



David Larson
Bryan Logan
Tim Massaro
Heiko Mueller
Brian R. Smith

Bringing PHP to Your IBM @server iSeries Server

Hypertext Preprocessor (PHP) is a powerful server-side scripting language for the Apache Web server. PHP is popular for its ability to process database information and create dynamic Web pages. *Server-side* refers to the fact that PHP language statements, which are included directly in your HTML, are processed by the Web server. *Scripting language* means that PHP is not compiled. Since the results of processing PHP language statements is standard HTML, PHP-generated Web pages are quick to display and are compatible with most all Web browsers and platforms. PHP is for the open source Apache community as Net.Data® is for the IBM® community.

To “run” PHP scripts with your HTTP Server (powered by Apache), a PHP engine is required on your IBM @server™ iSeries™ server. The PHP engine is an open source product, so this IBM Redpaper demonstrates the process to download, compile, install, and then configure PHP on your iSeries. It explains how to install versions 4.3.0 and the older version 4.2.2 of PHP.

The PHP engine is available both as an Apache module and a CGI-BIN. Support for PHP as a module is not yet available for OS/400®. The step-by-step implementation discussed in this Redpaper involves the CGI version of PHP running in OS/400 Portable Application Solutions Environment (PASE). This allows you to run AIX® binaries directly on an iSeries. It includes the necessary patches for the minor modifications needed to the PHP source code.

Note: If you want to know why this is so great, see the article “Programming with PHP on the iSeries” for iSeries Network by David Larson and Tim Massaro. You can find this article on the Web at:

http://www.iseriesnetwork.com/resources/artarchive/index.cfm?fuseaction=viewarticle&CO_ContentID=15746&channel=art&PageView=Search

With permission from iSeries Network, we include the article in this Redpaper. To skip the article, go to “Prerequisites” on page 11.

Programming with PHP on the iSeries server

Restriction: PHP is not supported on the iSeries server by IBM. We have provided these instructions for you to download a public domain open-source copy of a PHP engine to allow you to run PHP scripts on the iSeries server.

Specifically, the following items *are not* supported by IBM:

- ▶ The open-source CGI-BIN based PHP engine
- ▶ Any of the PHP scripts that you would write against this PHP engine
- ▶ The other open source tools described in this Redpaper used for building the PHP engine.

These items *are* fully supported by IBM:

- ▶ 5722-SS1 Option 33 OS/400 PASE and all utilities supplied with it
- ▶ The VisualAge® C++ compilers
- ▶ The HTTP Server (powered by Apache) support for OS/400 PASE CGIs

Hypertext Preprocessor Language is a powerful, server-side scripting language for Web page creation. Scripting language means PHP requires no compilation, very much like Perl or Rexx. Because PHP is a server-side language, you can include it directly in HTML, and it is recognized and processed by a Web server.

The first “P” in PHP is a remnant from the original acronym for Personalized Home Page. This was the term that PHP creator Rasmus Lerdorf used when he first used a set of Perl scripts to monitor access to his online resume. Since then, however, PHP has become the most popular optional module configured on Web servers. See the following Web sites:

<http://www.netcraft.com/survey>

http://www.securityspace.com/s_survey/data/man.200204/apachemods.html

Here, we introduce the PHP language and walk you step-by-step through configuring PHP to access DB2® Universal Database™ (UDB) from your Apache Web server. Then, we provide examples to show you how iSeries shops can use PHP to create dynamic Web pages based on new or existing iSeries DB2 UDB databases.

What is PHP?

PHP code can easily access database files and output HTML, resulting in non-static, up-to-date Web pages. It's a technique similar to JavaServer Pages (JSPs) or Common Gateway Interface (CGI) binary (often called CGI-BIN) programming. Also, PHP is an open-source project. Open-source code can be useful if you want to tweak the behavior of PHP, but it's even more valuable because there are many open-source PHP applications and code samples available on the Web. This means you can get a new PHP Web project up and running quickly with little investment.

Hundreds of ready-made applications written in PHP are available as shareware, and many commercial products employ it. Until recently, PHP has enjoyed a reputation for reliability and security. See “Beware of PHP bugs” on page 11.

Figure 1 shows the difference between standard static Web pages and dynamic Web pages using server-side PHP processing. In the first scenario on the left, a standard URL request arrives at the Web server asking for Web page `http://www.somepage.html`. The Web server sees this request and returns the HTML that is in the file `somepage.html`.

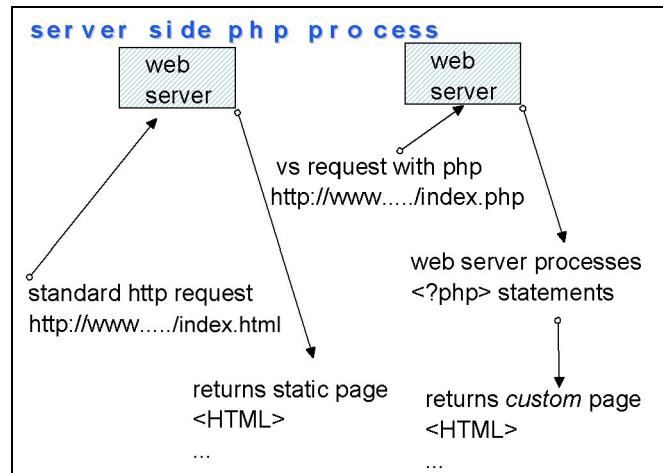


Figure 1 Left: Standard request for a Web page. Right: Request with PHP

In the second scenario on the right, the `index.php` file contains the special `<?php>` tag that tells the Web server to process embedded PHP statements. After PHP processes those statements, it returns HTML statements to the Web server. Those statements are then sent back to the user included in the original HTML found in the file `index.php`. Because the PHP statements can run a command or SQL statement, we say that the Web page is dynamically generated, as opposed to our previous static HTML page.

Why PHP?

Besides the fact that PHP is so popular, why would you want to use it? There are several reasons.

- ▶ **Easy to use:** As we mentioned before, PHP is a scripting language included directly in HTML. This means that getting started is easy. There's no need to compile PHP programs or spend time learning tools that create PHP. You can simply insert statements and get quick turnaround as you make changes.
- ▶ **Fully functional:** The PHP language has built-in functions to access your favorite database. With PHP, your HTML pages can reflect current information by querying those databases, or you can use information about the user viewing your HTML Web page to customize the page specifically for that user. In addition to good relational database support, PHP is a complete language that includes powerful functions. You can create classes for object-oriented programming and use flat file or Lightweight Directory Access Protocol (LDAP) databases. Plus, it includes a spell checker, XML functions, image generation functions, and more.
- ▶ **Compatible and quick:** Because PHP generates plain HTML, it's compatible with all Web browsers and refreshes quickly.
- ▶ **Secure:** Although PHP is open source, it's a secure environment. One of its advantages (over, JavaScript, for example) is that all that Web clients see is pure HTML. Because the logic of the PHP program is never exposed to the client, security exposures are reduced.
- ▶ **Open source:** Yet another reason to use PHP is because it's an open-source project, which makes it easy to find examples and get started quickly. If you search sites such as <http://www.sourceforge.net> and <http://www.phpresourceindex.com>, you can find code that you can use verbatim.

A code example

Let us look at a simple "HelloWorld!" example:

```
<html>
<head><title>Standard HTML Page with PHP
HelloWorld</title>
<body>
<?PHP
    print "Hello World";
    print "<br> Generated with <b>PHP</b>";
?>
</body>
</html>
```

This is as simple as it gets. The file starts as a normal HTML file. We simply insert PHP statements following the <?PHP tag. The <?PHP tag is the signal to the HTML processor that PHP processing is necessary. The print statement is a PHP statement.

To make this example a little more useful, let's add the following statements to query the state of our Web server:

```
<?PHP
    print "Hello World from System:" .
        $HTTP_SERVER_VARS['HTTP_HOST'];
    print phpinfo();
?>
```

The result of our PHP program would be similar to what is shown in Figure 2. This is a dynamic Web page that contains the name of our Web server and a table built by PHP with details about how PHP is configured on our Web server. This is accomplished by using one of several predefined PHP variables (for example HTTP_HOST) and the PHP function phpinfo.

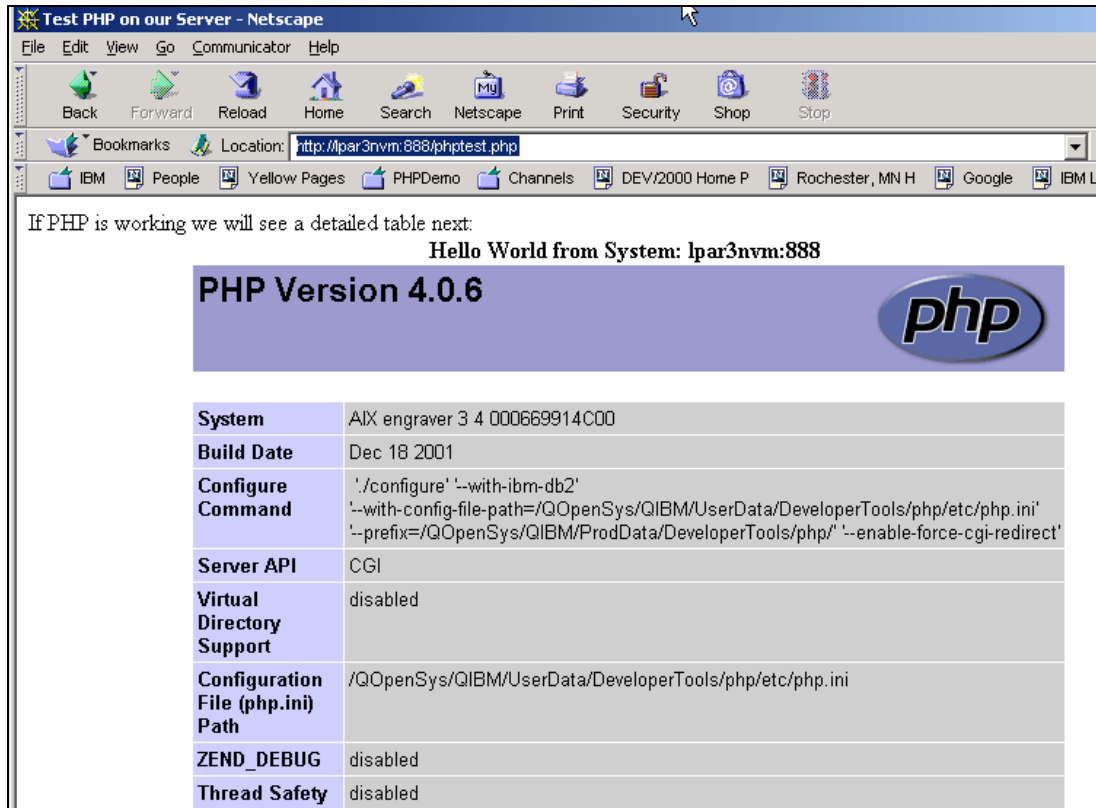


Figure 2 Dynamic Web page generated by the PHP 'Hello World' script

PHP on iSeries

An iSeries user has two options to set up PHP. You can use PHP with OS/400's Portable Application Solutions Environment (OS/400 PASE) and the HTTP Server (powered by Apache). Or you can install a Linux logical partition (LPAR) and run Apache and PHP in that partition. Table 1 shows factors to consider before you make this decision.

Table 1 Which is for you? PHP as a CGI in PASE versus PHP in a Linux LPAR

Factors to consider	PHP in PASE and Apache	PHP in Linux LPAR
OS/400 requirements	You should have V5R1 or newer. This should work for V4R5 - but we have not tried it ourselves.	You must have V5R1 or newer with specific hardware to run Linux.
Cost	A cost is associated with PASE (becomes free in V5R2). Note: in V5R1 and prior releases PASE was a nominal fee of around 100 US dollars.	A cost is associated with Linux distribution.
Setup required	No setup is required to use PASE.	Some setup associated with the creation of a Linux partition, user IDs, etc., and extra LPAR requires some dedicated processor resource.

Factors to consider	PHP in PASE and Apache	PHP in Linux LPAR
Availability and compatibility	You must obtain PHP from these instructions and have AIX skills to compile PHP as new versions come out.	Linux will be most compatible with new versions of PHP as they come out.
mySQL	mySQL is unavailable in PASE by default. You must download and compile it if desired.	mySQL is available as an alternative database (it's fairly common to use mySQL with PHP applications).
Web server module	PHP cannot be a Web server module. It must be a CGI-bin process only. This matters only in extremely performance-critical Web sites.	PHP can be installed as an Apache module.
Database	An ODBC driver is not necessary.	To use iSeries DB2 UDB, you must download, install, and configure iSeries ODBC Driver for Linux. This UNIX-based ODBC is free from IBM. It uses sockets to communicate between the Linux LPAR and the iSeries LPAR.

If you plan to install PHP on an iSeries server you need to be at V5R1 or later (as we mentioned in Table 1 this could work for V4R5 - but we have not tried it ourselves) and have PASE installed. PASE is the AIX runtime support for iSeries. See "Prerequisites" on page 11 to see if you have the "right stuff" for running PASE on your AS/400® or iSeries hardware.

If you plan to install PHP on a Linux LPAR, PHP is most likely included with your Linux distribution. If not, the installation instructions are virtually identical to those found in the PHP distribution itself and in the PHP site FAQs at:

<http://cvs.php.net/co.php/php4/INSTALL>

Regardless of where you install PHP, the configuration is the same. To get the Apache Web server to recognize PHP files, you must change the Web server configuration file to include some script aliases for PHP:

```
ScriptAlias /php-bin/ /usr/local/php/bin
AddType application/x-httpd-php .php
Action application/x-httpd-php /php-bin/php
```

The directory where PHP is installed may differ.

PHP as a CGI-BIN program

The next example shows a traditional HTML form that uses the Action tag to invoke a CGI-BIN program when a user clicks the Submit button. In this example, the CGI-BIN program is actually a PHP program that processes the fields in the HTML form and uses that information to query a DB2 database.

The database we use is called SAMPLE. SAMPLE is actually shipped with V5R1. To create it, follow the instructions at "Creating a sample database" on page 16 in this redpaper.

Figure 3 shows the basic HTML form that we use to perform a database query. Our system name is LPAR3NVM.

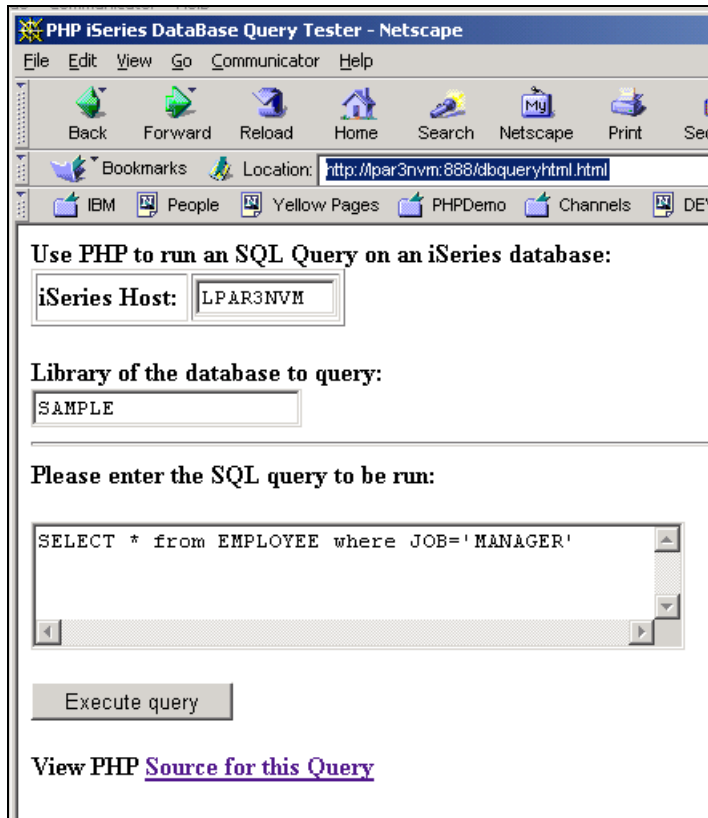


Figure 3 Basic HTML form used to perform a database query

Figure 4 shows the results of our query. Each record returned has been placed in a table row.

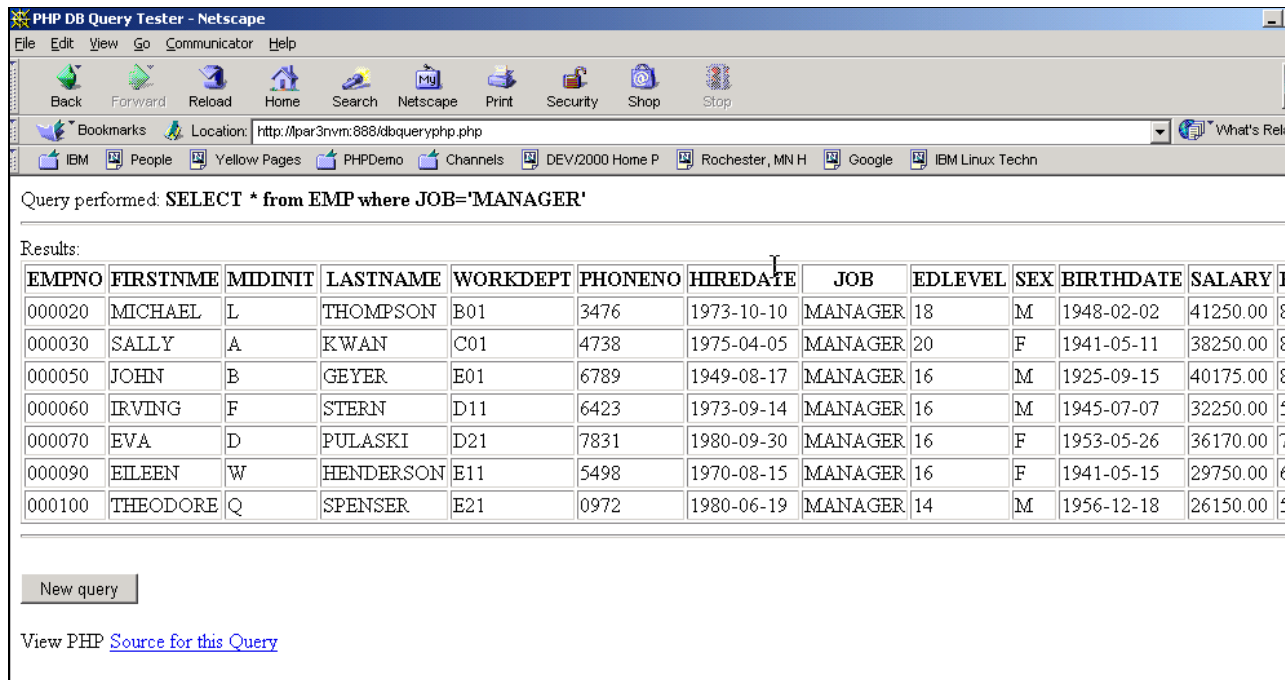


Figure 4 Results of the query

Here is the dbqueryphp.php script where the actual work is done:

```

<HTML>
<HEAD>
<TITLE>PHP DB Query Tester </TITLE>
</HEAD>
<BODY>
<!--dbqueryphp.php -->
<!--Called by dbqueryhtml.html -connect to sample db2 database and run an SQL
statement -->
<?php
    $host = $_POST['host'];
    $database = $_POST['database'];
    $query = $_POST['query'];
    if ($host && $database && $query) {
        $link =odbc_connect($host, "", "");
        if(!odbc_setoption($link,1,SQL_ATTR_COMMIT,SQL_TXN_NO_COMMIT)){
            echo "ERROR:unable to turn off commitment control!\n";
        }
        if(!odbc_setoption($link,1,SQL_ATTR_DBC_DEFAULT_LIB,$database)){
            echo "ERROR:unable to set default library to $database!\n";
        }
        $querynoslash =stripSlashes($query);
        $result =odbc_exec($link,$querynoslash);
?>
    Query performed:<B><?php echo ($querynoslash);?></B><HR>
    Results:<br>
    <?php
        if ($result ==0):
            echo ("<B>Error ".odbc_error()."."odbc_errormsg()."</B>");
        elseif (odbc_num_rows($result)==0):
            echo("<B>Query ran successfully</B>");
        else:
?>
    <TABLE BORDER=1>
    <TR>
    <?php
        for ($i =0;$i <odbc_num_fields($result);$i++){
            echo("<TH>".odbc_field_name($result,$i+1)."</TH>");
        }
?>
    </TR>
    <?php
        while(odbc_fetch_into($result,$row_array)!=FALSE){
            echo("<TR>");
            for ($j =0;$j <odbc_num_fields($result);$j++){
                echo("<TD>".$row_array [$j] . "</TD>");
            }
            echo("</TR>");
        }
        echo("</TABLE>");
    endif;
    } elseif ($host || $database || $query) {
        echo("All three fields must be filled in for a query<BR>");
    } else {
        echo("Use PHP to run an SQL Query on an iSeries database:<BR>");
    }
?>
    <HR><BR>
    <FORM ACTION=" <?php echo($_SERVER['PHP_SELF']) ?>" METHOD=POST>
    <TABLE BORDER=1><TR>

```



```

<TD>iSeries Host:</TD>
<TD><INPUT TYPE=TEXT NAME="host" VALUE="<?php echo ($host);?>"></TD>
</TR></TABLE>
<BR>
Library of the database to query:<BR>
<INPUT TYPE=TEXT NAME="database" VALUE="<?php echo ($database);?>">
<HR>
Please enter the SQL query to be run:<BR>
<TEXTAREA name="query" cols="40" rows="5">
<?php echo ($query); ?>
</TEXTAREA>
<BR>
<INPUT TYPE=SUBMIT VALUE="Execute query">
</FORM>
<p>View PHP <a href="dbqueryphp.php.source"target="_blank">Source for this Query</a>
</BODY>
</HTML>

```

The highlights include:

- ▶ **odbc_connect:** This is the “Open” of the database. The link variable is used by other odbc functions later in the script.
- ▶ **odbc_exec:** The variable filled in on the HTML form contains the string that we will run as an SQL statement. odbc_exe runs the SQL statement and returns results in the \$result variable.
- ▶ **odbc_numfields:** A function determines how many columns are returned for this record. We use this value to put HTML <TH></TH> tags around each cell.

Another PHP script

For one additional PHP example, let us include a script that will work only in the PASE version of PHP. This example takes advantage of the fact that the PASE “system” command writes any spooled output, produced by a command, to standard output. That is, you can run any commands with an OUTPUT(*PRINT) parameter in the PASE shell and have the results sent to STDOUT.

For example, if you're on the PASE command line QP2TERM, you can type the command `system wrkactjob` (Work with Active Jobs (WRKACTJOB)) and see the results as they scroll across the screen. Our example, `phpactjob`, simply formats this output into an HTML table. Figure 5 shows the output of this script.

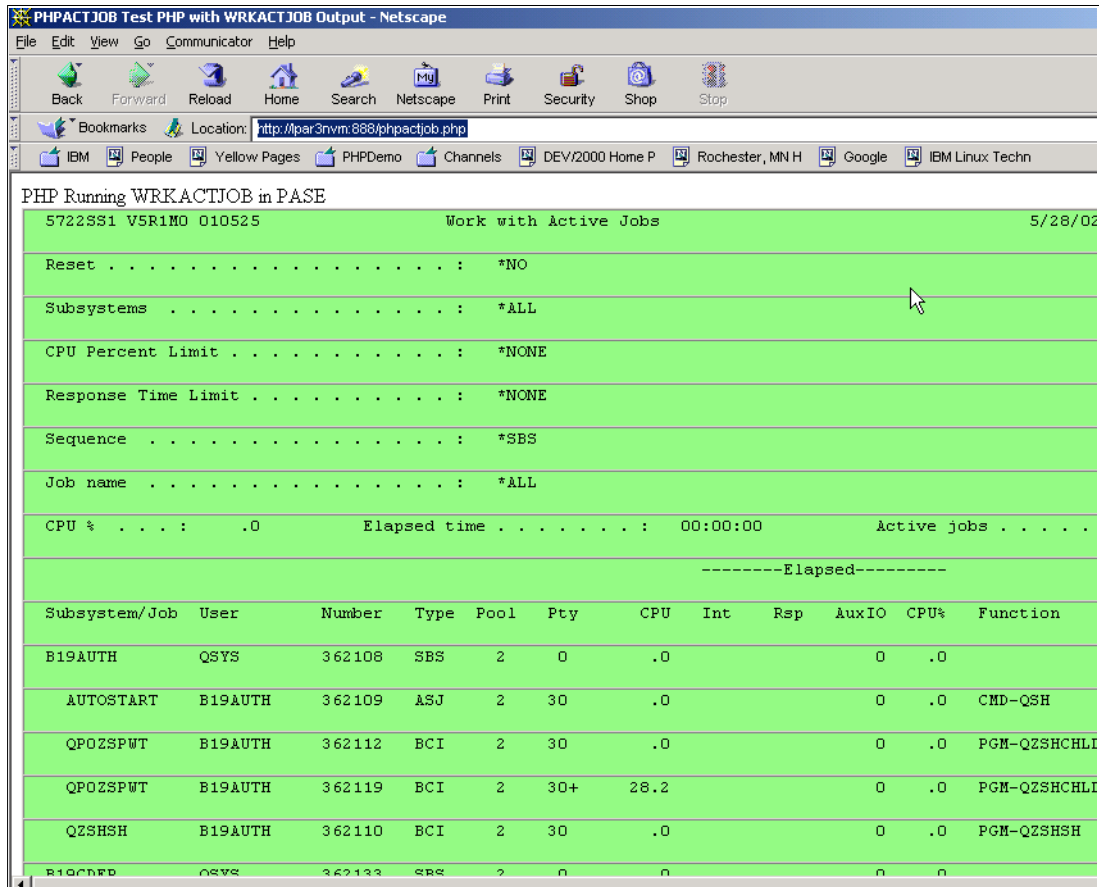


Figure 5 PHP formats the result of Work with Active Jobs (WRKACTJOB) at an HTML table

Here is the phpactjob source code. Note that we use back quotation marks (`) to run the command Work with Active Jobs (WRKACTJOB) and capture the output. This output is then broken into lines by searching for the new line character “\n” using the strtok function of PHP.

```

<html>
<head>
<title>PHPACTJOB Test PHP with WRKACTJOB Output</title>
</head>
<p>PHP Running WRKACTJOB in PASE<br>
<?PHP
$out=`/QOpenSys/usr/bin/system 'wrkactjob'`;
$line = strtok($out, "\n");
print "<table border CELLSPACING=0 CELLPADDING=0 BGCOLOR=\"#96FB84\">";
print "<tr>";
print "<td><pre>";
print $line;
print "</pre></td>";
print "</tr>";
while ($line = strtok("\n"))
{
print "<tr>";
print "<td><pre>$line</pre></td>";
print "</tr>";
}
print "</table>";
print "thend";
?>

```

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

A starting point

You have been introduced to PHP and how to use it on the iSeries server. Use this article as a starting point to find other examples and documentation that can have you running PHP in no time.

- ▶ Probably the best place to find help, tutorials, and examples about PHP is the PHP project Web site at:

<http://www.php.net>

- ▶ You can also check out the PASE on iSeries PartnerWorld® Web site at:

<http://www-919.ibm.com/developer/factory/pase/overview.html>

- ▶ Also see the iSeries Linux home page at:

<http://www.iseries.ibm.com/linux>

- ▶ You can find a neat demo application showing PHP using ODBC Driver for Linux at the following Web site:

<http://www-1.ibm.com/servers/eserver/iseries/linux/odbc/guide/demoindex.html>

This demo includes PHP using binary large objects (BLOBs) that contain employee photos in the sample iSeries EMPLOYEE database.

Beware of PHP bugs

Recently, a few security holes were discovered in PHP. The most recently discovered one involves the code for handling file uploads. This flaw lets hackers easily crash the PHP server and possibly take it over remotely. The flaw affects PHP versions 4.2.0 and 4.2.1. CERT rates the problem as *critical*.

The PHP Group announced a fix release, version 4.2.2, that all PHP users employing PHP's file-upload facility should install immediately. The fix is available on the Web at:

http://www.php.net/release_4_2_2.php

Prerequisites

This IBM Redpaper assumes that you have the following hardware and software on your iSeries server:

- ▶ **5722-SS1:** OS/400 (5722-SS1) at V5R2 (though the same basic steps should work on an iSeries server at V5R1)
- ▶ **5722-SS1 Option 13:** OS/400 - System Openness Includes
- ▶ **5722-SS1 Option 33:** OS/400 PASE

Note: We assume for this document that you are running at V5R2 of OS/400. If you have OS/400 V5R2 then all you must do is to make sure that 5722-SS1 Option 33 OS/400 PASE has been installed.

Since OS/400 V5R1 supports some levels of AS/400 hardware that is not supported by PASE (PASE requires a certain version (level) of PowerPC® processor) you must first determine if your AS/400 hardware supports PASE. You can find a detailed list of processors on which PASE can run on the Web at:

<http://www.as400.ibm.com/developer/factory/pase/ehardware.html>

- ▶ **5722-DG1:** IBM HTTP Server for iSeries. This LPP contains the HTTP Server (powered by Apache), which is the only HTTP server for which PHP will work.

Also, install the latest Apache group PTF package. For V5R2, the group PTF package number is SF99098.

- ▶ The command **make**

A **make** command can be found in OS/400's PASE for V5R2.

If you are using V5R1 of OS/400 then you will have to download the **make** command. We suggest using the GNU **make** command that can download from

<http://www.gnu.org/directory/gnu/make.html>

- ▶ **5799-PTL:** (Optional) PRPQ iSeries Tools for Developers. This tool kit is *optional* for this work but you may find it useful for some other similar projects. For details see

<http://www.iseries.ibm.com/developer/factory/tools>

This Redpaper also assumes that you have the following hardware and software on your *build machine*. The build machine could be either a separate pSeries™ running AIX or an iSeries running OS/400 with the following software:

- ▶ The command **patch**

If you do not have a patch program on your system, try the GNU **patch**. The GNU **patch** program is usually not on AIX or OS/400 machines. You can download version 2.5 (not 2.5.4) from:

<ftp://ftp.gnu.org/pub/gnu/patch>

To compile the source, follow these steps:

- a. Untar the source using the **tar** command.
- b. Type **cd** to go to the directory.
- c. Perform a **./configure**.
- d. Run the **make** command.
- e. Run **make install**.

- ▶ The GNU command **gzip** command to compresses and decompresses files. You can download this from:

<http://www.gnu.org/directory/GNU/gzip.html>

- ▶ The VisualAge C++ compiler for AIX. You can find information about this compiler at:

<http://www.ibm.com/software/ad/vacpp/>

If your build machine will be AIX (not OS/400) you must match the AIX version to the target OS/400 PASE version. That is, the application binary created on AIX needs to be compatible with the version of OS/400 PASE that you want to the application to run in. To help you plan this issue see:

<http://publib.boulder.ibm.com/series/v5r2/ic2924/info/rzalf/rzalfplanning.htm>

We have tested these instructions on AIX 4.3 and newer.

Alternatively, V5R2 of OS/400 PASE now supports installation of either the IBM VisualAge C++ Professional for AIX Version 6.0 or the IBM C for AIX Version 6.0 software products. This means you can compile OS/400 PASE applications within OS/400 PASE. A separate AIX system is not required. IBM VisualAge C++ Professional for AIX Version 6.0 (5765-F56) and IBM C for AIX (5765-F57) are separately available program products from IBM. Note that the VisualAge C++ Professional for AIX compiler product also includes the C for AIX compiler product.

Note: At the time of this writing an offer for free 60-day trial version of VisualAge C++ Professional for AIX, V6.0 is now available for download. See:

<http://www14.software.ibm.com/webapp/download/search.jsp?go=y&rs=vacpp>

Installation instructions

Follow these steps to download and prepare the PHP source files for compile.

Downloading PHP

Download the version of PHP you need for your iSeries.

Note: We include the patch files for both the 4.3.0 and the older 4.2.2 versions of PHP. These instructions, however, are written for version 4.3.0.

1. Download the tar file `php-4.3.0.tar.gz` for PHP 4.3.0 from the following Web site:
<http://www.php.net>
2. Using FTP, send this file to the machine on which you will build PHP. This may be the AIX machine or the iSeries machine with the VisualAge compiler. We will call this your *build machine*.
3. Untar the file by using the following commands:

```
gunzip php-4.3.0.tar.gz
tar -xvf php-4.3.0.tar
```

Patching the source code file

A patch is required to run PHP on the iSeries. We included patch files for both the 4.3.0 and the older 4.2.2 versions of PHP. The patch changes the default PHP DB2 support from AIX DB2 to OS/400 DB2.

1. Download and save the patch file to the build machine. You can find the file on the ITSO home page at:

<http://www.redbooks.ibm.com/>

Click **Additional materials** to access the directory listing of additional materials to download. Open the directory **REDP3639**, in which you will find the files `php422pase.patch` and `php430pase.patch`.

Download the patch file into the same directory from which you ran the `tar` command.

2. Change directory (`cd`) to that directory and run the following patch command:

```
patch -p0 < php430pase.patch
```

Locating iSeries specific files

You must locate and bring to your build machine the following iSeries files:

- ▶ The `sqlcli.h` and the `libdb400.exp` files contain DB2 UDB AS/400 support.
- ▶ The `as400_libc.exp` file is an iSeries-specific extension to the AIX file `libc.a`. This file is part of 5722-SS1 Option 13 OS/400 - System Openness Includes.

Follow these instructions to obtain the files from your iSeries:

1. Enter the following command:

```
CPY OBJ('/QIBM/include/sqlcli.h') TODIR('/home/yourid') TOCCSID(*STDASCII) DTAFMT(*TEXT)
```

- Using FTP, send the /home/yourid/sqlcli.h file from your iSeries to the build machine.
- Send, using FTP, the libdb400.exp and as400_libc.exp files from the iSeries directory /QOpenSys/QIBM/ProdData/OS400/PASE/lib to the AIX build machine.

Preparing for the PHP compile

Follow these steps to prepare the files and directories needed for the successful compile of PHP on your build machine. These steps are assuming ksh.

- Set the CFLAGS, CC, and CPPFLAGS environment variables as follows.

Note: The flags for -I and -bI are the uppercase format of the letter “i”.

```
export CFLAGS='-ma -DPASE -I /home/yourid -bI:/home/yourid/libdb400.exp
-bI:/home/yourid/path/as400_libc.exp'
export CC=xlc
export CPPFLAGS=-qflag=e:e
```

- Change to the php-4.3.0 directory using the `cd` command.
- Run the following command:

```
./configure --with-ibm-db2 \
--with-config-file-path=/QOpenSys/php/etc \
--prefix=/QOpenSys/php/ \
--enable-force-cgi-redirect \
--without-mysql
```

- If you are compiling directly in PASE on iSeries, add the following configure flags (be sure to add a “\” to the end of the previous line):

```
--build=powerpc-ibm-aix4.3.3.0 \
--host=powerpc-ibm-aix4.3.3.0
```

The configure should take some time to run. After it finishes, you need to make final adjustments to the files listed in the following steps.

- Edit the Makefile:

Note: The Makefile is generated with lines greater than 2048 characters. Some editors, such as vi, cannot handle the long lines, so you need to use a different editor. FTP the Makefile to a different machine and back if necessary.

```
remove -ldb2 from ODBC_LIBS
remove -ldb2 from EXTRA_LIBS
```

- Edit the config_vars.mk file:

Note: PHP version 4.3.0 does not have a config_vars.mk. This step is for PHP version 4.2.2 only.

```
remove -ldb2 from ODBC_LIBS
remove -ldb2 -lbind from EXTRA_LIBS
```

- Edit the main/build-defs.h file:

```
remove -ldb2 from PHP_ODBC_LIBS
```

- Edit the main/php_config.h file:

```
Delete #define HAVE_MMAP 1
Delete #define HAVE_SETITIMER 1
```

If you run this on a V5R1 OS/400 server, use the following command:

```
Delete #define HAVE_STATVFS 1
Delete #define HAVE_PREAD 1
Delete #define HAVE_PWRITE 1
```

Compiling (make)

You have two choices depending on whether you are compiling in AIX on the pSeries or in PASE on the iSeries.

If you are compiling in PASE on the iSeries

Follow these steps if you are compiling the PHP source code on your iSeries:

```
make
make install
mkdir /QOpenSys/php/etc
cp php.ini-dist /QOpenSys/php/etc/php.ini
```

This installs and puts all the files in the correct directory. You need write access to the /QOpenSys directory.

At this point, you may skip to “Testing PHP” on page 16.

If you are compiling in AIX on the pSeries

Follow these steps if you are compiling the PHP source code on your pSeries:

1. Edit the Makefile (see the note in step 5 on page 14 about the long lines of the Makefile) for the line "install_targets =", remove "install-pear".

2. Enter the following commands in the order shown:

```
mkdir /tmp/QOpenSys
```

3. At the AIX prompt, run the following commands:

```
make
make install INSTALL_ROOT=/tmp/
```

This installs PHP into /tmp/QOpenSys/php.

4. Enter the following commands in the order shown:

```
mkdir /tmp/QOpenSys/php/etc
cp php.ini-dist /tmp/QOpenSys/php/etc/php.ini
```

5. Edit the Makefile (see the note in step 5 on page 14 about the long lines of the Makefile) for the line "install_targets =", add "install-pear".

If the location of your home directory on your AIX box is different than the location of your home directory in PASE (for example, on AIX your home directory is /usr/home/usr4/jdoe and on PASE it is /home/john), replace all occurrences of "/usr/home/usr4/jdoe/" to "/home/john/" in the Makefile. Make sure that you include the first and last "/" so you don't lose your directory separator.

6. Enter the following commands in the order shown:

```
cd /tmp
tar -cvf ~/php430pasebin.tar QOpenSys
cd ~
tar -cvf php430pasesrc.tar php-4.3.0
```

- Using FTP, send both php430pasebin.tar and php430pasesrc.tar to your home directory on the iSeries server.
- Enter the following commands in the order shown:

Note: The following steps are all done in PASE on the iSeries and not in AIX.

```
cd /
tar -xvf ~/php430pasebin.tar
cd ~
tar -xvf php430pasesrc.tar
cd php-4.3.0
make install-pear
```

Testing PHP

From the PASE shell in OS/400, run the command:

```
/Q0penSys/php/bin/php -v
```

This should tell you the version of PHP you have.

Note: If you try running the PHP binary in PASE and it dies with an illegal instruction, check for the existence of a job log.

Several things can cause an illegal instruction signal and kill a PASE application. If the illegal instruction was caused by an unsupported system call, the name of the system call will be specified in the job log.

The job log should tell you the name of the illegal instruction. Next find the corresponding HAVE_ line in the php_config.h and delete it. Then recompile. This should only happen if you're compiling on a version of AIX that supports certain syscalls that PASE does not support (in addition to the five noted earlier).

Configuring HTTP Server (powered by Apache) to use PHP

Edit the file httpd.conf using the Apache GUI interface. The key statements needed are:

```
ScriptAlias /php-bin/ /Q0penSys/php/bin/
AddType application/x-httpd-php .php
Action application/x-httpd-php /php-bin/php
<Directory /Q0penSys/php/bin>
    Options +ExecCGI
    order allow,deny
    allow from all
</Directory>
```

Stop and start your HTTP Server (powered by Apache) Web server.

Creating a sample database

Here we could add the creation of the sample database as supplied with the system. Starting in V5R1, a sample database is shipped with the system. This is explained on the Web at:

<http://www.ibm.com/servers/eserver/iseries/db2/sqldata.htm>

To unpack and create the sample database, invoke the procedure from any SQL interface as follows:

```
CALL QSYS.CREATE_SQL_SAMPLE('SAMPLE')
```

Here SAMPLE is the name of the schema that you want to create.

However, currently the sample PHP requires some updates. For example, PASE PHP runs as a CGI-BIN and cannot use the `$_SERVER['PHP_AUTH_USER']` and `$_SERVER['PHP_AUTH_PW']` values.

Also, when connecting to a database, you may normally use something like this example:

```
$dsn = "DRIVER=iSeries Access ODBC Driver;SYSTEM=$isdb_system;DBQ=$isdb_database";
$db = odbc_connect($dsn, $user, $password);
```

But in PASE, you use something like this example:

```
$db = odbc_connect($isdb_system, "", "");
odbc_setoption($db, 1, SQL_ATTR_DBC_DEFAULT_LIB, $isdb_database);
```

Note: It does not matter which user ID and password are used when you connect to the ODBC database. It uses the authority of the user profile that is running the web server process. Use the `ServerUserID` directive in the Apache config to change this. It is actually somewhat of a security hole if you allow others to make a Web page and do not configure the Apache Web server to run the under a different user.

Limitations

Since PHP runs as a CGI application and not as an Apache module, some things will not work in this implementation on the iSeries:

- ▶ HTTP authentication will not work, so any script that tries using the variables `$_SERVER['PHP_AUTH_USER']` and `$_SERVER['PHP_AUTH_PW']` will not work. You need to use user accounts and make a form that gets the user name and password and sets a cookie instead.
- ▶ `PHP_SELF` does not work. There is a bug in the CGI version of PHP 4.3.0 that corrupts the `$_SERVER['PHP_SELF']` variable. For more details on this bug, see PHP's bug page at:

<http://bugs.php.net/bug.php?id=21261>

By the time you read this, that page may have a patch that will fix the issue. If it does, then apply the patch. If it doesn't, use the fix suggested by <mailto:tapken@engter.de> in the bug report:

Create a file called "self_fix.php" in `/QOpenSys/php/lib/php/` with the following script:

```
<?
$_SERVER['SCRIPT_NAME'] = substr($_SERVER['PATH_TRANSLATED'],
                                strlen($_SERVER['DOCUMENT_ROOT']));
$_PHP_SELF = $SCRIPT_NAME = $_SERVER['PHP_SELF'] = $_SERVER['SCRIPT_NAME'];
?>
```

Then in `/QOpenSys/php/etc/php.ini`, look for the line that says:

```
auto_prepend_file =
```

Change this line to:

```
auto_prepend_file = self_fix.php
```

This should fix the `$_SERVER['PHP_SELF']` bug.

PHP 4.2.2 errata

The biggest change from 4.2.2 to 4.3.0 was the configuration process. For this document to apply to 4.2.2, make the following changes to the steps listed previously as noted:

- ▶ For “Locating iSeries specific files” on page 13:
 - **Step 5 on page 14:** Ignore the note about the long line Makefile because it does not exist.
 - **Step 6 on page 14:** PHP version 4.3.0 does not have config_vars.mk. This step is for PHP version 4.2.2 only.
- ▶ For “If you are compiling in AIX on the pSeries” on page 15, follow these steps instead. This is because it does not use PHP itself to try to install PEAR.

```
cd php-4.2.2
make
cd ..
tar -cvf php422pasesrc.tar php-4.2.2
FTP the tar file to PASE
```

The following steps are all done in OS/400 PASE on the iSeries and not in AIX:

```
cd ~
tar -xvf php422pasesrc.tar
cd php-4.2.2
make install
```

The team that wrote this Redpaper

The following team created this IBM Redpaper:

David Larson is a staff software engineer for IBM in Rochester, Minnesota. His career began at IBM enhancing the functionality of OS/400 PASE and assisting groups inside and outside of IBM in using OS/400 PASE. His recent projects include OS/400 PASE integration with the OS/400 JVM, virtual device drivers for OS/400 and Linux, and hypervisor development.

Bryan Logan is a software engineer for IBM Rochester. He is currently on the OS/400 PASE development team. You can contact him by e-mail at: <mailto:bryanlog@us.ibm.com>

Tim Massaro is an advisory programmer for IBM in Rochester, Minnesota. Tim's career includes stints on several S/38 and AS/400 teams, including Work Management, Message Handler, and Operational Assistant. Tim is currently on the DEV/2000 Tools Team. You can reach him by e-mail at: <mailto:tmassaro@us.ibm.com>

Heiko Mueller is a member of the Communications and the OS team for IBM Austria. Prior to joining this team, he worked for a German company delivering support for the iSeries platform. He has passed the certification exams for WebSphere® Solution Designer and iSeries Professional System Operator and spent much of his spare time improving his skills in Java programming, Linux, and AIX. You can reach Heiko by e-mail at: mailto:heiko_mueller@at.ibm.com

Brian R. Smith is a Sr. Consulting I/T Specialist in the IBM International Technical Support Organization (ITSO) Rochester Center. The first third of his career was spent in the Rochester Lab with the design, coding, and testing of the System/38™ and AS/400 in the area of communications. He then jumped the wall into technical marketing support in 1990 to pursue the life of teaching and writing. Brian is the team leader of the iSeries e-business team at the ITSO Rochester Center. You can reach Brian via e-mail at: <mailto:brsmith@us.ibm.com>

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

This document created or updated on April 30, 2003.



Send us your comments in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:
ibm.com/redbooks
- ▶ Send your comments in an Internet note to:
redbook@us.ibm.com
- ▶ Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. JLU Building 107-2
3605 Highway 52N
Rochester, Minnesota 55901-7829 U.S.A.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server™
@server™
Redbooks (logo) ™
ibm.com®
iSeries™
pSeries™

AIX®
AS/400®
DB2 Universal Database™
DB2®
IBM®
Net.Data®

OS/400®
PartnerWorld®
PowerPC®
System/38™
VisualAge®
WebSphere®

The following terms are trademarks of other companies:

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.