Sun Java™ System Web Server 7.0 PHP Add-on Release Notes 1.0

For more up-to-date information, visit the product home page at
http://www.sun.com/software/products/web_srvr/php.xml.

Contents:

1. Overview

PHP: Hypertext Preprocessor (PHP) is a widely used scripting language for creating dynamic web content. These release notes describes how to install and configure PHP with Sun Java System Web Server 7.0 (henceforth referred as "Web Server"). For more information on PHP, see http://www.php.net.

The PHP plug-in has been tested on all platforms supported for Web Server 7.0. For more details, refer to the Sun Java System Web Server 7.0 Release Notes at http://docs.sun.com/app/docs/doc/819-2614/6n4t239c5?a=view.

1.1. Supported Databases

This PHP add-on was compiled with drivers for these databases:

- MySQL 5.0.27

- PostgreSQL 8.1.5

1.2. Configured Options

The set of configured options vary slightly for each platform. To see the definitive configured options, see the phpInfo page. This page can generated by passing the "-i" argument to the php executable.

UNIX/Linux:

```
bin/php -i > phpInfo.html
```

Windows:

```
php-cgi.exe -i > phpInfo.html
```

2. Installing PHP Add-On

1. Change directory to `<web server install dir>/plugins`
2. Unzip the downloaded php bundle here.

   For example:

   ```
   cd /sun/webserver70/plugins
   unzip phppack-solaris-sparc.zip
   ```

3. Configuring PHP on Web Server Using setupPHP Script

Important note: (Windows 2003 only) see Known Issues before proceeding.

This distribution contains a setup script (`setupPHP/setupPHP.bat`) that generates the necessary configuration to execute PHP scripts within Web Server 7.0. By default, this script sets up PHP as a FastCGI plug-in. PHP can also be set up as an NSAPI plug-in by passing "`-sapi=nsapi`" as an argument to `setupPHP`.

Follow these steps to setup PHP on a Web Server instance:

1. Change directory to the directory that contains the PHP files.

   For example (UNIX):

   ```
   /sun/webserver70/plugins/php
   ```

2. Run the setupPHP script:

   UNIX/Linux:

   ```
   ./setupPHP -instancename=<web server instance name>
   ```

   For example:

   ```
   ./setupPHP -instancename=https-mywebserver
   ```

Windows:

```
setupPHP.bat -instancename=<web server instance name>
```

For example:

```
setupPHP.bat -instancename=https-mywebserver
```

3. Pull the instance configuration into the administration server's configuration store. After verifying that the web server instance is able to successfully execute the 'php' scripts, you will need to pull these configuration changes into the administration server's configuration store. Administration CLI command's reference manual (http://docs.sun.com/app/docs/doc/819-3283/6n5h03huv?q=pull-config&a=view) describes the procedure to do this.

4. (Windows only) Restart Windows. `setupPHP.bat` adds the <php install directory> to the system PATH so that web server can load the required dlls. Because the web server runs as a windows service, any change in the system PATH requires a restart to take effect.

4. Configuring PHP on Web Server Manually

Important note: (Windows 2003 only) see Known Issues before proceeding.

You can configure the Web Server to use PHP as a FastCGI plug-in or as an NSAPI plug-in. To manually configure PHP with the Web Server, modify the following configuration files:

- magnus.conf: Contains the NSAPI plug-in initialization directives and settings that control the way NSAPI plug-ins are run
- obj.conf: Contains the HTTP request processing directives
- mime.types: Maps file extensions to MIME types, which enables the server to determine the content type of a request

For more information on the configuration files, refer to http://docs.sun.com/app/docs/doc/819-2630.

4.1. Configuring Web Server to use PHP as FastCGI Plug-in

1. Go to the web server instance's config directory.

   For example (UNIX):

   ```
   /sun/webserver70/https-mywebserver/config
   ```

2. Add the following lines to the `magnus.conf` file:

UNIX/Linux:

```
# Initialize FastCGI plug-in

Init fn="load-modules" shlib="libfastcgi.so"
```

Windows:

```
# Initialize FastCGI plug-in

Init fn="load-modules" shlib="fastcgi.dll"
```

3. Edit the `obj.conf` file to enable Web Server to process a PHP-based request.

   Important note: If there are more than one file with the name “*`obj.conf`”, be sure to edit the correct file. Use wadm's `get-virtual-server-prop` command to get the correct file name. For more information on this command, see [http://docs.sun.com/app/docs/doc/819-3283/6n5h03hsu?a=view.](http://docs.sun.com/app/docs/doc/819-3283/6n5h03hsu?a=view)

   For example:

   ```
   <web server install dir>/bin/wadm get-virtual-server-prop --user=admin --
   config=CONFIG --vs=VS object-file
   ```

   (a) Add the following lines as the first directive of the "default" object in the `obj.conf` file.

   UNIX/Linux:

   ```
   <Object name="default">

      Service type="magnus-internal/php"

      fn="responder-fastcgi"

      bind-path="localhost:3101"

      app-path="<web server install root>/plugins/php/bin/php"

      app-env="PHP_FCGI_CHILDREN=2"

      app-env="PHP_FCGI_MAX_REQUEST=200"

      app-env="FCGI_WEB_SERVER_ADDRS=127.0.0.1"

      app-env="LD_LIBRARY_PATH=<web server install root>/plugins/php"

      app-env="LD_LIBRARY_PATH_64=<web server install root>/plugins/php64"

      ...

   </Object>
   ```

   Windows:

   ```
   <Object name="default">

      Service type="magnus-internal/php"

      fn="responder-fastcgi"

      bind-path=<unique value for each instance. Ex: mywebserver.com>

      app-path="<web server install root>/plugins/php/php-cgi.exe"

      app-env="PHP_FCGI_CHILDREN=2"
   ```

```
        app-env="PHP_FCGI_MAX_REQUEST=200"

        ...

    </Object>
```

(b) Modify the `PathCheck` directive if you want the Web Server to handle the `index.php` file as an index for the document directories.

change

```
PathCheck fn="find-index" index-names="index.html,index.jsp"
```

to

```
PathCheck fn="find-index" index-names="index.html,index.jsp,index.php"
```

4. Add the following line to the `mime.types` file to register the PHP extensions with the Web Server:

```
type=magnus-internal/php exts=php,php3,php4,php5
```

5. (Windows only). Add the <php install directory> to the system PATH so that web server can load the required dlls. Because the web server runs as a windows service, any change in the system PATH requires a restart to take effect.

   For example:

```
PATH=...;C:\Sun\Webserver70\plugins\php;
```

4.2 Configuring Web Server to use PHP as NSAPI Plug-in

To configure the Web Server to use PHP as NSAPI plug-in, perform the following steps:

1. Go to the web server instance's config directory.

   For example (UNIX):

```
/sun/webserver70/https-mywebserver/config
```

2. Add the following lines to the `magnus.conf` file:

   UNIX/Linux:

```
# Initialize PHP v5.x

Init fn="load-modules" shlib="libphp5.so"
funcs="php5_init,php5_close,php5_execute,php5_auth_trans"

Init fn="php5_init" errorString="PHP Initialization Failed!"
```

   Windows:

```
# Initialize PHP v5.x
```

```
Init fn="load-modules" shlib="php5nsapi.dll"
funcs="php5_init,php5_close,php5_execute,php5_auth_trans"

Init fn="php5_init" errorString="PHP Initialization Failed!"
```

3. Edit the `obj.conf` file to enable the Web Server to process a PHP-based request.

   Important note: If there are more than one file with the name "*obj.conf", be sure to edit the correct file. Use wadm's `get-virtual-server-prop` command to get the correct file name. For more information on this command, see http://docs.sun.com/app/docs/doc/819-3283/6n5h03hsu?a=view.

   For example:

   ```
   <web server install dir>/bin/wadm get-virtual-server-prop --user=admin --
   config=CONFIG --vs=VS object-file
   ```

   (a) Add the following lines as the first directive of the "default" object in the `obj.conf` file.

   ```
   <Object name="default">

     Service type="magnus-internal/php" fn="php5_execute"

     ...

   </Object>
   ```

   (b) Modify the `PathCheck` directive if you want the Web Server to handle `index.php` as an index for document directories.

   change

   ```
   PathCheck fn="find-index" index-names="index.html"
   ```

   to

   ```
   PathCheck fn="find-index" index-names="index.html,index.php"
   ```

4. Add the following line to the mime.types file to register the PHP extensions with the Web Server.

   ```
   type=magnus-internal/php exts=php,php3,php4,php5
   ```

5. (Windows only). Add the <php install directory> to the system PATH so that web server can load the required dlls. Because the web server runs as a windows service, any change in the system PATH requires a restart to take effect.

   For example:

   ```
   PATH=...;C:\Sun\Webserver70\plugins\php;
   ```

5. Known Issues

6511133 - PHP does not work in FastCGI mode on Windows 2003.

Work around: Use the NSAPI plug-in.

6. Troubleshooting

This section describes some of the errors that might occur while running PHP with Web Server 7.0.

- When you start the server after configuring PHP with Web Server 7.0, you get the following error messages:

  config: CONF2257: Error parsing file obj.conf, line 51, column 12: Expected "="

  config: CORE3235: File server.xml lines 68-72: Error processing <object-file> element: Error processing file obj.conf

  failure: server initialization failed

Probable cause: Either you have made a spelling error or did not specify the closing tag for php -fcgi objects within the `<Object>` tag.

Solution: Note the line number displayed in the server console or the error logs and modify the `obj.conf` file.

- Your PHP configurations for Web Server 7.0 display the following error message:

  Method Not Allowed

  An error has occurred.

  HTTP2205: The request method is not applicable to the requested resource

Probable causes:

* Web Server 7.0 is loading an `obj.conf` file that contains no PHP configurations.

Solution: Modify the `obj.conf` file in use. Type the following command to verify the file:

```
<Web-Server-install-dir>/bin/wadm get-virtual-server-prop --user=admin --
config=CONFIG --vs=VS object-file
```

* You have not configured FastCGI within the Web Server 7.0 instance.

Solution: Verify the Web Server configuration for FastCGI.

- When you access a PHP file, for example, phpinfo.php, one of the following error messages is displayed:

  ERROR 500: Server Error

  ERROR 500: No Permission

Probable causes:

* Error in the PHP configuration.

Solution: Browse the server error logs.

* The location of the PHP binary as an argument to the app-path variable is incorrect.

Solution: Verify whether:

- The path to the PHP binary as an argument to the app-path variable is valid.

- The PHP binary in question can handle FastCGI, that is, whether the PHP binary can bind to a particular port. Type, for example:

```
/usr/local/php/bin/php -b localhost:3104
```

If the output reports no errors, press Control-C to exit. If an error occurs, your PHP is not FastCGI-compliant.

Also, run "`ldd`" on the PHP binary. Add as an argument the location of all your dependent libraries to LD_LIBRARY_PATH under the app-env variable.

* The PHP binary might not support FastCGI.

Solution: Run the `phpinfo.php` script with PHP to check whether the PHP compilation is in order.

* The port number 3101, which acts as the argument to bind-path might be in use by another process.

Solution: The PHP FastCGI plug-in might be unable to bind to the port number displayed in the error message. Specify another port number as the argument for `bind-path=localhost:portnumber`.

If you compile PHP with options such as --enable-cli, --enable-cgi, and --enable-fastcgi, a separate program, such as php-cgi resides under PHP-installation-dir. In this case, use php-cgi as an argument to app-env in the `obj.conf` file.

- When you access the phpinfo.php file, the following error message is displayed:

  ERROR 500: Server Process Creation Failure

Probable cause: FastCGI cannot create the PHP application for processing the Web page in question.

Solution: Run a system command such as "`ldd`" on PHP and ensure that the appropriate dependent libraries are in place as arguments for the app-env variable for configuring the object file.

- The Web Server 7.0 error logs show the following error:

  ERROR 500: Server Error:

  for host 127.0.0.1 trying to GET /phpinfo.php, func_exec reports: HTTP2302: Function responder-fastcgi aborted the request without setting the status code

Probable cause: You did not compile PHP with support for FastCGI.

Solution: Recompile PHP.

- You have configured PHP within your server instance but when you access the PHP scripts within that instance, the following error message is displayed: "No input file specified"

Probable cause: PHP CGI cannot process the Web page that the browser requested because of the following:

- You have not properly configured the `mime.types` file with the PHP extensions.

- You have not specified either the `NameTrans` or the Service directive in the server configuration's object file to describe how to process files with a PHP extension.

Solution: Check your configuration files and make the required changes.

- You access the MediaWiki URI and the browser displays the following error message: "Wiki has a problem"

Probable causes:

* You have deleted the `LocalSettings.php` file in MediaWiki-install-dir/config before MediaWiki completed the configuration.

Solution: Either reinstall MediaWiki or restore the original `LocalSettings.php` file to the config directory and then reconfigure MediaWiki.

* MediaWiki cannot access the database.

Solution: Verify whether you compiled PHP with the --disable-path-info-check option. If so, recompile without that option.

* The MySQL database is not running.

Solution: Ensure that the MySQL database is running.

- When you access MediaWiki, the browser displays the following error message:

  Error 2048 Not safe to rely on system's timezone settings

Probable cause: The `php.ini` file does not exist or is incomplete.

Solution: Add the following code to your `php.ini` file:

```
[Date]

date.timezone="time-zone";
```

Replace time-zone with the appropriate value, such as PST or GMT, as applicable.

If the php.ini file does not exist, look at the php.ini-recommended file in PHP-source-dir from which you compiled the PHP binaries and then create a php.ini file with the two-line code given above. Place the file in the PHP-install-dir/lib/php directory in which the PHP runtime looks for the php.ini file.

While configuring the `obj.conf` file within Web Server 7.0 for PHP to read `php.ini` from another directory, set up the PHPRC environment variable to point to that directory.

7 Support Information

While this add-on is not officially supported by Sun Microsystems, the Sun web tier online community, which includes web tier product development, support, and service engineers, offers can timely assistance. Visit the forum at http://forum.java.sun.com/forum.jspa?forumID=759.

7.1 How to Report Problems and Provide Feedback

Report problems or feedback to the product team at:

- Online forum: http://forum.java.sun.com/forum.jspa?forumID=759
- Email: webserver@sun.com

8. Tested Applications

While a wide range of PHP applications are expected to work with this distribution, the following PHP applications have been tested and verified:

- phpBB
- MediaWiki
- Gallery
- Squirrel Mail

9. Reference

For additional resources, see the following:

- PHP web site at http://www.php.net
- Technical article at http://developers.sun.com/prodtech/webserver/reference/techart/php.html
- Sun Software Forum at http://forum.java.sun.com/forum.jspa?forumID=759
- Sun Java System Web Server 7.0 Release Notes at http://docs.sun.com/app/docs/doc/819-2614
- Sun Java System Web Server 7.0 Documentation at http://docs.sun.com/app/docs/coll/1308.3