Using Director with Apache

Table of Contents

Chapter 1. Files

This section describes the files you use with Apache.

Apache extension module

The Apache extension module is mod_enhydra_director.so. It is in the Apache libexec directory and is loaded with the AddModule and LoadModule directives in httpd.conf. It does the work of forwarding application requests to the Enhydra server and of forwarding the server response to the client.

Enhydra Director Configuration File

This file, enhydra_director.conf, contains the configuration for Enhydra Director. By default, the Apache extension module described above looks for this file in the same directory as the httpd.conf file.

The Enhydra Director Daemon

This program, edir_daemon, runs in the background and monitors the state of the shared memory region that is used by child instances of the module to coordinate load balancing. This is needed in Apache because it uses many separate processes to allow multithreading of simultaneous requests. The daemon is automatically launched when Apache starts, and automatically cleans up the shared memory and exits when Apache is stopped.

The Enhydra Director Status Utility

This utility, edir_status, directly reads the shared memory region used by the Enhydra Director module. It reports the current state of each configured application and Enhydra server.

Enhydra Director Scoreboard

The scoreboard, enhydra_director.ipc, contains the shared memory data used by all of the apache child processes when they are handling Enhydra Director requests. Normally, the contents of this file are kept in memory as an mmap( ) based shared memory region. This file also serves as the scoreboard lock file, using fcntl() based locking. Its normal home ins in edir subdirectory of the Apache logs directory. It is normally only accessible to the Apache child user id.

Apache Error Log

The standard error log for the apache server is typically called error_log. Enhydra Director handler logs all errors encountered while handling requests to this file. Certain startup messages are also logged to this file, however the Enhydra Director Daemon logs most startup errors to the system log.

System Log

The system log is typically in /var/log/messages or /var/adm/messages. If the daemon encounters errors while it is initializing it has no access to the Apache error log, so it logs the error to the system log using the 'daemon' logging channel.

Chapter 2. Procedure for Solaris and Linux

These instructions assume that you will install the Enhydra Director files into your Apache installation directories. To build, you must first obtain, build, and install Apache 1.3.9 or later. You must configure Apache so that DSO support (mod_dso) is enabled. The build scripts use the Apache 'apxs' utility to automatically obtain the correct installation directories for the Enhydra Director components, so you should take not of where this utility is installed. Finally, you should make sure that Perl 5 is installed, since apxs is a perl script.

Step 1: Get Enhydra Director source code

The Enhydra Director source code is included with the full Enhydra source distribution. A smaller source distribution containing only the source code of Enhydra Director is included with the pre-built distributions of Enhydra 3.0. Obtain the Enhydra Director source code from either of the above sources and extract it into a working directory.

if you are using the Enhydra full source tree: Extract 'enhydra-src3.0.tar.gz' into the current directory. You will find the Enhydra Director source in:

```
Enhydra/modules/EnhydraDirector/src/
```

if you are using the Enhydra pre-built distribution: Extract 'lutris-enhydra3.0.tar.gz' or 'lutris-enhydra3.0.zip'. The Enhydra Director source archive is in: ./director/enhydra-director1.0.tar.gz or ./director/enhydra-director1.0.zip Change directory to ./director and extract one of the above archives into the current directory.

The extracted source code will be in: ./enhydra-director1.0 From where you extracted the Enhydra distribution: ./director/enhydra-director1.0

Now, change directory to the newly extracted Enhydra Director source code directory root. This is the directory named 'src' that contains the subdirectories 'common', 'apache', 'nsapi, etc. From this point on we will call this directory '$SRCROOT'

Step 2: Build the Enhydra Director Apache module

If you installed Apache in the default location on either Solaris or Linux, this step should be easy. Just run the './configure' script with no arguments:

```
cd $SRCROOT/apache ./configure make make install
```

This will build the Enhydra Director Apache module and install the 'mod_enhydra_director.so' file into the 'libexec' directory for your Apache distribution.

Option for 'configure': --with-apxs=/full/path/to/program/apxs This option allows the location of the Apache 'apxs' script to be specified if the default configure script cannot find it.

Option for 'configure': --enable-debugging Adds debugging messages to the build. This allows the 'EnhydraDirectorDebug' option to be specified in 'httpd.conf', or <Options debug="0xNNNNNN"/> to be specified in 'enhydra_director.conf' Enabling debugging can be useful if you experience problems with setting up the module.

After running 'make install', you will find 'mod_enhydra_director.so' in your Apache installation's 'libexec' directory. You will also find the file 'enhydra_director.conf.default' in the 'conf' directory.

Note: If you like doing things the hard way: Install the 'edir_daemon', 'edir_status', and 'enhydra_director.conf' files wherever you like. Install the 'mod_enhydra_director.so' file in the 'libexec' directory for your Apache installation. The 'libexec' directory may not be named 'libexec' on all installations. It's much easier to just './configure' and 'make install', using the --with-apxs if necessary to locate the 'apxs' script.

Use the following 'httpd.conf' options to specify the locations of Enhydra Director configuration and data files.

- EnhydraDirectorConfigFile Specifies the full path to the 'enhydra_director.conf' file. By default this is 'enhydra_director.conf' in the 'conf' subdirectory of your Apache installation.

- EnhydraDirectorLockFile Specifies the location of the file used for 'locking' the scoreboard during shared access. If you set this parameter, you should not need to set the 'EnhydraDirectorData' option, unless you are a developer or are porting to a strange platform, in which case you really know what you are doing. Setting this parameter may be necessary if you installed EnhydraDirector on an NFS filesystem. Due to locking bugs in many/ most NFS implementations, EnhydraDirector will not work if the lock file is located on an NFS file system. By default, the lock file is located under the 'logs' directory. In order to allow the load balancing code to run as the Apache 'Child' user ID, the actual lock file is located under 'logs/edir', where 'edir' has permissions set to allow the child processes to read and write. The permissions of the 'logs' directory are unaffected.

- EnhydraDirectorDataFile (Expert - Developers and Geeks Only) You should not ever need to set this parameter unless you are a developer or are experimenting with alternate shared memory modules (see edir_shmem.h). This allows the scoreboard shared data file to be specified as a different file than the lock file, or even as a Systme V IPC region. By default, the 'mmap' shared memory module uses the same file as the 'filesys' mutex module. (see edir_mutex.h). This means the locking and data sharing use the same file on most modern unix implementations, including Solaris, Linux, and FreeBSD.

- EnhydraDirectorDaemonPath (Advanced) if you must install the Enhydra Director daemon (edir_daemon) in a different location than the Apache 'bin' directory, this option can be used to specify the full path to the daemon program.

- EnhydraDirectorDebug (Advanced) if you used '--enable-debugging' when building Enhydra Director, then this option can be used to set the debugging mask in the module. The debug mask is also propagated to the daemon when it is started. See edir_debug.h or enhydra_director.conf.default for more information on possible values. To enable full, highly voluminous debugging, including hexdups of all communication between the module and Enhydra, specify the value '0xffffffff'. Be forewarned that doing this will severely limit Enhydra director's performance, and will dump hundreds of megabytes of logging data to the Apache error log. This option is identical to <option debug="0xNNNNNNNN"/> in enhydra_director.conf, except that it gets set much earlier, and allows debugging of the code that loads the config file and scoreboard, as well as the daemon launching code.

Chapter 3. Configuring Enhydra Director on Solaris and Unix

Step 1:

Find out where 'httpd.conf' is located for your Apache installation. Usually this is something like '/usr/local/apache/conf/httpd.conf'. However, Apache is very flexible in how it can be installed, so this may differ. The build scripts use 'apxs' to automatically install the compiled module and programs into the recommended locations.

Step 2:

Determine where Apache stores its DSO modules. This is usually 'libexec' under the Apache installation root. For example, on many Apache installations, the DSO module directory is '/usr/local/apache/libexec'. Take note of the 'libexec' part. This may be something else like 'sbin' or 'modules'. You will need this when configuring the 'LoadModule' line in 'httpd.conf'.

Step 3:

Add the Enhydra Director module to the Apache configuration.

- Open 'httpd.conf' in a text editor.

- Search to the last line in the file starting with 'LoadModule'. (The default 'httpd.conf' should at least have a comment line, but if not even a commented LoadModule line is found, just go to the end of the file.)

- Using the last component of the 'libexec' directory located in step 2, add a line to the list of LoadModules directives (if any), that looks like:

  ```
  LoadModule enhydra_director_module libexec/mod_enhydra_director.so
  ```

- If the last component of you Apache installation's 'libexec' path ends in something else (say, 'modules'), then you would instead add the line:

  ```
  LoadModule enhydra_director_module modules/mod_enhydra_director.so
  ```

- If there are other 'LoadModule' lines, they usually will contain the correct path to the libexec path, so you can use the other directives as a guide. Note: If you are using 'mod_rewrite', then you should make sure that the 'LoadModule rewrite_module ...' directive comes BEFORE the 'LoadModule enhydra_director_module ...' directive if you intend to use 'mod_rewrite' to redirect URLs to Enhydra Director. Also you should use the 'mod_rewrite' '[PT]' tag to force pass-through of redirected URLs.

- Now, search for the last line starting with 'AddModule'. Again, if such a line does not even exist as a commented-out line, then just go to the end of the file. Add the line:

  ```
  AddModule mod_enhydra_director.c
  ```

- Add any desired Enhydra-specific directives such as 'EnhydraDirectorDebug' to the 'httpd.conf' file. These directives are optional in almost all cases.

- Save the 'httpd.conf' file.

Step 4:

Create the Enhydra Director configuration file. The Enhydra Director configuration file for Apache is normally located in the same directory as 'httpd.conf'. However, you can specify the 'EnhydraDirectorConfigFile' directive in httpd.conf to specify a different file.

The full description of the syntax and all of the directives available in the configuration file is beyond the scope of this document. However, the file 'enhydra_director.conf.default' contains several useful examples that can be easily copied to match your site setup. The comments within the default file also describe most of the available options.

An important thing to note is that the configuration file is based on an XML DTD (EnhydraDirectorConfig.dtd) and is strictly validated. This means the the file MUST contain valid XML, and it MUST follow the format defined in the DTD. Case is important in all section and property names, and order is important among different sections.

Step 5:

Use the 'apachectl' script, or your own script, to restart the apache server. The easy way is 'apachectl restart'.

Step 6:

- Use 'ps -aef' or 'ps -ax' (depending on your system) to make sure 'edir_daemon' is running.

- In the Apache logs directory, there should be a subdirectory called 'edir' that should contain the file 'enhydra_director.ipc'. This file is the lock and scoreboard file.

- In the Apache 'bin' directory, where 'httpd' is located, run 'edir_status'. This should dump out the configuration in 'enhydra_director.conf' If this succeeds, Enhydra Director is running.

- If errors occur withing the 'edir_daemon' process, the errors are logged to the system log using 'syslog()'. The 'daemon' channel is used. Normally the messages go to the 'messages' file, which is '/var/adm/messages' on Solaris, and '/var/log/messages' on Linux.

- If initialization errors occur within the module, the errors are reported in the Apache error log.

Step 7:

- Start up a web browser.

- If you have configured a <Status> section in the configuration, you can connect to its URL prefix to ensure that the handler is functioning. See the default configuration file for more information on the <Status> section.

- Connect to one of your configured application prefixes.

- Errors encountered while handling requests are reported in the Apache error log.

Chapter 4. Procedure for Windows NT

These instructions assume that you will install the EnhydraDirector files into your Apache installation directories. In order to build, you must first obtain, build, and install Apache 1.3.9 or later. Moreover, you must configure Apache so that DSO support (mod_dso) is enabled. Visual C++ 6.0 is required in order to build the module.

Step 1: Get Enhydra Director source code

Obtain the Enhydra Director source code or download precompiled binaries.

Step 2: Build the Enhydra Director Apache module

To build the Win32 version of the Apache module, start Visual Studio,and open the Enhydra Director Apache workspace: <Director>/src/apacheWin32/ApacheDirectorWin32.dsw Set 'EnhydraDirectorApaWin32' as the active project, then set the active configuration to 'Debug' or 'Release' according to your needs. If you installed Apache server in a non-default location, you also need to edit the build settings (C Preprocessor) to set the correct 'include' directories for the apache header files. At this point, just do a 'Build All' and rebuild the .SO. In the same directory as the workspace, there is a copy of 'enhydra_director.conf.default'. The newly built .SO will be found in the 'Debug' or 'Release' subdirectory, according to the usual Visual Studio convention.

Step 3: Install the Enhydra Director files. (Win32)

Regardless of how you obtain the final .so and files, install: <Apache>\modules\mod_enhydra_director.so <Apache>\conf\enhydra_director.conf.default

Chapter 5. Configuring Enhydra Director on Windows NT

Step 1: Locate the 'httpd.conf' file

Find out where 'httpd.conf' is located for your Apache installation. Usually this is something like 'C:\Program Files\Apache Group\conf\httpd.conf'. However, Apache is very flexible in how it can be installed, so this may differ.

Step 2: Determine where Apache stores its DSO modules

This is usually 'modules' under the Apache installation root. For example, on many Apache installations, the DSO module directory is 'C:\Program Files\Apache Group\Apache\modules'.

Step 3: Add the Enhydra Director module to the Apache configuration

- Open 'httpd.conf' with your favorite editor.

- Search to the last line in the file starting with 'LoadModule'. (The default 'httpd.conf' should at least have a comment line, but if not even a commented LoadModule line is found, just go to the end of the file.)

- Using the last component of the 'modules' directory located in step 2, add a line to the list of LoadModules directives (if any), that looks like:

```
LoadModule enhydra_director_module modules/mod_enhydra_director.so
```

If the last component of you Apache installation's 'modules' path ends in something else (say, 'mod'), then you would instead add the line:

```
LoadModule enhydra_director_module mod/mod_enhydra_director.so
```

If there are other 'LoadModule' lines, they usually will contain the correct path to the modules path, so you can use the other directives as a guide.

NOTE: If you are using 'mod_rewrite', then you should make sure that the 'LoadModule rewrite_module ...' directive comes BEFORE the 'LoadModule enhydra_director_module ...' directive if you intend to use 'mod_rewrite' to redirect URLs to Enhydra Director. Also you should use the 'mod_rewrite' '[PT]' tag to force pass-through of redirected URLs.

- Now, search for the last line starting with 'AddModule'. Again, if such line does not even exist as a commented-out line, then just go to the end of the file.

- Add the line:

```
AddModule mod_enhydra_director.c
```

- Add any desired Enhydra-specific directives such as 'EnhydraDirectorDebug' to the 'httpd.conf' file. These directives are optional in almost all cases.

- Save the 'httpd.conf' file. Enhydra Director is now configured!

Step 4: Create the Enhydra Director configuration file

The Enhydra Director configuration file for Apache is normally located in the same directory as 'httpd.conf'. However, you can specify the 'EnhydraDirectorConfigFile' directive in httpd.conf to specify a different file.

The full description of the syntax and all of the directives available in the configuration file is beyond the scope of this document. However, the file 'enhydra_director.conf.default' contains several useful examples that can be easily copied to match your site setup. The comments within the default file also describe most of the available options.

An important thing to note is that the configuration file is based on an XML DTD (EnhydraDirectorConfig.dtd) and is strictly validated. This means the the file MUST contain valid XML, and it MUST follow the format defined in the DTD. Case is important in all section and property names, and order is important among different sections.

Step 5: Restart the Apache server

Step 6: Verify correct startup

- Use 'Test Configuration' from Apache Program Group.

Step 7: Verify connectivity

- Start up a web browser.

- If you have configured a <Status> section in the configuration, you can connect to its URL prefix to ensure that the handler is functioning. See the default configuration file for more information on the lt;Status> section.

- Connect to one of your configured application prefixes.

- Errors encountered while handling requests are reported in the Apache error log.

Chapter 6. Instructions for compiling Enhydra Director for Apache/Win32

Step 1: Open EnhydraDirectorApaWin32 project in Visual C++ 6.0

Step 2: Add following directories in project (Tools/Options/Directories):

```
..\common
```

```
..\
```

```
<Apache Dir>\include
```

Step 3: Add libraries to the project:

```
<Apache Dir>\libexec\ApacheCore.lib
```

Step 4: WS2_32.lib (usualy under Microsoft Visual Studio\VC98\Lib)

Step 5: Build project

Step 6: Under Debug and Release dir are precompiled versions of mod_enhydra_director.so

Chapter 7. Troubleshooting

Problem: The apache extension fails to load when Apache is started.

- Check that the 'httpd.conf' for correct configuration of the module, as described in the previous sections.

- Check the system log file for errors reported by the edir_daemon program.

- Check the Apache error log for errors reported by the Enhydra Director module.

- Ensure that the enhydra_director.conf file exists and contains valid and syntactically correct configuration data. If the syntax is not valid, Enhydra Director will report the specific errors and line numbers to the Apache error log.

Enhydra Director starts up OK but I can't connect to my application.

- If you configured the <Status> section, check the status prefix for connection failures to your back-end Enhydra servers.

- Make sure that the prefixes configured in enhydra_director.conf match the prefixes configured for your application on the Enhydra server. The prefixes MUST match EXACTLY, including case! Trailing slashes in the prefix are not significant and are ignored.

- Make sure that the application servers and ports in the enhydra_director.conf file refer to valid and running instances of your application. - Make sure your application is running on the Enhydra server.

- You application must use the 'EnhydraDirector' connection method on ports that the Apache Enhydra Director extension will dispatch requests to.

- It is highly recommended to configure the <Status> section in the configuration file so you can examine the run-time state and configuration of the handler.

Problem: Apache child processes dump core. (crash)

- This is evidence of an error in the handler or filter DLL that needs the attention of the Together company.

- There's also the possibility that there is a bug elsewhere in Apache, or in some other module. If possible, remove all unneeded modules from the 'httpd.conf' file to try to narrow the problem down to the Enhydra module.

- If you're an engineer and can give us stack traces from gdb, we would be VERY appreciative. ESPECIALLY stack traces from '--enable-debugging' compilations. :)

- Please report any bugs of this nature to the Enhydra.org folks, at www.enhydra.org. The more details you can provide about what led to the failure, the better. Also, machine state information and configuration information is helpful.

- If you have a way to reproduce the bug consistently, Please include the necessary steps to do so.