

Installing PHP 8.1.12 for Windows IIS 10

Download the Non-Threadsafe x64 (64-bit) version of PHP 8.1.12 (or latest version) from here <https://windows.php.net/download#php-8.1>

Unblock the Zip File (Properties - Clear Check Box - "This file came from another computer and is blocked . . . etc" Click OK.

Unzip the Zip File to php_8_1

Using Windows file explorer, copy and paste the contents of the "php_8_1" directory (that is, all files and subdirectories) to your new and separate PHP path, such as c:\php_8_1

Using Notepad++ or similar advanced text editor, in the root of the php_8_1 directory, find and open the file "php.ini-production" and save as a new file named "php.ini" without the quotes.

Here is a snapshot of the Windows PHP8 download page which also lists the other requirements such as the Visual C++ Redistributable that must be installed on the Windows server system.

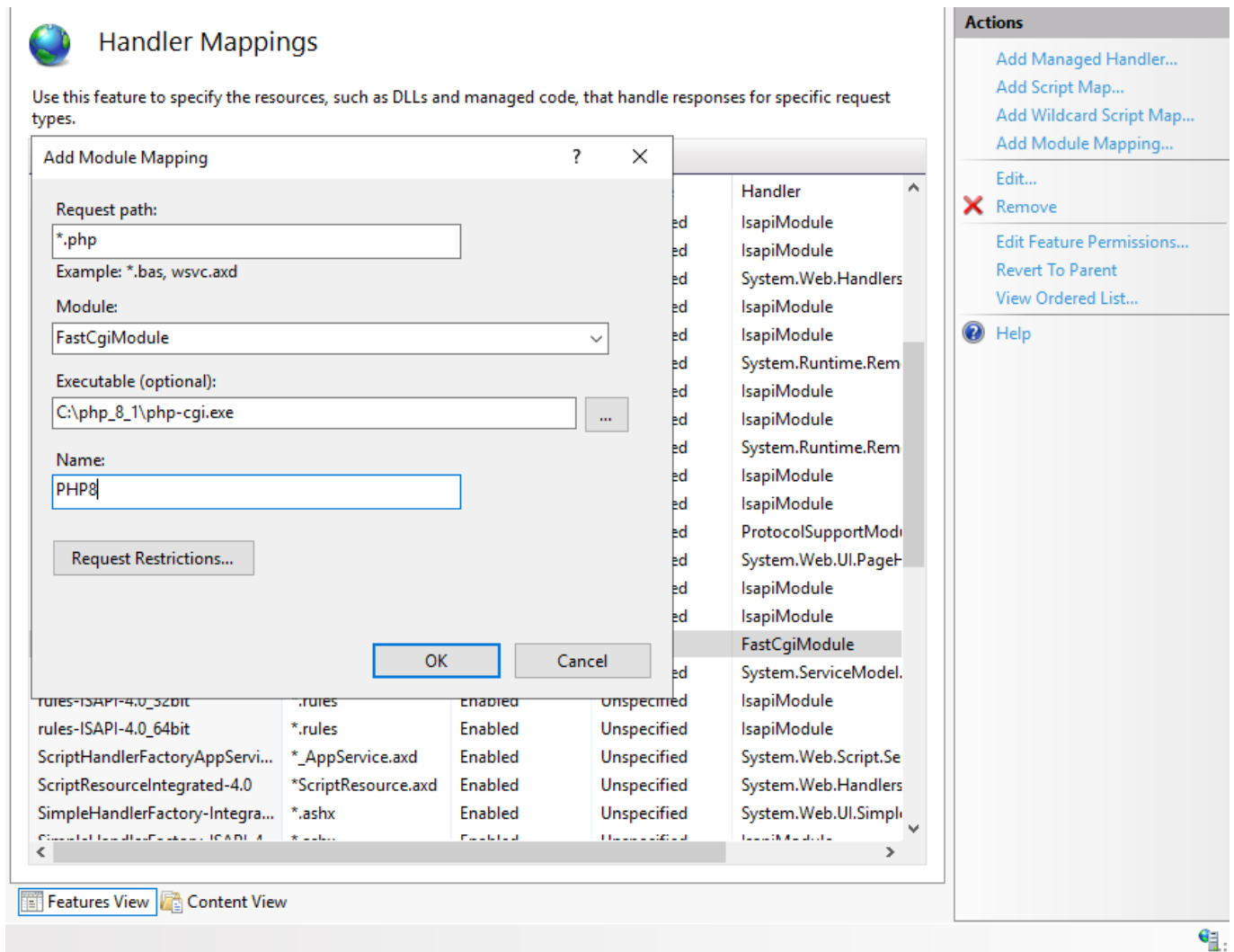
The screenshot shows the 'PHP For Windows' website. The main navigation bar includes links for Home, Downloads, QA Releases, and Snapshots. The left sidebar contains sections for 'PHP For Windows' (describing the site's purpose and linking to a Wiki for building binaries), 'PECL For Windows' (linking to PECL extensions and build directories), and 'Which version do I choose?' (providing instructions for IIS and Apache, and listing VC15 & VS16 builds). The right sidebar, titled 'Binaries and sources Releases', features a dropdown menu to select an option for direct access, a link to 'Past releases', and a section for 'PHP 8.2' which states '8.2 has no release.' Below this, the 'PHP 8.1 (8.1.12)' section lists download links for source code (25.37MB) and tests package (15.15MB). The main content area displays two download options for VS16 x64: 'Non Thread Safe' (dated 2022-Oct-25 19:03:48) and 'Thread Safe' (dated 2022-Oct-25 18:54:50). Each option includes links for Zip, Debug Pack, and Development package (SDK to develop PHP extensions), along with their respective SHA256 hashes.

Make Sure that you have installed on your Windows Server system the Visual Studio C++ Redistributable 2015-2019 X64 (or x86 32-bit as applicable). To Check, in the Windows search (bottom left of screen next to START), type "Control Panel" and Click the hit to open it, and then open "Programs and Features" and then on the left click the link for "Turn Windows Features on or off." Check from the resulting list whether VS C++ 2015-2019 x64 is installed.

Configure Handler Mappings for FastCGI Module in IIS Manager

1. Start IIS Manager by clicking on Start, Programs, Administrative Tools, and then Internet Information Services (IIS) Manager.
2. From the IIS Manager, click on the hostname of your server in the Connections panel on the left to add php mapper handling on a global basis (or instead select the hostname of the particular website if you are adding php mapper handling at the Website Level)

3. Double-click on the Handler Mappings icon. Insert a screen capture here depicting IIS and handler mappings
4. From the Handler Mappings Actions panel, click on Add Module Mapping.
5. Type the following information into the appropriate text boxes, and then click OK. Request path: *.php then Module : FastCgiModule then Executable: C:\php_8\php-cgi.exe then Name: PHP8
6. Click OK, and then click Yes.
7. NOTE: If you are trying to add a new or additional *.php mapping (or remove script access from an existing php version) AT THE WEBSITE LEVEL IN IIS Server and you receive an error message that you don't have the necessary permissions to do so, then try clearing the checkmark next to "read-only" on the first tab of property pages for the file "web.config" in the root of the particular website.



At the Website level, in order to disable the particular version of PHP that you will not be using, you may be able to select the unwanted mapping and click remove at the right, or select the unwanted php version and select edit permissions and disable "script" access for the unwanted mapping.

Configure the php.ini File

1. In the c:\php_8_1 folder, open the php.ini-production file using a good text editor like Notepad++ and then "save as" the contents of this file using the new file name "php.ini" and then close (x) both the php.ini-production file (tab) and the new php.ini file (tab) from within Notepad++.

2. Re-open the php.ini file using Notepad++ text editor and then un-comment and modify certain settings within php.ini as follows:
3. Set `fastcgi.impersonate = 1` ;REMARK FastCGI under IIS supports the ability to impersonate security tokens of the calling client. This allows IIS to define the security context that the request runs under.
4. Set `cgi.fix_pathinfo = 0` ;REMARK The `cgi.fix_pathinfo` provides `PATH_INFO`/`PATH_TRANSLATED` support for Common Gateway Interface (CGI). Setting this to 1 will cause PHP CGI to fix its paths to conform to the specification.
5. Set `cgi.force_redirect = 0`
6. Set `open_basedir` = a path pointing to a folder or network path where the content of the Web site(s) is located. (Such as `c:\hostdata`), or just leave it commented like this `”;open_basedir =` without the quotes
7. Set `extension_dir` to point to a location where PHP extensions reside. This is typically `extension_dir = ".\ext"` However, per remarks in the “php.ini-production” file in both PHP7 and PHP8 it says for “Windows” set it instead as follows: `extension_dir = “ext”`
8. Set `error_log=“C:\php_errors.log”` ;REMARK This can help with troubleshooting. Alternatively, set **`error_log = “c:\php\logs\errors_log”`** and make a new sub-folder in `c:\php` named “logs” and grant the sub-folder “logs” full control write access for both IUSR and IIS_USERS.
9. Enable the required PHP extension by un-commenting the appropriate lines within the extensions section of the php.ini file. See a further discussion of which extensions to enable.

Enable Extensions in the php.ini file for PHP 8

Using Notepad++ or similar text editor, find and edit the php.ini file to enable extensions. Make sure that the applicable/corresponding `php_*.dll` file for the extension that you wish to activate is still available within the latest `php8/ext/` directory

See the following screen snippet for examples of extensions to enable:

```
929 extension=bz2
930 ; REM ENABLED curl ON PHP8 INSTALLATION
931 extension=curl
932 ;extension=ffi
933 ; REM ENABLED ftp ON PHP8 INSTALLATION
934 extension=ftp
935 ; REM ENABLED fileinfo ON PHP8 INSTALLATION
936 extension=fileinfo
937 ; REM ENABLED gd ON PHP8 INSTALLATION
938 extension=gd
939 ;extension=gettext
940 ;extension=gmp
941 ; REM ENABLED php_intl.dll ON INSTALLATION
942 extension=intl
943 ; REM ENABLED imap ON PHP8 INSTALLATION
944 extension=imap
945 ;extension=ldap
946 ; REM ENABLED mbstring ON PHP8 INSTALLATION
947 extension=mbstring
948 ;extension=exif ; Must be after mbstring
949 ; REM ENABLED mysqli on PHP8 INSTALLATION
950 extension=mysqli
951 ;extension=oci8_12c ; Use with Oracle 12c
952 ;extension=oci8_19 ; Use with Oracle 19c
953 ;extension=odbc
954 ; REM ENABLED openssl ON PHP8 INSTALLATION
955 extension=openssl
956 ;extension=pdo_firebird
957 ; REM ENABLED pdo_mysql ON PHP8 INSTALLATION
958 extension=pdo_mysql
959 ;extension=pdo_oci
960 ;extension=pdo_odbc
961 ; REM ENABLED pdo_pgsql ON PHP8 INSTALLATION
962 extension=pdo_pgsql
963 ;extension=pdo_sqlite
964 ;extension=pgsql
965 ;extension=shmop
966
967 ; The MIBS data available in the PHP distribution
968 ; See https://www.php.net/manual/en/snmp
969 ;extension=snmp
970
971 ; REM ENABLED soap ON PHP8 INSTALLATION
972 extension=soap
973 ;extension=sockets
974 ;extension=sodium
975 ;extension=sqlite3
976 ;extension=tidy
977 ; REM ENABLED xsl ON PHP8 INSTALLATION
978 extension=xsl
```

Enable File Uploads in php.ini file for PHP 8

Use a file editor like Notepad++ to edit the php.ini file.

By default file uploads are enabled.

file_uploads = On

See the following screen snippet for examples of configuring php.ini for file uploads:

```
846 ;;;;;;;;;;;
847 ; File Uploads ;
848 ;;;;;;;;;;;
849
850 ; Whether to allow HTTP file uploads.
851 ; https://php.net/file-uploads
852 file_uploads = On
853
854 ; Temporary directory for HTTP uploaded files (will use system default if not
855 ; specified).
856 ; https://php.net/upload-tmp-dir
857 upload_tmp_dir =
858 ; REM CONFIGURED PHP 8 UPLOADS DIRECTORY AND ADDED \temp\uploads FOLDERS AND ENABLED IUSR WRITE PERMISSIONS ON "uploads" DIRECTORY
859 upload_tmp_dir="c:\php_8_1\temp\uploads"
860
861 ; Maximum allowed size for uploaded files.
862 ; https://php.net/upload-max-filesize
863 ; REM INSERTED ";" TO DISABLE NEXT LINE WITH 2M LIMIT AND INSTEAD INSERTED 750M LIMIT ON SECOND LINE
864 ; upload_max_filesize = 2M
865 upload_max_filesize = 750M
866
867 ; Maximum number of files that can be uploaded via a single request
868 ; REM DISABLED 20 ON NEXT LINE AND INSERTED AND ENABLED 60 SECOND LINE
869 ;max_file_uploads = 20
870 max_file_uploads = 60
871
```

Enable Sessions Save Path to New TMP Directory

See the following screen snippet.

```
1391 ; REM ENABLED 2ND LINE -SESSIONS NEED TO WORK FOR PHPMYADMIN, CREATE php_8_1/TMP FOLDER
1392 ; REM (cont) AND GRANT MODIFY AND WRITE PERMISSIONS TO IIS_IUSERS GROUP AND OPTIONALLY IUSR
1393 session.save_path = "c:/php_8_1/tmp"
1394
```

Matters to Consider with PHP 8

- Do we need to install PEAR any longer? <https://pear.php.net/package/PEAR/>
<https://pear.php.net/package/PEAR/download>
- Do we need to install and enable an extension for XMLRPC any longer? Is it deprecated or just no longer located in directory /ext/ as a dll extension, and no longer listed in the extensions section of php.ini?

From:

<https://www.installconfig.com/> - Install Config Wiki

Permanent link:

https://www.installconfig.com/doku.php?id=installing_php_8_1_12_windows_iis_10

Last update: 2022/10/30 00:07

