conf` file, add an entry for `local_urls` that provides a mapping from the web space into the file system location. For example, if the content for `http://www.isp.com` is located on `/usr/local/apache/htdocs` and the content for `http://www.isp.com/products` is on `/home/data/`, you would have an entry that looks like this:

```
local_urls:
  http://www.isp.com/=/usr/local/apache/htdocs/
  http://www.isp.com/products/=/home/data/
```

3.1.6 Building Search Engine Database

After installing and optimizing `ht://Dig` to index the database, you run a script to build the search database. `ht://Dig` includes such a script in `/opt/www/htdig/bin/rundig.sh`. To work out any configuration problems, run the script from the command line. Once the script is working properly, run the script periodically as a cron job.

See “man cron” and “man 5 crontab” for information on how to set up batch jobs.

3.2 Verifying the Installation

Once all the software components have been installed, you need to verify that all the components are working properly. The following sections describe how to verify the following:

- Apache invokes the PHP3 scripting engine to process files and PHP3 has MySQL support built in.
- A MySQL test database can be set up.
- PHP scripting engine can connect to the MySQL database.
- `ht://Dig` search engine has indexed the site and is working.

3.2.1 Verifying Automatic Boot

Since the system startup scripts have been modified to start the Apache web server and MySQL database a system restart should be done before deploying the system. At this point of the procedure, all the servers are running so you can do the following verification steps and then check the startup procedure.

3.2.2 Verifying PHP3 Operation

Create a file named `phpinfo.php3` and add the following line to the file.

```
<?phpinfo()?>
```

Put the file in the Apache DocRoot, which, by default, is called `/usr/local/apache/htdocs/`.

Start a web browser and go to `http://127.0.0.1/phpinfo.php3`. If PHP3 is working properly, the web browser will display a screen similar to the one displayed in Figure 1. Be sure to verify that MySQL is listed under the extensions.

Figure 1. PHP3 Validation Screen

PHP Version 3.0.16

by [Rasmus Lerdorf](#), [Andi Gutmans](#), [Zeev Suraski](#), [Stig Bakken](#), [Shane Caraveo](#), [Jim Winstead](#), and countless others.

System: Linux example.compaq.com 2.2.14-5.0 #1 Tue Mar 7 21:07:39 EST 2000 i686 unknown
Build Date: Aug 23 2000

Extensions

Extensions	Additional Information
PHP core	CFLAGS= HSREGEX=
Basic Functions	No additional information.
PHP_DL	Dynamic Library support enabled.
PHP_dir	No additional information.
PHP_filestat	No additional information.
PHP_file	No additional information.
PHP_head	No additional information.
Sendmail	Path to sendmail: /usr/sbin/sendmail -t
Syslog	No additional information.
MySQL	Allow persistent links: Yes Persistent links: 0/Unlimited Total links: 0/Unlimited Client API version: 3.22.32 Compilation definitions: MySQL_INCLUDE= MySQL_LFLAGS= MySQL_LIBS=
Socket functions	No additional information.
Regular Expressions	Bundled regex library enabled

100%

3.2.3 Verifying MySQL Support

From a shell window, try to connect to the database server and then show what databases are available. Use the password you established in preceding Section 3.1.1 .

```
# mysqlshow -u root -p
Enter password: <password>
+-----+
| Databases |
+-----+
| mysql     |
| test      |
+-----+
```

Then try to connect to the database named mysql and look at the tables.

Note: A semicolon (;) terminates all SQL commands.

```
# mysql mysql -u root -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 29 to server version: 3.22.23b
Type 'help' for help.
```

```
mysql> show tables;
+-----+
| Tables in mysql |
+-----+
| columns_priv   |
| db              |
| func            |
| host            |
| tables_priv    |
| user            |
+-----+
6 rows in set (0.00 sec)
```

```
mysql> quit
Bye
#
```

3.2.4 Setting Up a MySQL Test Database

After verifying that MySQL is working properly, create a database and populate it with some data. The user password file is a convenient place to find some information in a fixed format. The following dialog provides sample commands and system responses.

1. The first step uses the MySQL admin utility to create a database named "etcpasswd".

```
# mysqladmin -u root -p create etcpasswd
Enter password:
Database "etcpasswd" created.
#
```

2. Enter the MySQL interactive monitor and connect to the newly created "etcpasswd" database using root privileges.

```
# mysql etcpasswd -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39 to server version: 3.22.32b

Type 'help' for help.
```

3. Create a table to hold the information. The following command creates a table named "users" that will hold the contents of the /etc/passwd file.

```
mysql> CREATE TABLE users (login VARCHAR(20), optpw VARCHAR(20), \
uid INT, gid INT, name VARCHAR(40), homedir VARCHAR(40), \
shell VARCHAR(40));
Query OK, 0 rows affected (0.00 sec)
```

4. Give all access privileges to the etcpasswd database to the user named, for example, "admin", but only from the same system on which MySQL is running. Be aware that this username is unrelated to Linux user accounts.

```
mysql> GRANT ALL PRIVILEGES ON etcpasswd.* \
TO admin@localhost \
IDENTIFIED BY 'somepassword';
```

5. Direct MySQL to import the file /etc/passwd into the "users" table. The /etc/passwd uses a single colon to separate values.

```
mysql> LOAD DATA INFILE '/etc/passwd' INTO TABLE users \
FIELDS TERMINATED BY ':';
Query OK, 34 rows affected (0.00 sec)
Records: 34 Deleted: 0 Skipped: 0 Warnings: 0
```

6. Verify that the information was loaded. The following SQL command displays only the login and name columns from the users table. The results are limited to the first five entries. To display all the information, use the command "select * from users;"

```
mysql> select login, name from users limit 1,5;
+-----+-----+
| login | name |
+-----+-----+
| bin   | bin   |
| daemon | daemon |
| adm   | adm   |
| lp    | lp    |
| sync  | sync  |
+-----+-----+
5 rows in set (0.00 sec)

mysql> quit
Bye
#
```

3.2.5 Verifying the PHP3 Connection to MySQL

Now, make sure all the software works together. After importing test data, verify that you can obtain data from MySQL using PHP. Below is sample code that will display in a table some information about all the users in the database that contain the letter "o". Put the code in some convenient location, for example /usr/local/apache/htdocs/demo.php3.

```
<?php
// Connect to the MySQL server running on this system
mysql_pconnect( "localhost", "admin", "somepassword" )
  or die( "Unable to connect to SQL server" );

// Use the etcpasswd database

mysql_select_db( "etcpasswd" )
  or die( "Unable to select database" );

// Get a list of the column names for the users database
$result = mysql_list_fields ( "etcpasswd", "users" );

// Get all fields for all rows whose login name contains the letter 'o'
$query = mysql_query("select * from users where login like '%o%'");

// Get a count of the numbers of rows that were returned
$affected = mysql_num_rows($query);

if(!$affected) {
  echo "No users with an 'o' in their name returned";
}
else {
  // Display the column names
  echo "<table align=center border=1>" ;
  echo "<tr> <th>" . mysql_field_name($result,0) . "</th>";
  echo "      <th>" . mysql_field_name($result,4) . "</th>";
  echo "      <th>" . mysql_field_name($result,5) . "</th></tr>";

  // Display each row
  while($row=mysql_fetch_row($query)) {
```

```
        echo "<tr><td> $row[0]</td>";
        echo "        <td> $row[4]</td>";
        echo "        <td> $row[5]</td></tr>";
    }

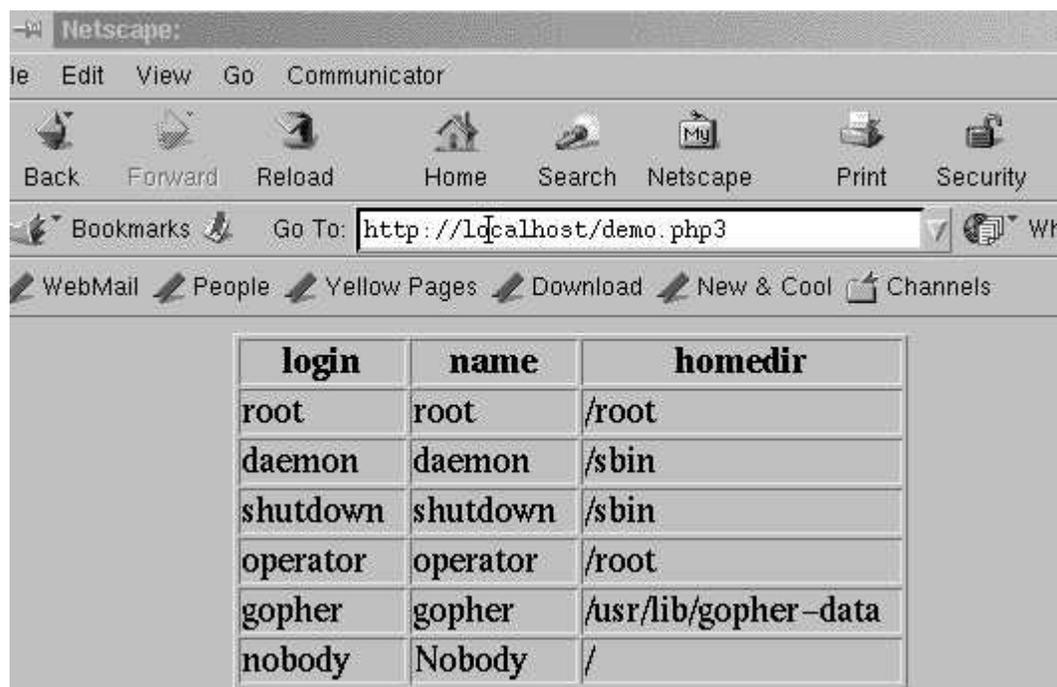
    echo "</table>\n";
}

// release memory associated with the query
mysql_free_result($query);

?>
```

The resulting web page will look similar to the one shown in Figure 2.

Figure 2. Result of Query to MySQL via PHP



3.2.6 Verifying that ht://Dig Is Working

After the `rundig.sh` script completes as described in Section 3.1.5, open the URL `http://127.0.0.1/search.html`. This URL should present a search query page, similar to the one shown in Figure 3.

Figure 3. ht://Dig Search Page



Enter a query and view the results. The `rundig.sh` script should have indexed the local copy of the Apache manual. Try searching for “server.” Figure 4 displays a sample of the results.

Figure 4. ht://Dig Search Results



4 Managing an Apache Server

Once installed, the Apache web server should run unattended for long periods of time. Yet it is wise to monitor the server to ensure it runs smoothly.

4.1 Monitoring the Log Files

To make certain your system does not run out of disk space, monitor the disk partition that holds the access and error logs. By default, the Apache server stores these log files at `/usr/local/apache/log/`.

The growth rate of log files varies widely because log size depends on the frequency of hits as well as the type of URL being requested. Although you can assume that each hit logs about 150 bytes of data, such a guideline is contingent on the structure of the web content's directory structure and naming conventions. One hundred and fifty bytes per hit translate to one megabyte of log growth for every 7000 hits.

The `logrotate` tool is designed to ease administration of systems that generate large numbers of log files. It allows automatic rotation, compression, removal, and mailing of log files. Each log file may be handled daily, weekly, monthly, or when it grows too large. If your Linux distribution does not include `logrotate`, you can obtain it from the following site:

<ftp://ftp.redhat.com/pub/linux/RedHat/redhat/code/logrotate/>

or one of the many RedHat mirrors.

4.2 Monitoring Network Connections

Some third-party open source Apache modules can limit the number of simultaneous connections to particular directories or slow down responses for users who exceed a given "bytes per second" download rate.

ApacheWeek maintains a list of modules that throttle access to web domains. For more information, see their web page at <http://www.apacheweek.com/features/modulesoup>

4.3 Monitoring System Performance

There are a number of open source tools to monitor systems in general and web servers in particular.

4.3.1 Status Displays

The module `mod_status` relays information about the general operation of the Apache web server, for example, its running time and a count of current requests, Apache child processes, total accesses and bytes sent

By default, the Apache configuration file (`httpd.conf`) has web access to status reports denied from everywhere.

To allow access, remove the comment from the Allow line and edit the location of Internet Service Providers (isp.com) to include a list of systems or domains that have access to the URL.

```
# Allow server status reports, with the URL of http://servername/server-status
# Change the ".your_domain.com" to match your domain to enable.
#
#<Location /server-status>
#   SetHandler server-status
#   Order deny,allow
#   Deny from all
#   Allow from .yourdomain.com
#</Location>
```

See the Apache documentation for more information at:

http://www.apache.org/docs/mod/mod_status.html

4.3.2 Configuration Information

The module mod_info returns comprehensive information about the server configuration; this information includes all installed modules and directives in the configuration files.

By default the Apache configuration file (httpd.conf) has web access for that URL denied from everywhere. Uncomment and change the Allow line to include a list of systems or domains that are allowed access to that URL.

```
#
# Allow remote server configuration reports, with the URL of
# http://servername/server-info (requires that mod_info.c be loaded).
# Change the ".your_domain.com" to match your domain to enable.
#
#<Location /server-info>
#   SetHandler server-info
#   Order deny,allow
#   Deny from all
#   Allow from .yourdomain.com
#</Location>
```

See the Apache documentation for more information at:

http://www.apache.org/docs/mod/mod_info.html

4.3.3 Monitoring Tools

Monitoring tools include the following:

- big brother system and network monitor
- mon, the Service Monitoring Daemon

The big brother system and network monitor is a web-based monitoring and notification package. With these monitoring tools, a website operator can centrally observe remote UNIX-based and Microsoft Windows NT systems.

The monitor is comprised of several clients: Local clients test system conditions and the availability of network services. These clients send status reports to display and pager servers who then notify system administrators about basic availability of web servers, DNS, NNTP (USENET news), SMTP, and POP3.

System status reports include information such as CPU usage, system log messages, operation of critical system processes, and disk space.

See the website <http://bb4.com> to download the sources, to see a demonstration, or for more information.

mon, the Service Monitoring Daemon is a general-purpose, resource monitoring system designed to check such conditions as network service availability, server problems, and environmental conditions.

Resource monitoring can be viewed as two separate tasks: the testing of a condition, and triggering some sort of action upon failure. mon was designed to keep the testing and action-taking tasks separate, as stand-alone programs. mon is implemented as a scheduler which executes the monitors (which test a condition), and calls the appropriate alerts if the monitor fails. See the website <http://www.kernel.org/software/mon> for more information, a sample page, and to download the source.

4.4 Analyzing Log Files

Webalizer

Webalizer is a fast and free, log file analysis program for a web server. It produces highly detailed, easily configurable usage reports, in HTML format, for viewing with a standard web browser. It can access information displayed by top URLs, source IP addresses of web browsers, URLs of pages that refer to your website, a chart of countries from which people are viewing the website, and browsers used.

To download the sources and gain more information, see their website at:

<http://www.mrunix.net/webalizer>

Analog

Analog can generate highly detailed and configurable usage reports in HTML format. While Analog is not as graphical as Webalizer, it is a solid tool.

To download the sources and gain more information, see their website at:

<http://www.analog.cx/>

Freshmeat.net

This is another good resource for locating log analysis tools specifically for Linux. It is located at:

<http://www.freshmeat.net/appindex/console/log-analyzers.html>

5 Further Reading

For information on the software packages discussed in this guide, see the following sites:

- Official website for the Apache web server at <http://www.apache.org>
Download the Apache source kit from here. Documentation for the web server is also available.
- Apache support site at <http://www.apacheweek.com>
Aims to be a resource for anyone running an Apache server, or anyone responsible for running Apache-based services. The site has a wealth of information on user authorization, server configuration, and procedures for extending an Apache server with software modules.
- Popular Linux distributions include:
 - Caldera Systems OpenLinux at <http://www.calderasystems.com>
 - Debian at <http://www.debian.org/>
 - Red Hat Software, Inc. at <http://www.redhat.com>
Offers one of the more popular versions of Linux in the United States
 - SuSE Linux at <http://www.suse.com/>
 - TurboLinux at <http://www.turbolinux.com/>
- Official website of the PHP3 scripting language at <http://www.php.net>
Support site for PHP3 at <http://www.phpbuilder.com/>

For relevant tutorials, articles, sample scripts and source files.

- Source for PHP library routines at <http://phplib.netuse.de/>
These routines simplify a number of complex operations. Most useful is the cookie or URL based authentication library that uses a MySQL database to maintain user information.
- Source for Zend software at <http://www.zend.com>
Zend dramatically increases the performance of the PHP scripting language. As of this writing, Zend is in beta testing.
- Official MySQL website at <http://www.mysql.com>
For general MySQL information, documentation, source files, mirror sites, pointers to sites with relevant tutorials.
- Official ht://Dig search engine site at <http://www.htdig.org>
For an indexing and searching search engine for a small domain or intranet.

- Clearinghouse for new and updated Open Source packages at <http://www.freshmeat.net>
Before writing your own solution, check the Freshmeat site for new and updated Open Source packages.
- The home page for the Open Source Organization at <http://www.opensource.org>