

June 2001  
156A-0602A-WWEN

Prepared by:  
dotCOM & Service Provider  
Solutions  
Compaq Computer Corporation

**Contents**

**Introduction..... 3**  
**Solution Overview ..... 3**  
    The DISA Architecture ..... 3  
    Web Server Layer ..... 5  
    Data Resources Layer ..... 5  
**MySQL Approach..... 6**  
    Web Server Layer  
    Configuration..... 6  
    Data Resources  
    Configuration..... 7  
    Verification of MySQL  
    Approach..... 8  
**PostgreSQL Approach ..... 10**  
    Web Server Layer  
    Configuration..... 10  
    Data Resources  
    Configuration..... 11  
    Verification of PostgreSQL  
    Approach..... 13  
**Further Reading..... 16**

# Integrating a Compaq *TaskSmart* W-Series Appliance into a DISA Environment

***Abstract:** This guide explains how to integrate a Compaq TaskSmart W-series appliance into a Compaq Distributed Internet Server Array (DISA) environment, as the Web server layer, with access to a MySQL or PostgreSQL server on the data resources layer. DISA is a best-practice architecture for multi-tier Internet infrastructure with both scalability and availability characteristics.*

*This architecture allows a Web hosting service provider or large website to quickly and easily add or expand their DISA environment.*

## 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, Compaq Insight Manager, Deskpro, FASTART, NetFlex, NonStop, PaqFax, ProLiant, Prosignia, QuickFind, Qvision, RomPaq, SmartStart, and Systempro/LT are registered with the United States Patent and Trademark Office.

ActiveAnswers is a trademark and/or service mark of Compaq Information Technologies Group, L.P.

Microsoft, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Intel, Pentium, and Pentium® III 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.

SteelEye Technology, Inc., LifeKeeper and the SteelEye Technology logo are trademarks of SteelEye Technology, Inc.

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.

MySQL is a registered trademark of MySQL AB in the U.S.A. and other countries.

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

©2001 Compaq Computer Corporation. All rights reserved. Printed in the U.S.A.

Integrating a Compaq TaskSmart W-Series Appliance into a DISA Environment  
Technical Guide prepared by dotCOM & Service Provider Solutions

First Edition (June 2001)  
156A-0602A-WWEN

# Introduction

This document explains how to integrate a *Compaq TaskSmart* W-series appliance into a Compaq Distributed Internet Server Array (DISA) architecture, as the Web server layer, with access to a MySQL or PostgreSQL server on the data resources layer.

This architecture allows a Web hosting service provider or large website to quickly and easily add or expand their *DISA* environment.

# Solution Overview

## The DISA Architecture

The Compaq Distributed Internet Server Array (DISA) architecture is a blueprint and set of recommended best practices for deploying highly-available and highly-scalable e-commerce/Internet applications. Internet applications requiring high availability and scalability should adhere to the DISA architecture best practices.

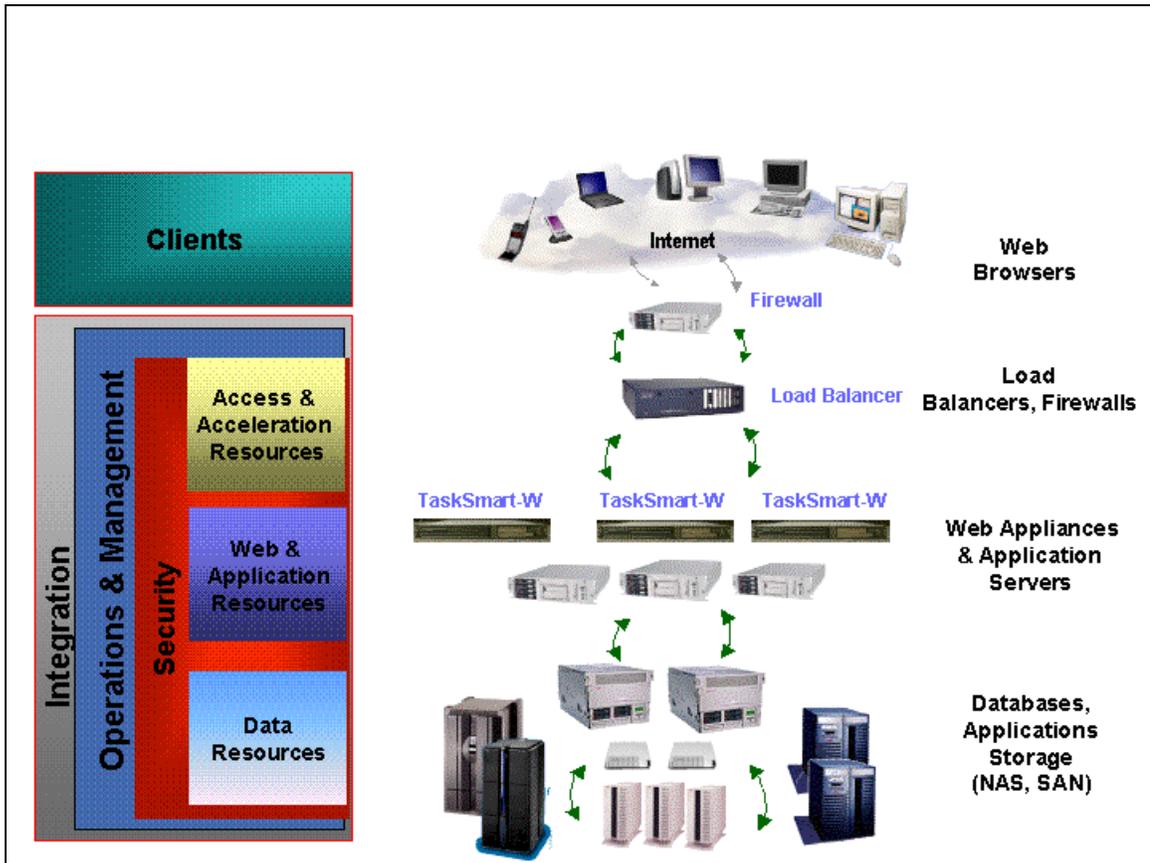


Figure 1: The *DISA* Architecture

The DISA Architecture defines three layers containing the components that deliver the application functionality for Internet client support:

- Access and Acceleration Resources Layer
- Web and Application Resources Layer
- Data Resources Layer

**Access and Acceleration Resources Layer:** This layer presents a single system image to clients in the form of a virtual host name to distribute client requests across multiple application servers. Load-balancing technology currently comes in three forms:

- Software-based, which is installed on a general-purpose server, such as Steeleye's LifeKeeper product or the open source Linux Virtual Server package.
- Device-based networking "appliances," such as Cisco, F5 Labs, RND, or HydraWeb
- Switch-based, such as Alteon, ArrowPoint, or Holontech

The selection of a particular load-balancing technology and vendor depends on the business requirements and specific characteristics of the application in question. For more information on Load Balancing, see the Compaq *ActiveAnswers* document entitled "Load-Balancing Considerations for Distributed Internet Server Array Environments" available at <http://www.compaq.com/activeanswers>.

**Web and Application Resources Layer:** The application servers perform the primary work of the Internet application. This layer includes Web servers, such as a Linux-based Apache Web server and the Compaq *TaskSmart* W-series appliance. Application servers may be linked to other data repositories, such as Enterprise Resource Planning (ERP) systems like SAP or other sales automation systems.

**Data Resources Layer:** The data resources layer is the location where the application data is stored, managed, and accessed. The DISA architecture avoids replication problems by providing a highly-available centralized file server rather than replicated data on application server drives. This layer contains the database that stores e-commerce data, such as product catalogs, user registration information, shipping information, and site activity logs. It also may provide connectivity to other systems that hold application data resources. Linux solutions in this area include Steeleye's LifeKeeper, Mission Critical Linux's Convolo clusters.

In addition to the three layers defined above, Security, and Operations and Management are two global components of a DISA system that require special attention.

**Security:** The DISA architecture allows for integration of required security components based on customer business requirements. DISA allows for strategic deployment of multiple firewalls between the distinct layers, such as a firewall between the load balancing and application server layers, and another between the application and data resources layers. Internet applications deployed via the DISA architecture can be integrated with all popular management frameworks.

**Operations and Management:** DISA systems involve many different servers that work together to deliver a complete application. A robust approach to systems management is required to keep all the systems online and running at peak performance. For in-depth guidance regarding current best practices for management of systems deployed using the DISA architecture, see the *ActiveAnswers* document entitled "Managing Multi-Tiered E-Business Applications" available at <http://www.compaq.com/activeanswers>. This document provides an application-centric perspective on systems management and provides detailed configuration examples and advice for using existing management tools and utilities to meet specific business requirements.

**Clients:** The Clients area is where application data is consumed. In a storefront, the most common client is a Web browser.

## Web Server Layer

The *TaskSmart* W-series appliance is deployed as the Web server layer in this configuration. Because the *TaskSmart* W-series already has the Apache Web server installed with PHP4 scripting support and all the necessary components to allow for remote database connections, adding this layer of the DISA environment is nothing more than writing the appropriate PHP scripts.

The PHP scripting language supports the creation of dynamic, customized websites. With PHP, developers embed scripts within Web pages. The *TaskSmart* W-series provides the PHP4 interpreter already integrated with the Apache Web server. 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-Supported Services	Examples
Multiple databases	MySQL, PostgreSQL, Oracle
E-mail protocols:	IMAP (Internet Message Access Protocol) SMTP (Simple Mail Transfer Protocol)
Directory Services	LDAP (Lightweight Directory Access Protocol) interface
Graphics Support	PNG, JPEG, GIF generation
Miscellaneous features	Session-based authentication SNMP support Low-level network and socket interface

## Data Resources Layer

The configuration of the server at the data resources layer of a DISA environment is beyond the scope of this document. It is assumed that you have access to either a MySQL or PostgreSQL database and have the privileges to create new database users, databases, and tables.

Both MySQL and PostgreSQL are popular choices for open source databases. The choice between these database servers or others is primarily one of preference of the service provider.

### 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. PHP, 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>.

## PostgreSQL Database

PostgreSQL (pronounced “post-gres-Q-L”), frequently referred to as ‘postgres’, 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. PHP, C, C++, Java, and Perl access PostgreSQL directly. Another key feature of PostgreSQL is its full support of SQL rollback and transactions.

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

## MySQL Approach

### Web Server Layer Configuration

The first step is to install and configure the *TaskSmart* W-series appliance into your existing network. When choosing the Web server configuration during the installation phase, be sure to choose either the ‘Dynamic Site, Normal Traffic’ or ‘Dynamic Site, High Traffic’ configuration. This will install the Apache Web server with the appropriate PHP4 scripting support.

### Preparing to Connect from the *TaskSmart* W-series to the Database Server

Below is sample code that will display information from a sample database needed for setup in later sections. Simply put this code in some convenient location, for example `/etc/apache/htdocs/mysql.php`.

This sample code connects to a MySQL database server residing on a data resources layer system (192.168.0.10), then opens a database (etcpasswd) and issues various SQL commands that will display all login names from the database that contain the letter ‘o’:

```
<?php

echo "<h1>Connecting via MySQL</h1>";

// Connect to the MySQL server running on the database server
mysql_connect( "192.168.0.10", "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]&nbsp;</td>";
        echo "      <td> $row[4]&nbsp;</td>";
        echo "      <td> $row[5]&nbsp;</td></tr>";
    }
    echo "</table>\n";
// release memory associated with the query
mysql_free_result($query);
}
?>

```

## Data Resources Configuration

The configuration of the server at the data resources layer of a DISA environment is beyond the scope of this document. It is assumed that you have access to a MySQL database and have the privileges to create new database users, databases, and tables. The following steps are done on the system hosting the database server.

### Setting Up a MySQL Test Database

The first step is to 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. Use 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. Also note that the lines are wrapped for readability. The backslash is the line continuation character.

```
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. Also, note that in this example, it is assumed that the IP address of the *TaskSmart* W-series Web server appliance is 192.168.0.1.

```
mysql> GRANT ALL PRIVILEGES ON etcpasswd.* \
      TO admin@192.168.0.1 \
      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 5;
+-----+-----+
| login | name |
+-----+-----+
| root  | root |
| bin   | bin  |
| daemon| daemon|
| adm   | adm  |
| lp    | lp   |
| sync  | sync |
+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> quit
Bye
#
```

## Verification of MySQL Approach

At this point, all the components have been configured and should be ready to verify. First, as described below, verify that the *TaskSmart* W-series Web server has access to the data resources layer database.

### Verifying the PHP Connection to MySQL

From a terminal on the *TaskSmart* W-series system, verify that the database is properly configured.

```
# mysql -h user184.ispworks.com -u admin -p etcpasswd
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 10 to server version: 3.23.33

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql> show tables;
+-----+
| Tables_in_etcpasswd |
+-----+
| users                |
+-----+
1 row in set (0.00 sec)

mysql> show columns from users;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| login      | varchar(20)   | YES  |     | NULL     |       |
| optpw      | varchar(20)   | YES  |     | NULL     |       |
| uid        | int(11)       | YES  |     | NULL     |       |
| gid        | int(11)       | YES  |     | NULL     |       |
| name       | varchar(40)   | YES  |     | NULL     |       |
| homedir    | varchar(40)   | YES  |     | NULL     |       |
| shell      | varchar(40)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> quit
Bye
$
```

## Verify Web Page

Using a Web browser, open the URL for the MySQL page, in this example:  
<http://192.168.0.1/mysql.php>.

The resulting Web page should look similar to the one shown in Figure 2.

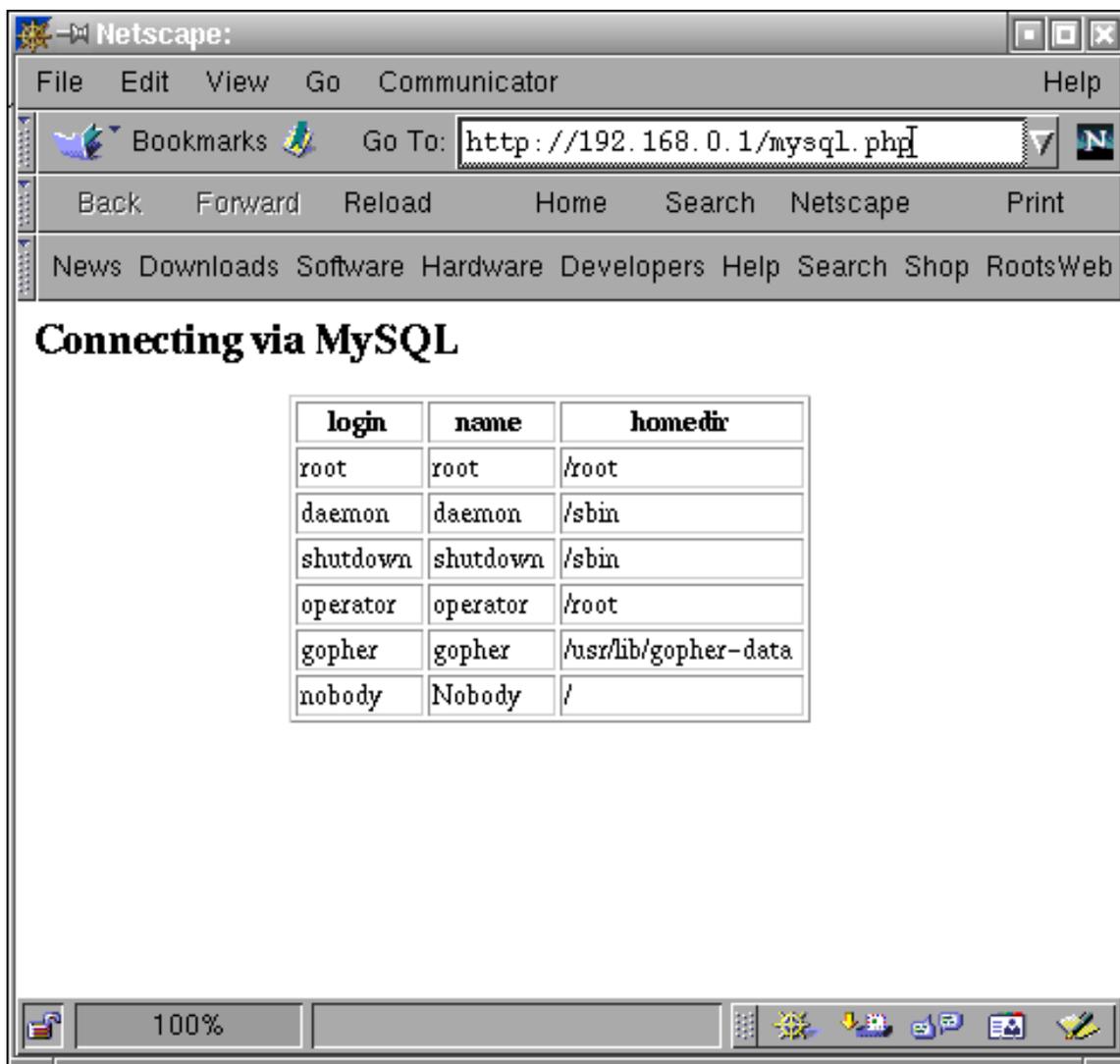


Figure 2: Result of Query to MySQL via PHP

## PostgreSQL Approach

### Web Server Layer Configuration

The first step is to install and configure the *TaskSmart* W-series appliance into your existing network. When choosing the Web server configuration during the installation phase, be sure to choose either the 'Dynamic Site, Normal Traffic' or 'Dynamic Site, High Traffic' configuration. This will install the Apache Web server with the appropriate PHP4 scripting support.

### Preparing to Connect from the *TaskSmart* W-series to the Database Server

Below is sample code that will display information from a sample database needed for setup in later sections. Simply put this code in some convenient location, for example:  
`/etc/apache/htdocs/pgsql.php`.

This sample code connects to a PostgreSQL database server residing on a data resources layer system (192.168.0.10), then opens a database (etcpasswd) and issues various SQL commands that will display all login names from the database that contain the letter 'o';

```
<?php
echo "<h1>Connecting via PostgreSQL</h1>";

// Connect to the PostgreSQL server running on the database server
$db_conn = pg_connect("host=192.168.0.10 dbname=etcpasswd user=admin
password=somepassword")
    or die( "Unable to connect to PostgreSQL server");

// Get all fields for all rows whose login name contains the letter 'o'
$result = pg_exec($db_conn, "select * from users where login like
'%_o%'");

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

if(!$affected) {
    echo "No users with an 'o' in their name returned";
}
else {
    // Display the column names
    echo "<table align=center border=1>" ;
    echo "<tr> <th>" . pg_fieldname($result,0) . "</th>";
    echo "      <th>" . pg_fieldname($result,4) . "</th>";
    echo "      <th>" . pg_fieldname($result,5) . "</th></tr>";
    // Display each row
    for ($i=0; $i<$affected; $i++) {
        $row=pg_fetch_row($result, $i);
        echo "<tr><td> $row[0]&nbsp;</td>";
        echo "      <td> $row[4]&nbsp;</td>";
        echo "      <td> $row[5]&nbsp;</td></tr>";
    }
    echo "</table>\n";
}
// release memory associated with the query
pg_freeResult ($result);
pg_close ($db_conn);
?>
```

## Data Resources Configuration

The configuration of the server at the data resources layer of a DISA environment is beyond the scope of this document. It is assumed that you have access to a PostgreSQL database and have the privileges to create new database users, databases, and tables. Access to the root account is also required because a system file needs to be updated. The following steps are done on the system hosting the database server as the root superuser.

## Setting Up a PostgreSQL Test Database

A postgres user needs to be created to 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. Create the PostgreSQL user that will be using the Web database.

```
$ createuser -U postgres --pwprompt
Enter name of user to add: admin
Enter password for user "admin": somepassword
Enter it again: somepassword
Shall the new user be allowed to create databases? (y/n) y
Shall the new user be allowed to create more new users? (y/n) n
CREATE USER
$
```

2. Use the PostgreSQL admin utility to create a database named "etcpasswd".

```
$ createdb -U admin etcpasswd "Sample database"
CREATE DATABASE
COMMENT
$
```

3. Enter the PostgreSQL interactive monitor and connect to the newly created "etcpasswd" database using the admin account.

```
$ psql -U admin etcpasswd
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

etcpasswd=>
```

4. 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.

```
etcpasswd=> create table users (login character varying (20), optpw
character varying (20), uid int4, gid int4, name character
varying(40), homedir character varying(40), shell character
varying(40));
CREATE
etcpasswd=>
```

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

```
etcpasswd=> \copy users from '/etc/passwd' using delimiters ':'
etcpasswd=>
```

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;"

```
etcpasswd=> select login, name from users limit 5;
select login, name from users limit 5;
 login | name
-----+-----
 root  | root
 bin   | bin
 daemon | daemon
 adm   | adm
 lp    | lp
(5 rows)
```

```
etcpasswd-> \q
$
```

7. Allow access to the etcpasswd database from the *TaskSmart* W-series system by editing the PostgreSQL host based authentication configuration file, `pg_hba.conf`. Typically the file is located in `/var/lib/pgsql/data/pg_hba.conf`, but depending on how PostgreSQL was installed on the database system, it may be in another location. Try using the following command to find it.

```
# locate pg_hba.conf
```

Then, as the root user, use a text editor, and add the following line highlighted in bold:

```
host etcpasswd 192.168.0.1 255.255.255.0 password
```

In this example, adding of the line to `pg_hab.conf` will allow a client on a *TaskSmart* W-series whose IP address is 192.168.0.1 to connect to the PostgreSQL etcpasswd database after the script running on the Web server has authenticated with a password.

---

**IMPORTANT:** Be sure that the file ownership on the `pg_hba.conf` file is reset to `postgres.postgres` by using the following command: **`# chmod postgres.postgres pg_hba.conf`**

---

## Verification of PostgreSQL Approach

At this point all the components have been configured and should be ready to verify. First, as described below, verify that the *TaskSmart* W-series Web server has access to the data resources layer database.

### Verifying the PHP Connection to PostgreSQL

From a terminal on the *TaskSmart* W-series system, verify that the database is properly configured.

```
# psql -h 192.168.0.10 -U admin etcpasswd
Password:
Welcome to psql, the PostgreSQL interactive terminal.

Type: \copyright for distribution terms
      \h for help with SQL commands
```

```
\? for help on internal slash commands
\g or terminate with semicolon to execute query
\q to quit

etcpasswd=> \d
      List of relations
  Name | Type | Owner
-----+-----+-----
  users | table | admin
(1 row)

etcpasswd=> \d users
 login      | varchar(20) |
 optpw     | varchar(20) |
 uid       | integer     |
 gid       | integer     |
 name      | varchar(40) |
 homedir   | varchar(40) |
 shell     | varchar(40) |

etcpasswd=> \q
$
```

### Verify Web Page

Using a Web browser, open the URL for the PostgreSQL page, in this example:

<http://192.168.0.1/pgsql.php>.

The resulting Web page should look similar to the one shown in Figure 3.

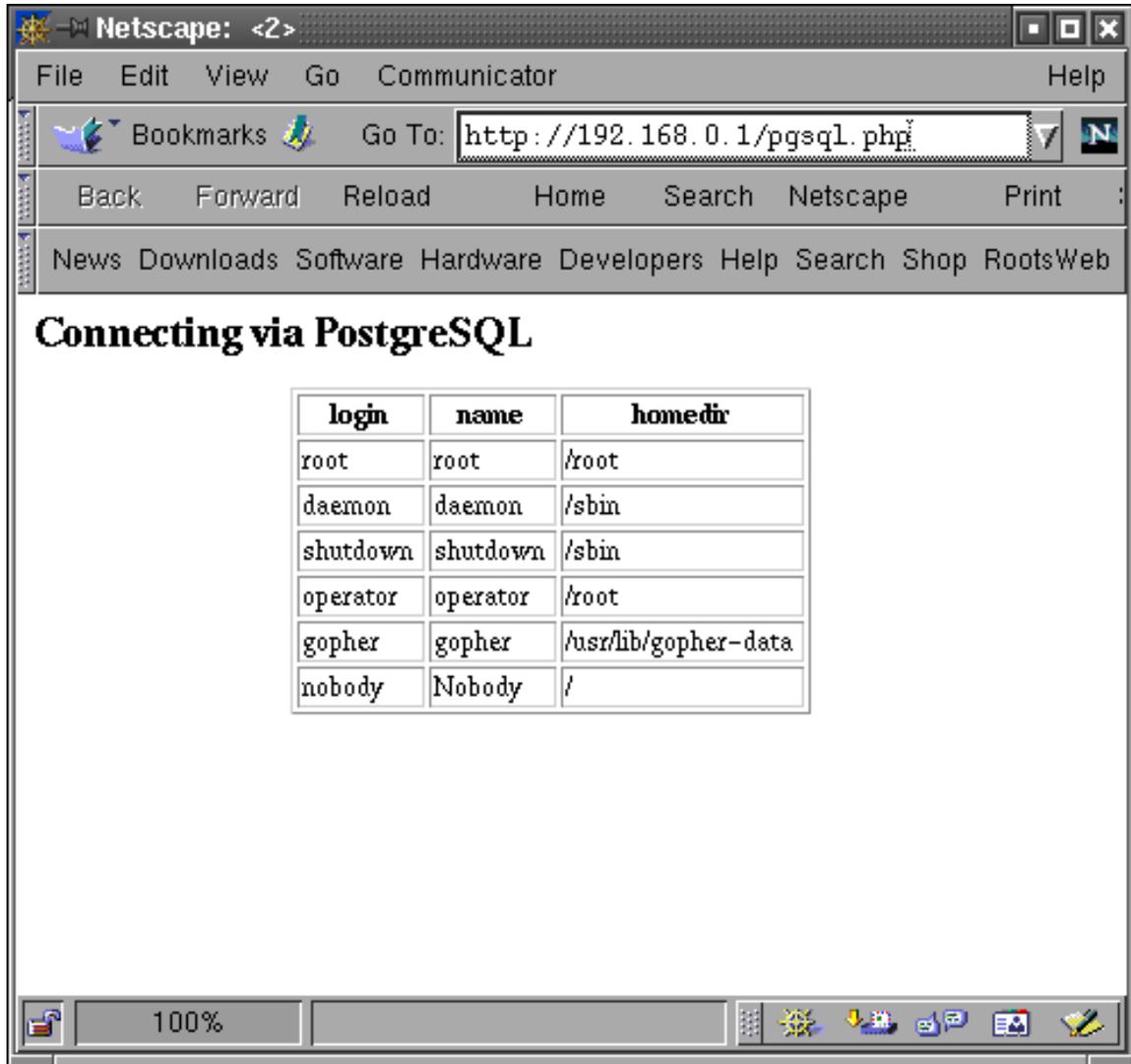


Figure 3. Result of Query to PostgreSQL via PHP

## Further Reading

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

- Official website for the Apache Web server at <http://www.apache.org>
- 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.

- Official website of the PHP scripting language at <http://www.php.net/>
- Support site for PHP 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.

- Official MySQL website at <http://www.mysql.com>

For general PostgreSQL information, documentation, source files, mirror sites, pointers to sites with relevant tutorials:

Official PostgreSQL website at <http://www.postgresql.org/>

For general PostgreSQL information, documentation, source files, mirror sites and pointers to sites with relevant tutorials:

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>