

USER'S GUIDE

# Abyss Web Server™ X1

For Windows®





## **Abyss Web Server X1 For Windows User's Guide**

Copyright © 2001, 2002, 2003 by Aprelium Technologies

### Revision History

Revision 1.0.0 February 20, 2002

Revision 1.0.3 May 10, 2002

Revision 1.0.7 July 1, 2002

Revision 1.1.0 October 3, 2002

Revision 1.1.6 July 7, 2003

Revision 1.2.0 b1 October 18, 2003

Revision 1.2.0 b2 November 10, 2003

### **DISCLAIMER OF WARRANTIES**

Aprelium Technologies makes no representations or warranties, either express or implied, by or with respect to anything in this guide, and shall not be liable for any warranties of merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

### **LICENSE NOTES**

No part of this guide may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, photocopying, recording, or otherwise, without prior written consent of Aprelium Technologies.

No patent liability is assumed with respect to the use of the information contained herein.

While every precaution has been taken in the preparation of this guide, Aprelium Technologies assumes no responsibility for errors or omissions. This guide and features described herein are subject to change without notice.

### **TRADEMARKS**

Aprelium and The Aprelium logo are registered trademarks and Abyss Web Server is a trademark of Aprelium Technologies.

Other products or brand names are trademarks or registered trademarks of their respective holders.

### **CONTACT INFORMATION**

<a href="http://www.aprelium.com">http://www.aprelium.com</a>	Web Site
<a href="mailto:support@aprelium.com">support@aprelium.com</a>	Technical support
<a href="mailto:suggest@aprelium.com">suggest@aprelium.com</a>	Product enhancement suggestions
<a href="mailto:doc@aprelium.com">doc@aprelium.com</a>	Documentation error reports
<a href="mailto:contact@aprelium.com">contact@aprelium.com</a>	General inquiries



# Table of Contents

<b>Before You Begin</b> .....	<b>7</b>
System requirements .....	7
Alternative formats.....	7
What you should know .....	7
Chapter quick reference .....	7
Conventions used in this guide .....	8
Typographical conventions .....	8
Terminology .....	8
<b>1. Introduction</b> .....	<b>11</b>
What is Abyss Web Server? .....	11
How do web servers work?.....	11
<b>2. Getting Started</b> .....	<b>13</b>
Getting the software .....	13
Installing/Upgrading the software .....	13
First contact.....	14
Setting up a web site.....	15
Shutting down the server.....	16
Asking for support.....	16
<b>3. Using The Console</b> .....	<b>17</b>
What is the console? .....	17
Accessing the console .....	17
Console interface basics .....	17
Help buttons .....	18
Tables .....	18
Applying configuration changes .....	18
Console Configuration .....	19
Changing the console port.....	19
Changing the console access credentials .....	20
<b>4. Server Management</b> .....	<b>23</b>
Overview .....	23
Server status.....	23
Server access statistics .....	24
General server configuration.....	25
Advanced server configuration .....	26
Server parameters .....	26
Index files .....	27

MIME types.....	28
Aliases.....	29
Custom error pages.....	30
CGI parameters .....	31
SSI parameters.....	34
Users and Groups .....	35
Access Control.....	35
<b>5. CGI Scripts .....</b>	<b>39</b>
Setting up an interpreter .....	39
How are they run? .....	39
CGI environment variables.....	40
<b>6. Server Side Includes.....</b>	<b>43</b>
What are SSI? .....	43
SSI Syntax.....	43
Directives.....	43
#config.....	43
#echo .....	45
#exec.....	45
#flastmod.....	46
#fsize .....	46
#include .....	46
<b>Glossary .....</b>	<b>49</b>
<b>A. Configuration File Format.....</b>	<b>53</b>
<b>B. Patterns Format .....</b>	<b>57</b>
<b>C. Uninstalling The Software.....</b>	<b>59</b>
<b>D. Solving Ports Listening Problems.....</b>	<b>61</b>
<b>E. Troubleshooting Guide .....</b>	<b>63</b>

# Before You Begin

## System requirements

Abyss Web Server X1 is a multiplatform software designed for Microsoft Windows 95, 98, Millennium Edition (ME), NT 3.51 (and higher), 2000, XP, 2003, MacOS X 10.2 (and higher), Linux 2.0 (and higher), and FreeBSD 4.8 (and higher) systems.

This guide documents the Windows version. Abyss Web Server X1 for Windows minimum requirements are:

- An Intel 80386 processor at 33 MHz.
- 8 Mb of RAM.
- A web browser with [HTML 4.0](#) support.
- [TCP/IP](#) protocol stack installed.

Although Abyss Web Server can run on a standalone computer, it is recommended to have a Modem or a network card and to be connected to a LAN (Local Area Network) or to the Internet.

## Alternative formats

This guide exists in several formats: Online HTML, Offline HTML, and PDF. For more information, browse the documentation section at <http://www.aprelium.com/abyssws>.

## What you should know

Because it is impossible to provide highly detailed information for every given instruction in this guide, you are encouraged to read the documentation accompanying your operating system if you are not already familiar with it. This guide assumes also that you master the basics of web browsing.

## Chapter quick reference

This guide documents the features and capabilities of Abyss Web Server. It can guide you to quickly master the basics and move on to use the advanced features.

If you are new to web servers, go to Chapter 1, "[Introduction](#)," to get a quick overview.

If you want to get started right away, go to Chapter 2, "[Getting Started](#)," for step by step instruction on installing and running the software.

Chapter 3, "[Using the Console](#)," describes the web configuration interface of Abyss Web Server.

Use Chapter 4, "[Server Management](#)," to understand how to configure and manage your web server.

See Chapter 5, "[CGI Scripts](#)," for information on using CGI scripts with Abyss Web Server and making PHP or Perl scripts work.

Read Chapter 6, "[Server Side Includes](#)," for the complete reference on SSI.

Refer to the "[Glossary](#)" for definitions of key terms and concepts used in this guide.

## Conventions used in this guide

### Typographical conventions

#### UI Object

Refers to user interface objects such as buttons, icons and text fields.

#### Computer I/O

Used for computer input and output such as URLs, filenames and commands.

#### <Variable>

Indicates a variable. You have to substitute it with its real value.

## **Terminology**

Your computer

The computer on which Abyss Web Server is installed.

The server

The instance of Abyss Web Server under execution.

The browser

The web browser you use.

The console

The web configuration interface of Abyss Web Server.

The network

The network to which your computer is connected. It can be:

- A single computer network if you are not physically connected to any network and your computer is alone.
- A LAN (Local Area Network).
- The Internet

*Before You Begin*

# Chapter 1. Introduction

## What is Abyss Web Server?

Abyss Web Server turns your personal computer in a full-featured web server. People all over the Internet can view documents, download graphics, listen to music and even view movies hosted on it. Abyss Web Server was designed with both novice and experimented users in mind. That's why it is easy to use and incredibly powerful.

## How do web servers work?

What happens when you enter in the address field of your browser the [URL](http://www.aprelium.com/doc/sample.html) `http://www.aprelium.com/doc/sample.html?`

First, the browser slices the URL in 3 parts:

- `http://`: This part indicates that the document you want to access can be retrieved from web server, which understands the [HTTP](#) protocol. The HTTP protocol is a standardized language of communication between browsers and web servers.
- `www.aprelium.com`: This is the host name of the computer from which the document can be downloaded.
- `/doc/sample.html`: This is the virtual path of the document in the `www.aprelium.com`'s web server.

Then, the browser establishes a connection channel with the web server on the computer which host name is `www.aprelium.com` and requests the document which virtual path is `doc/sample.html`. The server decodes the request and maps the virtual path to a real one, which should match an existing file. The server sends the file to the browser with some useful information such as its last modification time and its MIME type. The MIME type helps the browser deciding how to display the received document. In our example, it is a HTML file. So the server sets its MIME type to `text/html` and the browser understands that it must render it as text.

Sometimes you enter a URL without an explicit filename such as `http://www.aprelium.com/doc`. The browser sends the request to the web server as in the previous example. The server detects that the virtual

path maps to a directory and not to a file. It searches then in this directory an index file. Index files are usually named **index.html** or **index.htm**. If it finds for example **index.html**, it acts as if the requested URL was **http://www.aprelium.com/doc/index.html**. If no index file is found, the web server generates a listing of the directory contents and sends it to the browser or reports an error.

# Chapter 2. Getting Started

## Getting the software

If you already have the Abyss Web Server installation package, skip this section.

To download Abyss Web Server, browse the download section at <http://www.aprelium.com/abyssws>. Follow the provided instructions and choose the correct version of the software suitable to your computer and to the operating system you are using. Once done, the browser displays a dialog box asking where to save the downloaded file. Choose a directory and validate.

## Installing/Upgrading the software

To setup Abyss Web Server:

- Open the directory where you saved the software package.
- Double-click on the software package icon.
- If an old version of Abyss Web Server is already installed on your computer, you may be asked to uninstall it. All you have to do is to follow the on-screen instructions. *IMPORTANT NOTICE: When the uninstaller asks you to confirm the deletion of the installation directory, select **No**.*
- Read carefully the license agreement. If you agree with it, press **I Agree**. If you do not, press **Cancel** and delete Abyss Web Server package from your computer.
- Deselect components you do not want to install. **Auto Start** enables Abyss Web Server auto starting when a Windows session starts. **Start Menu Shortcuts** enables adding Abyss Web Server shortcuts in the **Start Menu**. **Documentation** installs help files.
- Press **Next**.
- Choose a directory where you want to install Abyss Web Server files. From now on, **<Abyss Web Server directory>** will refer to this directory.
- Press **Install**.

After installing, you will be asked if you want to launch Abyss Web Server. If you press **No**, you should press **Close** to close the setup program.

## First contact

If Abyss Web Server is not running, open the **Start** menu, choose **Programs**, then **Abyss Web Server** and select **Abyss Web Server**. You can also open the directory where you have installed it and double-click on the **abyssws.exe** icon.



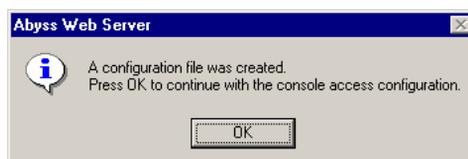
**Figure 2-1. Abyss Web Server main window**

Abyss Web Server inserts a small icon in the system tray. The icon represents a globe that spins when the server is accessed.



**Figure 2-2. Abyss Web Server icon in the system tray**

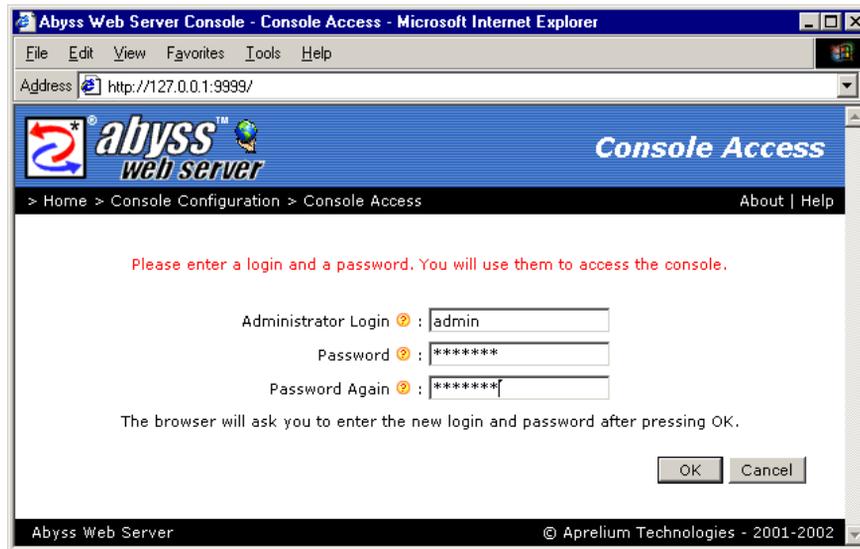
Abyss Web Server creates automatically a configuration file where it stores the web server's parameters. Then, it opens the browser and displays the console.



**Figure 2-3. Configuration file creation notification**

*Note: Refer to the "Troubleshooting Guide" appendix to solve the most common problems that can occur with Abyss Web Server startup.*

Choose a login and a password to fill with the form displayed in the browser and press **OK**.



**Figure 2-4. Console Access setup**

The browser asks you for your credentials. Enter the login and the password you have already chosen and validate.



**Figure 2-5. Credentials dialog**

At this point, Abyss Web Server is ready to serve. To test it, point the browser to the web server URL, which is printed on the server's window. You should see the **Welcome to Abyss Web Server** page.

## Setting up a web site

Using your favorite web page editor, create a web site and put its files in the directory **<Abyss Web Server directory>/htdocs**. The index files must be called **index.html** or **index.htm**. The CGI scripts must be put in **<Abyss Web Server directory>/cgi-bin**.

Other computers on the network can access your web site if they browse `http://<your host name>:<server port>`.

`<your host name>` is the host name of your computer or its IP address. Ask your network's administrator for this information. If you want to connect to the server from the computer it runs on, you can also use `localhost` or `127.0.0.1`.

`<server port>` is the number of the port the server waits for connections on. It is printed on the server's window or terminal. If `<server port>` is 80 (which is the default), it can be omitted from the URL and the web server can be accessed with only `http://<your host name>`.

## Shutting down the server

Select **Exit** from the **Server** menu or close the server's main window. If the server is running, you will be asked to confirm exiting.

You can also select **Server Status** in the console and press **Halt**. You have to manually close the server's main window after.

## Asking for support

If you have questions or need help, please select **Help and Support** in the console and use one of the support links available there.

# Chapter 3. Using The Console

## What is the console?

The console is a remote web based configuration and monitoring system. With the console, you can:

- Configure Abyss Web Server.
- Stop, run and shutdown the web server.
- View the web server's access statistics.
- Read the documentation.

## Accessing the console

From your computer

Browse `http://127.0.0.1:<console port>`.

From any computer connected to the network

Browse `http://<your host name>:<console port>`.

`<console port>` is printed on the web server's window . By default, it is 9999.

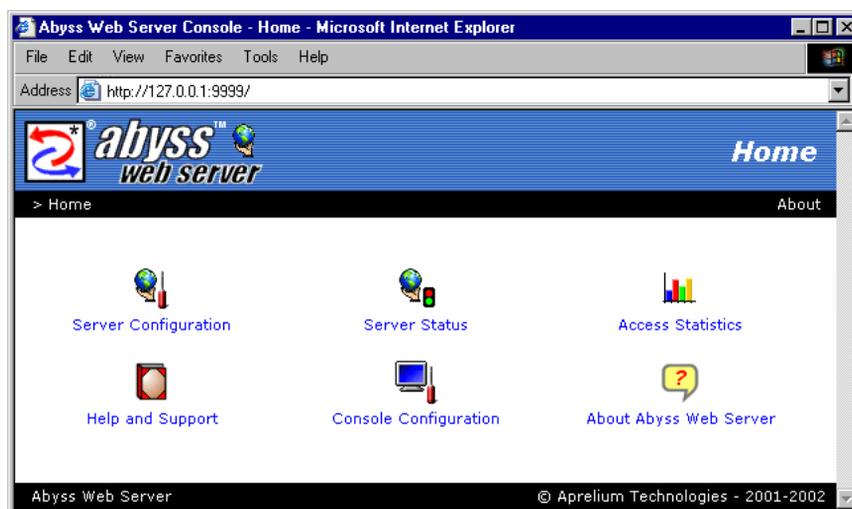


Figure 3-1. Console homepage

## Console interface basics

Although the console has an intuitive point and click interface, some of its elements are described more in depth in this section.

### Help buttons



**Figure 3-2.** A tip on a help button

A small tip is displayed when you point a help button . Pressing it displays the related documentation topic.

*Note:* Not all the browsers support displaying tips.

### Tables



**Figure 3-3.** A sample table

Each line of a table contains a pen button  and a trash button . Press the pen button on a line to display a dialog where you can modify its contents or press the trash button to delete it. Pressing **Add...** displays a dialog where you can enter the contents of a new line.

## Applying configuration changes

After any configuration change, the console displays a request to restart the server. Just press the **Restart** button and wait until the server restarts.



Figure 3-4. Example of a restart request

*Note: You can accumulate many configuration changes before restarting.*

## Console Configuration

To change the console parameters, open the console and select **Console Configuration**.

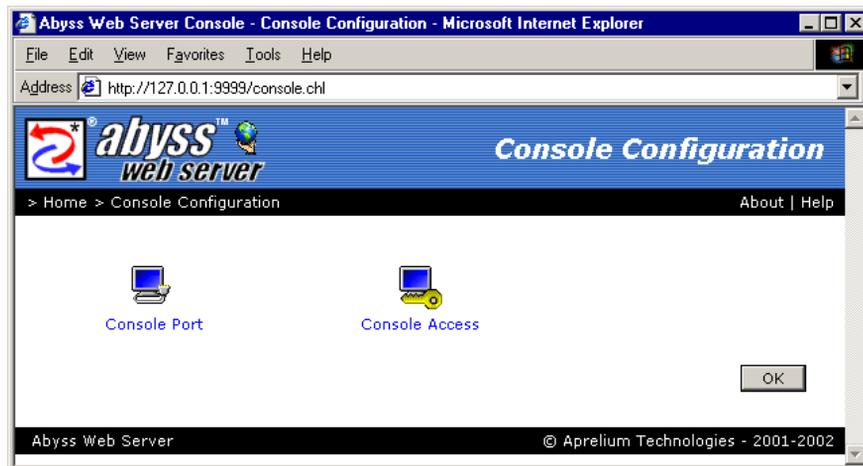
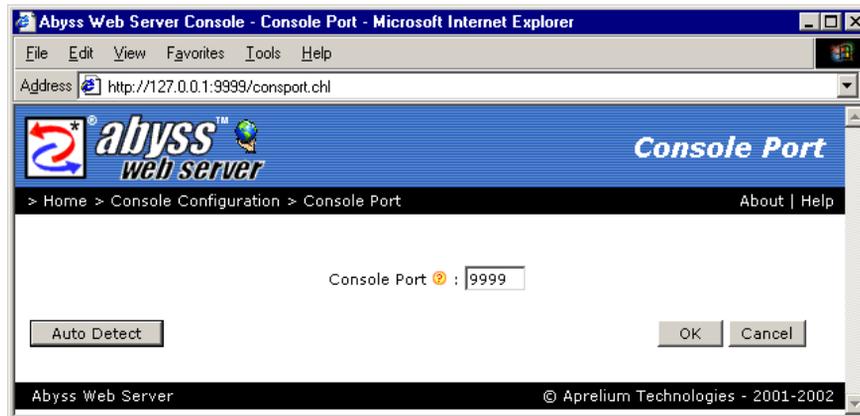


Figure 3-5. Console configuration dialog

## Changing the console port

In the **Console Configuration** dialog, select **Console Port**. Enter a new port number and press **OK** or press **Auto Detect**.



**Figure 3-6. Console port dialog**

If the port number is invalid or is already used by another application on your computer, the console displays an error and asks you to enter another value.

## Changing the console access credentials

Because the console access needs authorization, it is important to keep your console access credentials secret. It is also highly recommended not to choose them similar to your system login or password or any valuable credentials. Indeed, as the console relies on the browser to perform the authorization and as browsers use a relatively weak authorization scheme, console access information could be intercepted and deciphered by malicious persons on the network.

To change the console access credentials:

- Go to the **Console Configuration** dialog and select **Console Access**.
- Enter an administrator login.
- Enter a password. It must be at least four-character-long.
- Reenter the same password.
- Press **OK**.



**Figure 3-7. Console access dialog**

Once the change is made, the browser asks you to enter the new login and password.

## *Chapter 3. Using The Console*

# Chapter 4. Server Management

## Overview

All the server management tasks can be accomplished using the console. In this section, you will learn how to:

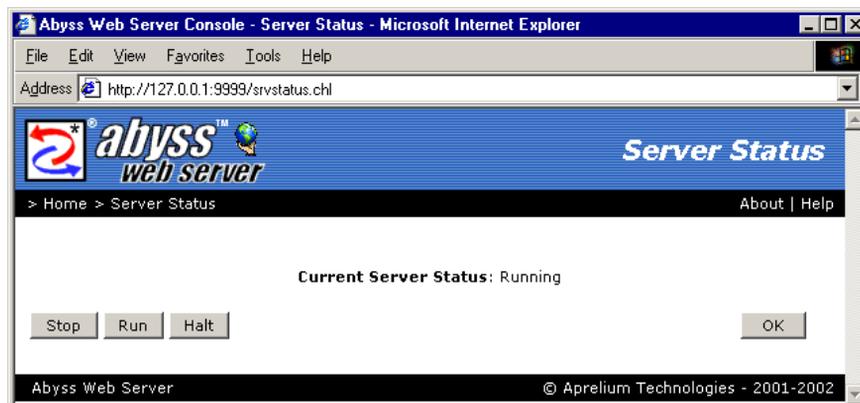
- Check and change the web server's status.
- Monitor the accesses to the web server.
- Modify its parameters.

## Server status

Open the console and select **Server Status**. The console displays the current status of the server. To change it, press one of the following buttons:

- **Stop**: Stop the server. The console is still available.
- **Run**: Start or restart the server.
- **Halt**: Halt the server. The web server and the console are shutdown.

Press **OK** to go back to the main console page without changing the server's status.



**Figure 4-1. Server status dialog**

*Note: The console freezes for a few seconds while the server status is changing.*

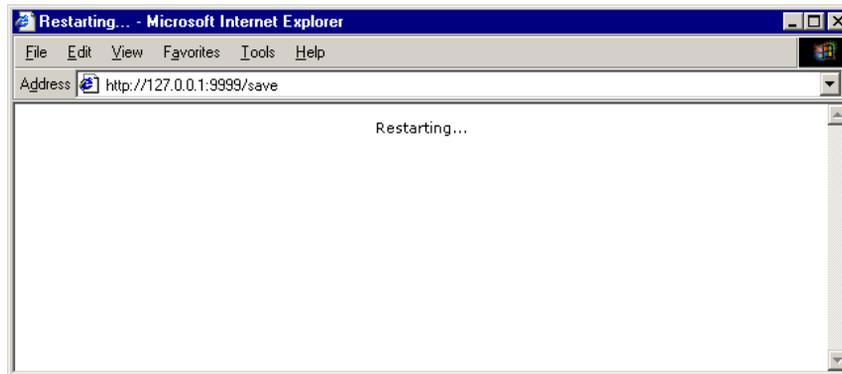


Figure 4-2. The server is restarting...

## Server access statistics

Open the console and select **Access Statistics**. The console displays a dialog containing a set of statistics on the server's activity:

- **Server Uptime:** The time elapsed since the server startup.
- **Hits:** The number of processed requests.
- **Error Hits:** The number of requests for which the server replied by an error.
- **HTML Hits:** The number of requests the server replied to by a document which MIME type was `text/html`.
- **Not Modified Hits:** The number of requests for which the server detected that the requested document has not changed.
- **Output (Kb):** The total size of the payload sent by the server to the clients in Kilobytes.



Figure 4-3. Access statistics dialog

The access statistics are refreshed automatically every 10 seconds. You can also press **Refresh Now** for immediate refreshing. They can be reset by pressing **Reset Statistics**.

Press **Cancel** to go back to the main console page.

## General server configuration

Open the console and select **Server Configuration** to display the general server configuration dialog.

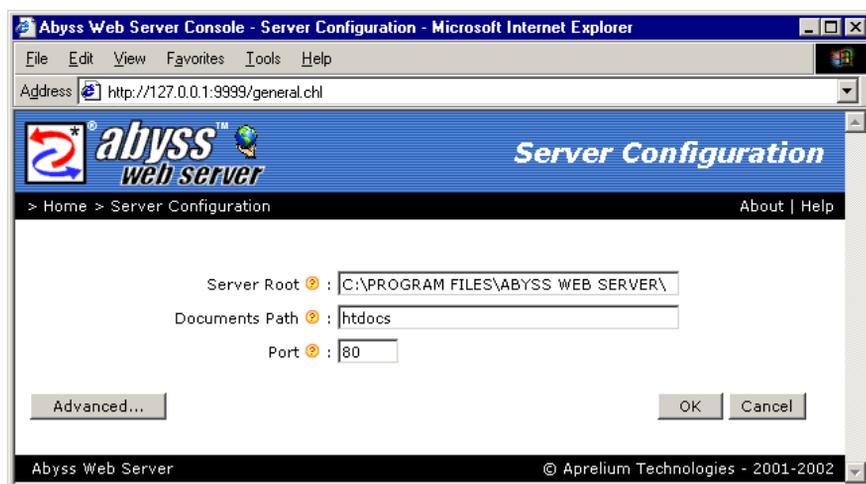


Figure 4-4. General server configuration dialog

The dialog includes the following fields:

- **Server Root:** The root path of the web server. By default, it is the directory where Abyss Web Server executable is installed. It is used as the base path for all relative real paths.
- **Documents Path:** The path of the web site files. If it is a relative path then it is considered as a subpath of the server root.
- **Port:** The number of the port the web server waits connections on. Its default value is 80.

Press **OK** to save the configuration changes. The console checks the entered values and displays a message if an error is detected.

## Advanced server configuration

To display the advanced server configuration menu, open the console, select **Server Configuration** and press the **Advanced...** button.

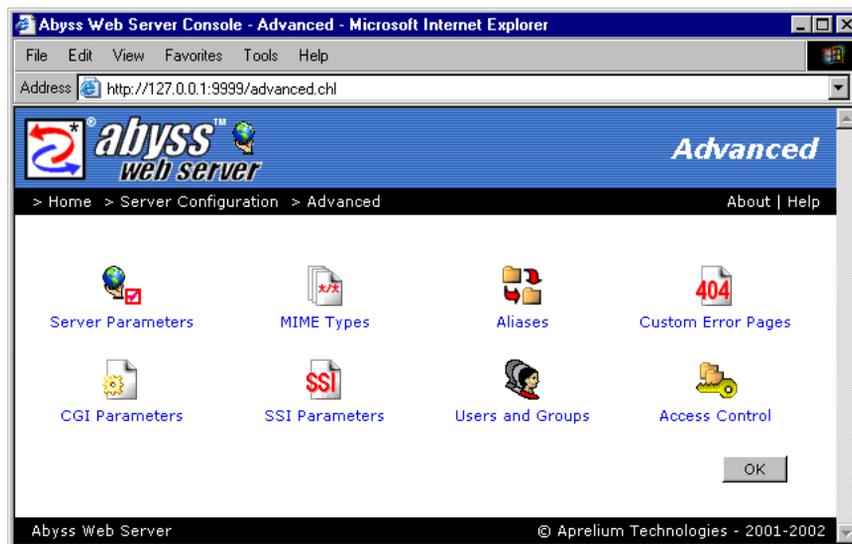
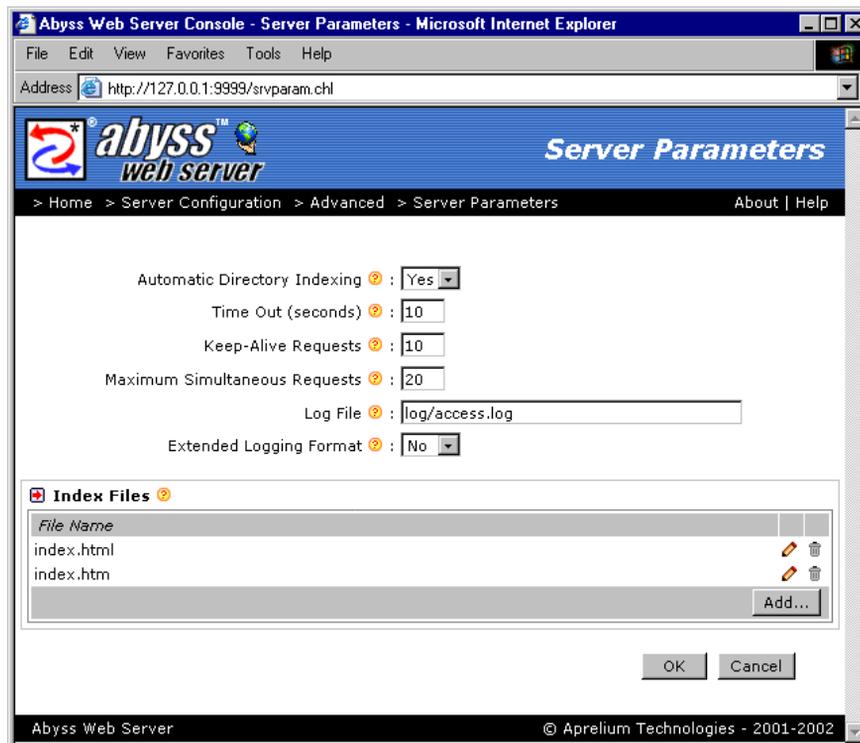


Figure 4-5. Advanced server configuration dialog

The following subsections describe each advanced configuration option.

### Server parameters

Select **Server Parameters** in the advanced server configuration menu to display the server parameters dialog.



**Figure 4-6. Server parameters dialog**

The dialog includes the following fields:

- **Automatic Directory Indexing:** Enable automatic generation of a directory listing if no index file is found.
- **Timeout:** How many seconds the server waits for an inactive connection before closing it.
- **Keep-Alive Requests:** The maximum number of requests that can be served over the same connection. Only HTTP/1.1 and some HTTP/1.0 compliant browsers can benefit from this feature.
- **Maximum Simultaneous Requests:** The maximum number of requests the server can serve in parallel.
- **Log File:** The path of the log file. If it is relative, it is considered as a subpath of the server root. If empty, logging is disabled.
- **Extended Log Format:** Add the referrer and the user agent to each log line.

## Index files

When a browser asks for a URL that does not contain a filename, the server checks for the existence of each index file in the mapped directory. If none is

found and automatic directory indexing is enabled, a directory listing is generated and sent to the browser. Otherwise, an error is reported.

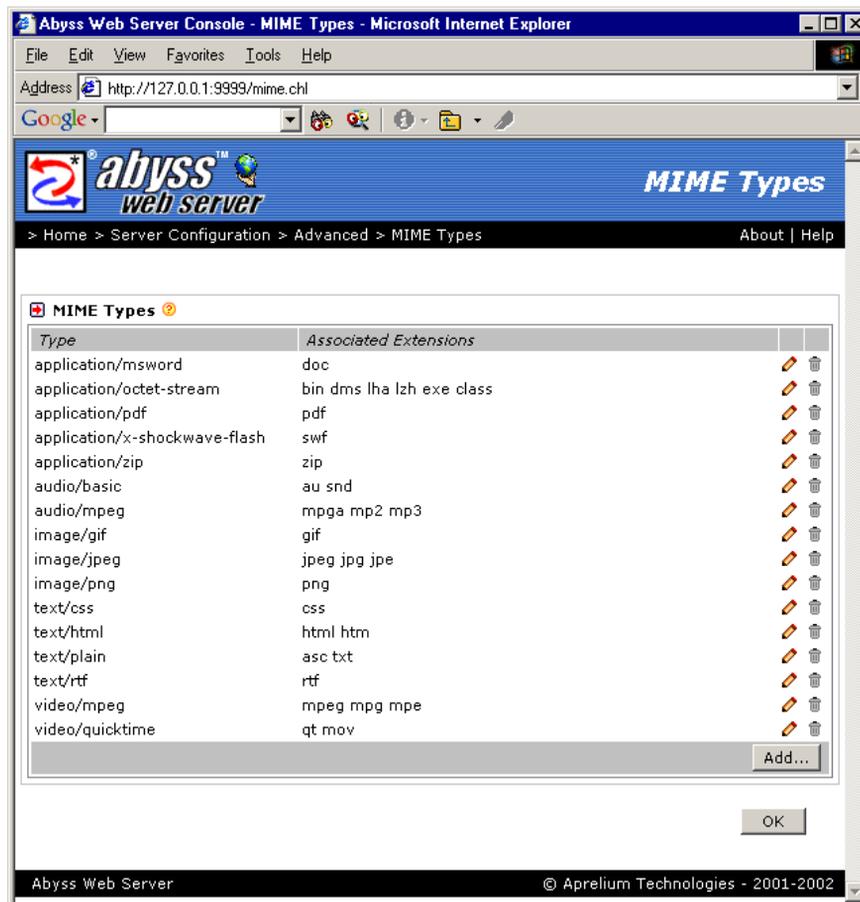
To edit, remove or add index filenames, use the **Index Files** table in the **Server Parameters** dialog.

***Example:** `index.htm` and `index.html` as index files*

*Assume that `index.html` and `index.htm` are set as index filenames. If a browser asks for `http://<your host name>:<server port>/hello/`, the server checks if `<documents path>/hello/index.html` exists. If not, it checks if `<documents path>/hello/index.htm` exists. If so, it is sent to the browser. If not, a listing of the directory `<documents path>/hello/` is generated and sent to the browser if automatic directory indexing is enabled. If it is disabled, the server replies with a forbidden error message.*

## MIME types

When the server sends a document to a browser, it also sends its MIME type. This information helps the browser to know what kind of file it is (HTML, ZIP, JPEG image, etc. . .) and what to do with (Display it, save it on the disk, launch a configured application to read it, etc. . .). Abyss Web Server comes with a preset list of common MIME types. To modify this list, select **MIME Types** in the advanced server configuration menu.



**Figure 4-7. MIME types dialog**

Use the displayed table to edit, remove or add MIME types. A MIME Type has the format **type/subtype** and is associated to one or more extensions separated by spaces.

*Example: The text/html MIME type*

HTML files have **text/html** as MIME type. By default, **text/html** is associated to extensions **html** and **htm**. As a consequence, when the server sends to a browser a document which extension is **html** or **htm**, it will also set the document's MIME type to **text/html**.

## Aliases

Select **Aliases** in the advanced server configuration menu to display the aliases table.

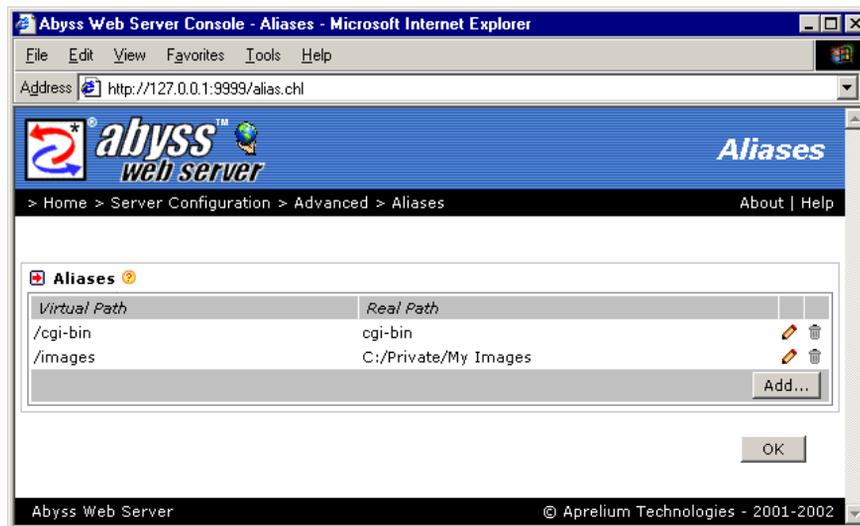


Figure 4-8. Aliases dialog

If a URL matches an alias' virtual path, the web server maps it to the alias' associated real path.

Use the displayed table to edit, remove or add aliases.

*Example: Relative real path*

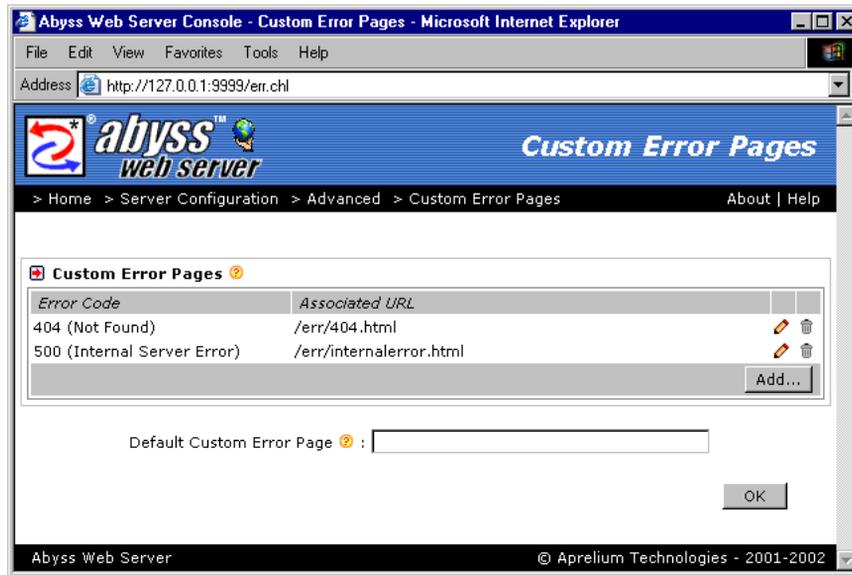
Assume that there exists an alias which virtual path is **/images** and which real path is **web/artwork**. If a browser asks for **http://<your host name>:<server port>/images/logo.jpg**, the server maps the requested URL to the file **<server root>/web/artwork/logo.jpg**. The **<server root>** is added because the real path was relative.

*Example: Absolute real path*

Assume now that there exists an alias which virtual path is **/images** and which real path is **d:\web\artwork**. If a browser asks for **http://<your host name>:<server port>/images/logo.jpg**, the server maps the requested URL to the file **d:\web\artwork\logo.jpg**. The difference with the previous example is that the real path is absolute and not relative.

## Custom error pages

With Abyss Web Server, you can override the standard error pages and replace them with yours. To do so, select **Custom Error Pages** in the advanced server configuration menu.



**Figure 4-9. Custom error pages dialog**

This dialog includes the following elements:

- **Custom Error Pages:** This table contains the customized errors and their associated URLs.
- **Default Custom Error Page:** The URL used when an error which code is not listed in the **Custom Error Pages** table occurs. If empty, Abyss Web Server generates automatically a standard error page.

An error URL can be:

- **Local:** If it begins with a slash /, the URL is local to the web server.
- **Global:** If it begins with **http://**, the URL is global and the web server informs the browser a redirect to that URL when an error occurs.

***Note:** It is only relevant to set 4xx and 5xx error codes. Other error codes are handled internally in the web server and do not lead to displaying an error page.*

***Note:** When using a local URL that is a CGI script or an SSI page as a custom error page, the server operates an internal redirection and adds to the custom error page's environment variables all the faulty request environment variables prefixed with **REDIRECT\_**. It adds also the special variable **REDIRECT\_STATUS** which contains the status code of the faulty request. For more information, read "[CGI environment variables](#)" section in "[CGI Scripts](#)" chapter.*

## CGI parameters

To configure CGI scripts execution, select **CGI Parameters** in the advanced server configuration menu.

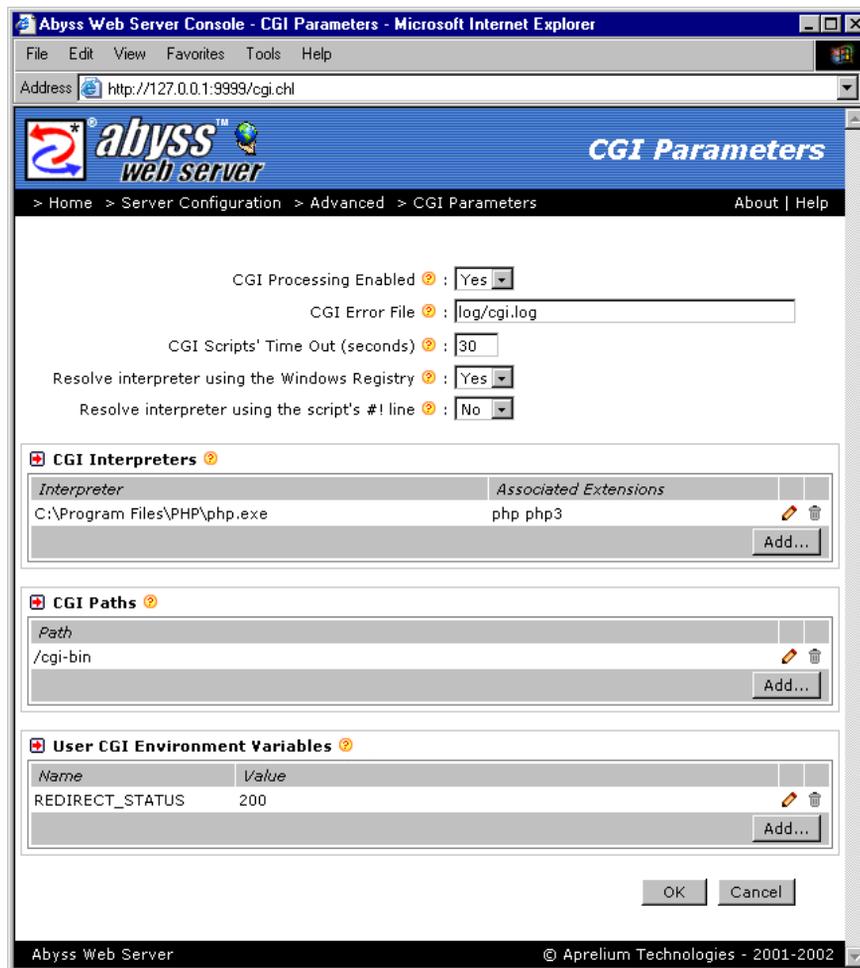


Figure 4-10. CGI parameters dialog

This dialog includes the following fields:

- **CGI Processing Enabled:** Enable/disable CGI scripts execution.
- **CGI Error File:** The path of the file where CGI scripts write error messages. You can leave it empty if you do not want to trace CGI scripts' errors.
- **CGI Scripts' Timeout:** How long (in seconds) the server should wait for a CGI script to deliver content before aborting it.
- **Resolve CGI Interpreter using Windows Registry:** Use the Windows Registry to find automatically the interpreter that can run a CGI script.

This parameter should be used very carefully as it can make the server report Error 500 (Internal Server Error) for normal documents. For example, if a HTML file is in one of the CGI Paths (or matches with one of the CGI Paths patterns), and if this parameter is set to **Yes**, the server asks the Windows Registry about the executable that is normally used to open HTML files (in a similar fashion to what does Windows Explorer to know what application to launch when you double-click on a document icon). The Windows Registry gives back your browser executable path and Abyss Web Server runs the HTML file as a CGI Script with this browser as its interpreter. But after launching it, Abyss Web Server understands that this executable is not a valid CGI Interpreter. So it aborts the executable and reports Error 500.

- **Resolve CGI Interpreter using the #! Line in Script:** Read the first line of the CGI script. If it begins with **#!**, the rest of the line is considered as the path to the script's interpreter.

The dialog contains also the following tables:

- **CGI Interpreters:** The server uses this table to know which interpreter to use to execute a CGI script. The choice is based on the script's extension: an interpreter runs a script if the extension of that script matches with one of the associated extensions or extensions patterns of the interpreter. Each line of the table contains the path of an interpreter and one or more associated extensions or extensions patterns separated by spaces. Refer to the "[Patterns Format](#)" appendix for more information about patterns.

When adding or editing a CGI Interpreter, letting the **Automatic Update of the CGI Paths** set to **Yes** makes Abyss Web Server add/remove the pattern **/\*.ext** in the CGI Paths table for every added/removed extension **ext** in associated extensions list.

- **CGI Paths:** Only files that are in one of the CGI paths or theirs subpaths, or that match with one of the CGI Paths patterns can be processed as CGI scripts. These paths are virtual paths. See the "[Patterns Format](#)" appendix for more information about patterns.
- **User CGI Environment Variables:** This table contains the variables you want to add to the execution environment of the CGI scripts. Each variable is defined by its name and its value.

For more information about CGI scripts, see "CGI Scripts" chapter.

## SSI parameters

To configure SSI (Server Side Includes), select **SSI Parameters** in the advanced server configuration menu.

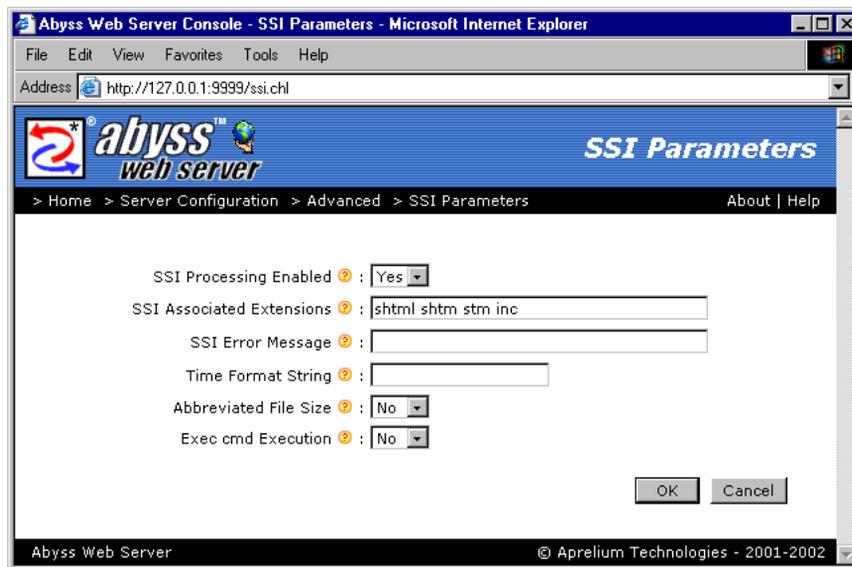


Figure 4-11. SSI parameters dialog

This dialog includes the following fields:

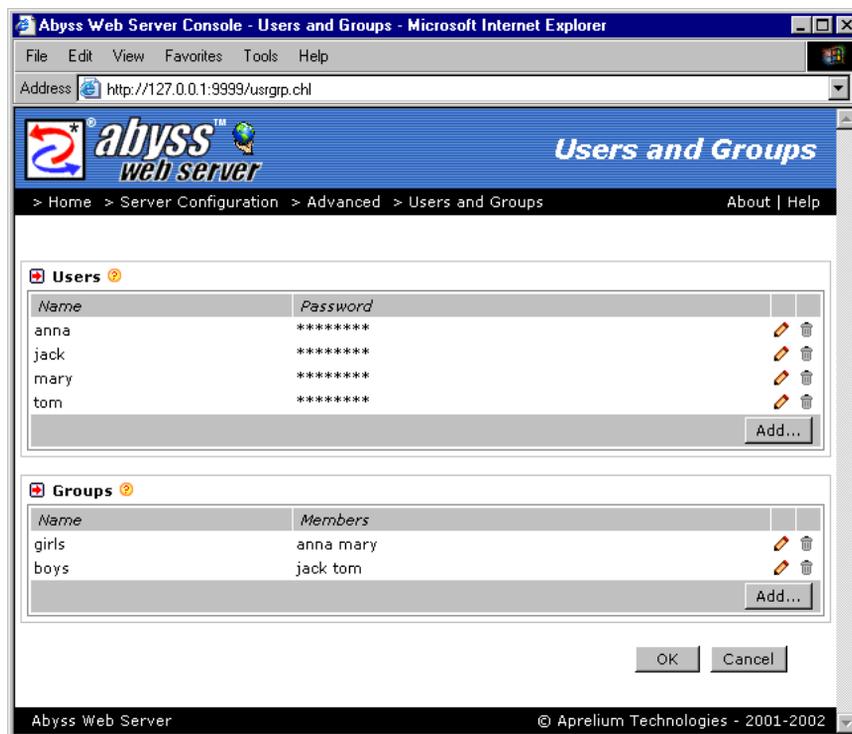
- **SSI Processing Enabled:** Enable/disable SSI processing.
- **SSI Associated Extensions:** If a file name extension matches with one of these extensions or extensions patterns, it is processed as an SSI file. Read the "[Patterns Format](#)" appendix for more information about patterns.
- **SSI Error Message:** The default error message that the server inserts when an error is detected while processing an SSI directive. If empty, an accurate error description with debugging information is generated.
- **SSI Time Format:** The default time format string that the server uses to display times while processing SSI directives. If empty, the string `%A, %d-%b-%Y %H:%M:%S %Z` is used. For the complete reference of the time format string, read the description of `<!-- #config timefmt="time_format" ->` directive in "[Server Side Includes](#)" chapter.
- **Abbreviated File Size:** The default way to display file sizes. If set to **Yes**, file sizes are displayed in Kb or Mb. Otherwise, they are displayed in bytes.

- **Exec cmd Execution:** Enable/disable the execution of shell commands in SSI.

For more information about SSI, refer to "[Server Side Includes](#)" chapter.

## Users and Groups

Select **Users and Groups** in the advanced server configuration menu to display the users and groups tables.



**Figure 4-12. Users and groups dialog**

Use the displayed tables to edit, remove or add users and groups.

A user is defined by its name and its password. A group is defined by its name and its members which can be users and other groups.

*Note:* The console hides automatically groups that can lead to circular references when editing a group.

## Access Control

To restrict and manage the access to paths in your web site, select **Access Control** in the advanced server configuration.

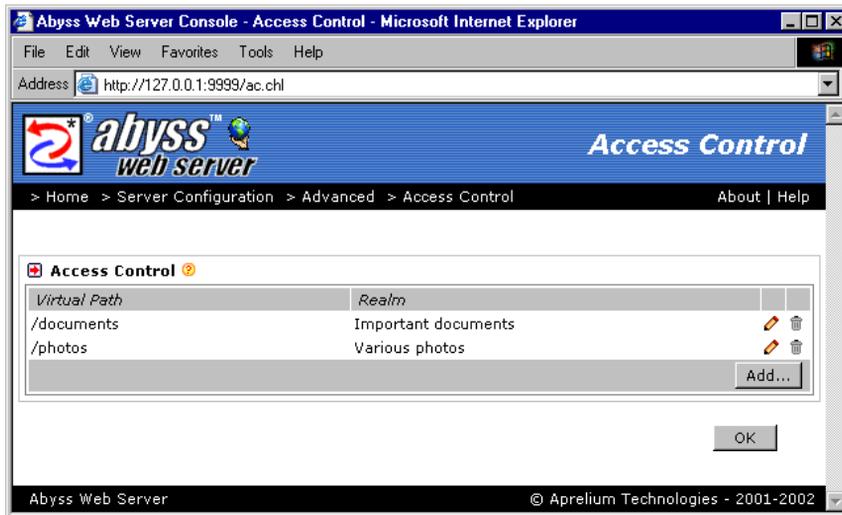


Figure 4-13. Access control dialog

To edit, remove or add path access rules, use the displayed table.

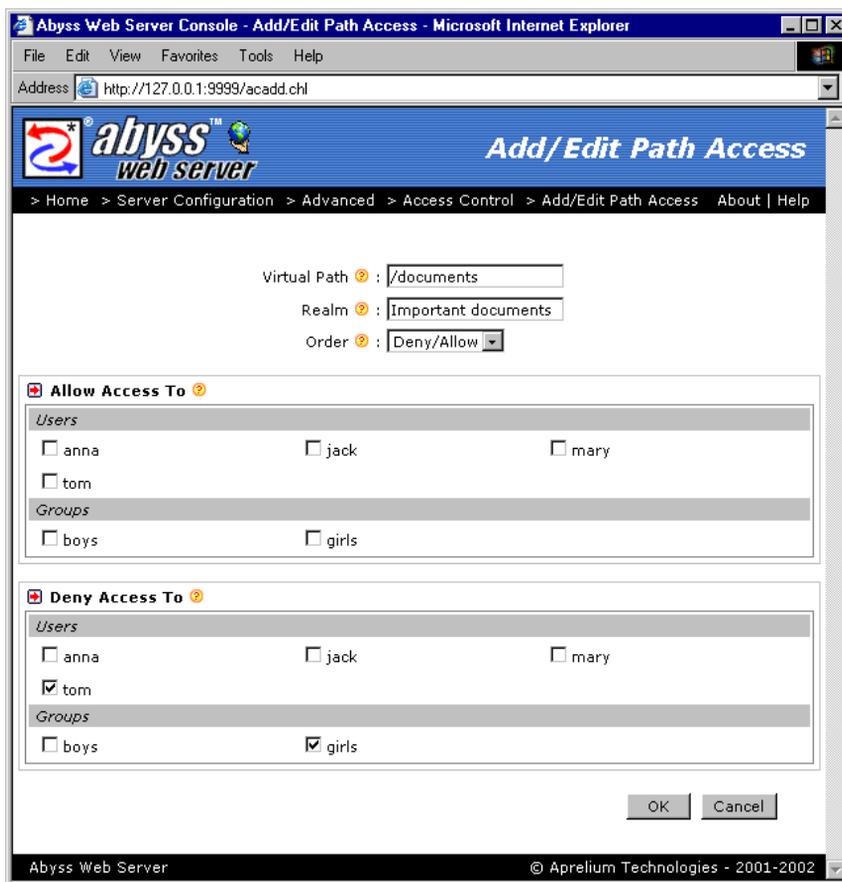


Figure 4-14. Path access dialog

The path access edition dialog contains the following fields:

- **Virtual Path:** The virtual path which contents' access is to be restricted. It

must always begin with a slash /. This field accepts also path patterns. In that case, the access rule applies to any virtual path that matches with the specified pattern. Refer to the "[Patterns Format](#)" appendix for more information about patterns.

- **Realm:** A short description of the path's contents. It is prompted to users by the browser when it asks them for credentials.
- **Order:** The order that the server follows to check if access is granted to a user. If it is set to **Allow/Deny**, access is denied by default and is allowed only if the user is in the **Allow Access To** list and is not in the **Deny Access To** list. If it is set to **Deny/Allow**, access is allowed by default and is denied only if the user is in the **Deny Access To** list and is not in the **Allow Access To** list.
- **Allow Access To:** The list of users and groups for whom access is allowed.
- **Deny Access To:** The list of users and groups for whom access is denied.

*Note: If **Deny Access To** list is empty and the order is **Deny/Allow**, access is granted to all declared users and groups.*

## *Chapter 4. Server Management*

# Chapter 5. CGI Scripts

## Setting up an interpreter

To setup up a language interpreter to be used by CGI scripts in Abyss Web Server, follow these instructions:

- Download the language package.
- Install it. Refer to the documentation accompanying it for details about this operation.
- In the **CGI Parameters** dialog, use the **CGI Interpreters** table to add the new interpreter and the filename extensions it handles.
- CGI scripts requiring this interpreter to run must have their filename extensions set to one of the declared extensions.

Visit <http://www.aprelium.com/abyssws> to get updated information about downloading and setting up PHP, Perl and other scripting languages.

## How are they run?

When the server is asked for a document, it first checks if it is a CGI script. CGI scripts must belong to one of the CGI paths or their subpaths, or must match with one of the path patterns listed in the CGI Paths.

Abyss Web Server is able to run standalone executables as CGI scripts. These executables can be GUI or non-GUI Windows applications or MS-DOS EXE or COM programs. The CGI Paths table must contain explicitly the general patterns `/*.exe` and/or `/*.com` (or more path restrictive patterns ending with `*.exe` and/or `*.com`) in order to have them executed. Otherwise, the executable files are downloaded to the client browser.

For other kinds of scripts, the server tries to find the suitable interpreter able to run them by:

- Checking the **CGI Interpreters** table and trying to match the script's extension with a declared interpreter.
- Using the Windows Registry to match the script's extension with an available interpreter.

- Reading the first line of the script. If it begins with **#!**, the rest of the line must contain a valid interpreter path.

If none of these steps is successful, the file is considered as a normal document and will be sent to the browser.

Abyss Web Server automatically detects NPH (Non Parsed Headers) scripts. NPH scripts output is sent directly to the browser without prior header decoding by the server. It is up to NPH scripts to generate correctly all the necessary HTTP response headers. Abyss Web Server considers a script to be NPH if the the first line it outputs starts with the string **HTTP/**.

## CGI environment variables

Before running a CGI script, Abyss Web Server sets its environment variables in conformity with the CGI/1.1 specification (as described in <http://hoohoo.ncsa.uiuc.edu/cgi>) and adds variables declared in the **User CGI environment variables** table as well as some system environment variables (namely **PATH**, **TEMP**, **TMP**, **SystemRoot**, **COMSPEC**, **PATHEXT**, and **WINDIR** .)

The following list contains the variables documented in the CGI/1.1 specification and some variables commonly set by web servers:

- **PATH\_INFO**: The extra path information, as given in the requested URL. In fact, scripts can be accessed by their virtual path, followed by extra information at the end of this path. The extra information is sent as **PATH\_INFO**.
- **PATH\_TRANSLATED**: The virtual-to-real mapped version of **PATH\_INFO**.
- **SCRIPT\_NAME**: The virtual path of the script being executed.
- **SCRIPT\_FILENAME**: The real path of the script being executed.
- **SERVER\_SOFTWARE**: The web server's software identity.
- **SERVER\_NAME**: The host name or the IP address of the computer running the web server as given in the requested URL.
- **SERVER\_PORT**: The port to which the request was sent.
- **GATEWAY\_INTERFACE**: The CGI Specification version supported by the web server; always set to **CGI/1.1**.
- **SERVER\_PROTOCOL**: The HTTP protocol version used by the current request.

- **REQUEST\_METHOD**: The method used by the current request; usually set to **GET** or **POST**.
- **QUERY\_STRING**: The information which follows the **?** character in the requested URL.
- **REMOTE\_ADDR**: The IP address of the computer that sent the request.
- **REMOTE\_PORT**: The port from which the request was sent.
- **CONTENT\_TYPE**: The MIME type of the request body; set only for **POST** or **PUT** requests.
- **CONTENT\_LENGTH**: The length in bytes of the request body; set only for **POST** or **PUT** requests.
- **DOCUMENT\_ROOT**: The absolute path of the web site files. It has the same value as **Documents Path**.
- **AUTH\_TYPE**: The authentication type if the client has authenticated itself to access the script.
- **AUTH\_USER** and **REMOTE\_USER**: The name of the user as issued by the client when authenticating itself to access the script.

In addition to these variables, all header lines received in the request are added to the environment with the prefix **HTTP\_** followed by the header name in upper cases. All - characters in the header name are changed to underscore **\_** characters. For example, **User-Agent** is translated to **HTTP\_USER\_AGENT**.

If the request results from an internal redirection (from an SSI document or if it is used as a custom error page for example), the environment variables of the parent request are also added and each variable name is prefixed by **REDIRECT\_**. The parent request's status code is stored in the special variable **REDIRECT\_STATUS**. The cookies of the parent request are also passed to the redirected request in the **COOKIES** environment variable.

*Example: Using the redirection information in CGI scripts*

*If a CGI script is invoked after an internal redirection, it can retrieve some interesting information from the environment about its parent request such as:*

- **REDIRECT\_SCRIPT\_NAME**: *The virtual path of the parent document.*
- **REDIRECT\_STATUS**: *The status code of the parent request. It is useful when writing scripts that generate custom error pages.*

## *Chapter 5. CGI Scripts*

# Chapter 6. Server Side Includes

## What are SSI?

SSI (Server Side Includes) are a set of directives that can be put in a HTML page and interpreted by the web server while it sends the page to a client. You can use SSI to insert in your HTML page:

- The value of an environment variable.
- The size and the last modification time of a file.
- A file.
- The output of a CGI script or a shell command.

## SSI Syntax

```
<!-- #directive attr1="value1" attr2="value2" ... -->
```

Directives are enclosed in standard [HTML](#) comments. So if SSI processing is not enabled, your browser will ignore them. Otherwise, each directive is evaluated and is replaced by its results.

*Note: Enclosing an attribute's value in double quotes is not mandatory if it does not contain whitespaces.*

*Note: A whitespace should precede the comment terminator -->.*

## Directives

### #config

This directive is used to override the default SSI settings while processing the SSI file.

**Form 1:** `<!-- #config errmsg="error message" -->`

`#config errmsg` is used to change the message sent to the client when an SSI error occurs. By default, SSI errors are reported in a detailed manner to allow easy debugging.

**Form 2: <!-- #config sizefmt="size\_format" -->**

**#config sizefmt** controls the way the server displays SSI file sizes. **size\_format** can have two values:

- **ABBREV**: displays file sizes in kilobytes (Kb) or megabytes (Mb).
- **BYTE**: displays file sizes in bytes. It is the default.

**Form 3: <!-- #config timefmt="time\_format" -->**

**#config timefmt** specifies the format in which SSI dates are inserted. **time\_format** must contain formatting tokens that are substituted by the corresponding parts of the date. The available tokens are listed below with the description of their substitution:

- **%a**: Abbreviated name for day of the week.
- **%A**: Complete name for day of the week.
- **%b**: Abbreviated month name.
- **%B**: Complete month name.
- **%c**: Date and time representation.
- **%d**: Day of the month (01-31).
- **%H**: Hour in 24-hour format (00-23).
- **%I**: Hour in 12-hour format (01-12).
- **%j**: Day of the year as a decimal number (001-366).
- **%m**: Month as a decimal number (01-12).
- **%M**: Minute as a decimal number (00-59).
- **%p**: A.M. or P.M. indicator for 12-hour format.
- **%S**: Second as a decimal number (00-59).
- **%U**: Week of the year as a number (00-51). Sunday is the first day of the week .
- **%w**: Day of the week as a number (0-6). Sunday is the first day of the week.
- **%W**: Week of the year as a number (00-51). Monday is the first day of the week.
- **%x**: Date representation.
- **%X**: Time representation<– for the current C locale–>.

- **%y**: Year without the century (00-99).
- **%Y**: Year with the century (0000-9999).
- **%z**: Time zone name; empty if unknown.
- **%Z**: Time zone name abbreviation; empty if unknown.
- **%%**: Percent sign.

## #echo

**Form:** `<!-- #echo var="variable_name" -->`

**#echo** inserts the value of an SSI environment variable. SSI environment variables include those available for CGI scripts (see "[CGI environment variables](#)" in the "[CGI Scripts](#)" chapter) and five special variables:

- **DOCUMENT\_NAME**: The name of the processed SSI file.
- **DOCUMENT\_URI**: The virtual path to the current processed file.
- **DATE\_LOCAL**: The current local date.
- **DATE\_GMT**: The current universal (GMT) date.
- **LAST\_MODIFIED**: The last modification date of the processed file.

If **variable\_name** is not a valid SSI environment variable, the directive is ignored.

## #exec

**Form 1:** `<!-- #exec cgi="cgi_file" -->`

**#exec cgi** executes a CGI script and inserts its output. If **cgi\_file** begins with a /, it is considered as a virtual path and the directive will behave as if it is `<!-- #include virtual="cgi_file" -->`. If **cgi\_file** does not begin with a /, it is considered as a relative file name and **#exec cgi** will behave as if it is `<!-- #include file="cgi_file" -->`.

*Note: **#exec cgi** is supported only for compatibility purposes but its use is deprecated. You should use the **#include** directive instead.*

**Form 2:** `<!-- #exec cmd="shell_command" -->`

`#exec cmd` executes a shell command and inserts its output. The shell executable path is found in the system environment variable `COMPSEC`. If this variable is empty or does not exist, `command.com` is used as the default shell.

*Note: #exec cmd execution must be used with extreme care. If users can edit or modify HTML pages in your web sites (which is the case if you have a forum or a guestbook for instance), you must disable it. In fact, they can include SSI tags with dangerous #exec cmd directives and remotely invoke them.*

## #flastmod

**Form 1:** `<!-- #flastmod file="file_name" -->`

`#flastmod file` inserts the last modification date of the file `file_name`. `file_name` is a relative path from the directory containing the processed SSI file. It file can only be in the same directory or in one of its subdirectories.

**Form 2:** `<!-- #flastmod virtual="virt_path" -->`

`#flastmod virtual` inserts the last modification date of the file which virtual path is `virt_path`.

## #fsize

**Form 1:** `<!-- #fsize file="file_name" -->`

`#fsize file` inserts the size of the file `file_name`. `file_name` is a relative path from the directory containing the processed SSI file. It can only be in the same directory or in one of its subdirectories.

**Form 2:** `<!-- #fsize virtual="virt_path" -->`

`#fsize virtual` inserts the size of the content resulting from requesting `virt_path` from the server. If `virt_path` is a static file, its size will be inserted. If it is a CGI script or a SSI file, it is evaluated by the server and the size of the resulting output is inserted.

## #include

**Form 1:** `<!-- #include file="file_name" -->`

**#include file** instructs the web server to insert a file which must contain valid HTML contents. **file\_name** is a relative path from the directory containing the processed SSI file. The included file can only be in the same directory or in one of its subdirectories.

**Form 2:** `<!-- #include virtual="virt_path" -->`

**#include virtual** inserts the content resulting from requesting **virt\_path** from the server. If **virt\_path** is a static file, its contents will be included "as is". If it is a CGI script or a SSI file, it is evaluated by the server and the results are inserted. **virt\_path** must contain or generate valid HTML content.

## *Chapter 6. Server Side Includes*

# Glossary

## C

### **Common Gateway Interface**

*CGI* defines how web servers interact with an external program in order to generate dynamic content. Such programs can be written in any programming language including C, Perl and PHP.

## H

### **Host Name**

Each computer on a *TCP/IP* network has usually one or more *host names*. Networking software transparently translate them to *IP Addresses* for internal use.

### **HyperText Markup Language**

*HTML* is a markup language used to describe web pages.

### **HyperText Transfer Protocol**

*HTTP* defines how browsers and web servers communicate over a *TCP/IP* network.

## I

### **Internet Protocol**

*IP* is a low level networking protocol. It defines how computers can exchange messages over a network. *IP* is usually used in conjunction with *TCP*, a higher-level protocol.

### **IP Address**

An *IP Address* is a unique identifier for a computer connected to a [TCP/IP](#) network. It is made of four numbers separated by dots. Each number is between 0 and 255.

## **S**

### **Server Side Includes**

*SSI* are a set of directives that can be included in [HTML](#) pages and processed by the web server to create simple dynamic documents.

## **T**

### **Transport Control Protocol**

*TCP* is a high level networking protocol. It allows establishment of virtual connections between computers to send and receive data streams. *TCP* relies on [IP](#).

### **TCP/IP**

*TCP/IP* is a protocol suite developed by the U.S. Department of Defense for communications between computers. It has become the de facto standard for data transmission over networks, including the Internet. [TCP](#) and [IP](#) are the main protocols of *TCP/IP*.

## **U**

### **Universal Resource Locator**

An *URL* is the address of a document or a resource available on the Internet.



## *Glossary*

# Appendix A. Configuration File Format

Abyss Web Server configuration file is usually located in the same directory as the Abyss Web Server executable.

The configuration file is a text file. Lines which begin with the # character are ignored and considered as comments. Each line contains only a unique option and one or more parameters. Option names are not case sensitive. For example, **ServerRoot** is the same as **SERVERROOT** and **seRverROoT**. Boolean parameters can be either **Yes** or **No**.

Available options are described below:

## **Version** <Abyss Web Server version>

The version of the software that generated the configuration file.  
*Appeared in version 1.0.3.*

## **ServerRoot** <real path>

The Root path for the web server. It is the base path for all relative real paths. It must occur before any other option that have a relative path as a parameter.

## **Path** <real path>

The path of the directory that contains the hosted web site.

## **Port** <integer>

The number of the port on which the web server listens to incoming connections. It must be between 1 and 65534.

## **KeepAlive** <integer>

The number of requests allowed to be served over the same connection when communicating with HTTP/1.1 and some HTTP/1.0 clients.

## **TimeOut** <integer>

How long (in seconds) the server waits for an inactive connection before closing it.

## **AdvertiseServer** <boolean>

Include the server software identity in each response.

## **MaxConnections** <integer>

The maximum number of requests served simultaneously.

## **LogFile** <real file>

The file where the server logs accesses.

## Appendix A. Configuration File Format

### **ExtendedLog** <boolean>

Add the referrer and the user agent to each log line.

### **AutoIndex** <boolean>

Enable automatic generation of a directory listing if no index file is found.

### **IndexFile** <string>

Add an index filename. This option can occur more than once.

### **MIMEFile** <real file>

Name of an external MIME types file. Each of its lines must contain a MIME type and one or more of its associated extensions separated by spaces.

### **MIMETYPE** <string>/<string> <extension> [ <extension> ... ]

Associate the extension(s) with the MIME type. This option can occur more than once.

### **Alias** <virtual path> <real path>

Map the virtual path to the real one. This option can occur more than once.

### **CGIEnabled** <boolean>

Enable CGI scripts execution.

### **CGITimeout** <integer>

How long (in seconds) the server waits for a CGI script to deliver content before aborting it. *Appeared in version 1.0.3.*

### **CGIUseRegistry** <boolean>

Use the Windows Registry to find automatically the interpreter that can run the CGI script.

### **CGIUseScript** <boolean>

For each CGI script, read its first line. If it begins with **#!**, the rest of the line should contain the path to a valid interpreter.

### **CGIErrorFile** <real file>

The file where the CGI scripts write error messages.

### **CGIInterpreter** <real file> <extension> [ <extension> ... ]

Declare an interpreter that runs CGI scripts which extensions are in the list of extensions given as parameters. This option can occur more than once.

**CGIPath <virtual path>**

Allow CGI scripts execution if they belong to the given virtual path or its subpaths. This option can occur more than once.

**CGIEnv <string>=<string>**

Add a user variable to the CGI scripts' environment. This option can occur more than once.

**ConsolePort <integer>**

The number of the port on which the console listens to incoming connections.

**Login <string>**

The console access login.

**Password <string>**

The encrypted console access password. *Encryption of the password appeared in version 1.0.3.*

**SSIEnabled <boolean>**

Enable SSI processing. *Appeared in version 1.0.7.*

**SSIExtension <extension>**

Enable SSI processing for files with the given extension. *Appeared in version 1.0.7.*

**SSISizeAbbrev <boolean>**

Define the default way to display file sizes in SSI. If set to true, it is in Kb or Mb. Otherwise, it is in bytes. *Appeared in version 1.0.7.*

**SSITimeFmt <string>**

Define the default displaying format of dates in SSI. *Appeared in version 1.0.7.*

**SSIErrMsg <string>**

Define the default error message displayed when an SSI error occurs. *Appeared in version 1.0.7.*

**CustomError <status-code> <url>**

Define a custom error page. *Appeared in version 1.0.7.*

**CustomErrorDefault <url>**

Define the default custom error page. *Appeared in version 1.0.7.*

**User <user name> <encrypted password>**

Declares a user. *Appeared in version X1.*

## Appendix A. Configuration File Format

**Group** <group name> [ <member> ...]

Declares a group. Members are users or groups already declared.  
*Appeared in version X1.*

**AccessControl** <virtual path> <realm string> +|- [ <user/group> ...] +|- [ <user/group> ...]

Adds a path access rule. Users and groups preceded by + are those for whom access is allowed. The allow/deny order is deduced from the + and - order. *Appeared in version X1.*

# Appendix B. Patterns Format

When matching a string (a sequence of characters containing an extension or a virtual path) with a pattern, the following rules apply:

- `*` matches any sequence of characters with any length (zero or more),
- `?` matches any character,
- `[set]` matches any character in the specified set,
- `[!set]` or `[^set]` matches any character not in the specified set.
- `\` suppresses the syntactic significance of a special character,

A set is made of characters or ranges. A range is formed by two characters with a `-` in the middle (as in `0-9` or `a-z`).

Preceding a special character with `\` makes it loose its syntactic significance and match that character exactly. Special characters are `[ ] * ? ! ^ - \`.

*Example: Examples of Patterns*

- `*`: any string including the empty string matches with this pattern.
- `*.*`: `list.txt` and `holiday.jpeg` match while `my-directory` doesn't.
- `/*.php`: `/hello.php` and `/mysite/scripts/test.php` match but `/mysite/test2.pl` doesn't.
- `/**/*.php`: `/mysite/test.php` and `/mysite/scripts/test.php` match with this pattern while `/test.php` doesn't.
- `mp?`: Any 3 character long string that begins with `mp` matches.
- `?????`: Any 5 character long string matches.
- `mp[2-4]`: Only `mp2`, `mp3`, and `mp4` match.
- `mp[!2-4]`: All 3 character long strings that begin with `mp` match except `mp2`, `mp3`, and `mp4`.
- `mp[2-4g]`: Only `mp2`, `mp3`, `mp4`, and `mpg` match.
- `mp[2\-4]file`: Only `mp2file`, `mp-file`, and `mp4file` match.

*Appendix B. Patterns Format*

# Appendix C. Uninstalling The Software

To run the uninstaller, open the **Start** menu, choose **Programs**, then **Abyss Web Server** and select **Uninstall**. An alternative way is to open the **Add/Remove Programs** section in the Window's **Control Panel**, select **Abyss Web Server** from the dialog's listbox and press **Install/Uninstall...**

Follow the on-screen information to remove Abyss Web Server. The uninstaller may ask you if you wish to completely delete Abyss Web Server's directory. Be careful ! If you confirm, it erases the server's configuration file and eventually the hosted web site.

## *Appendix C. Uninstalling The Software*

# Appendix D. Solving Ports Listening Problems

TCP/IP ports range from 1 to 65535. Abyss Web Server reports a listening problem on a port when one of the following situations is happening:

- Another instance of Abyss Web Server is already running on your computer.
- The port is already used by another application: Try stopping the other applications running on your computer to know which one is already using this port. More specifically, port 80 is usually used by other web servers. So if Abyss Web Server cannot listen to this port, there are chances that another web server is already running. Some Windows versions install by default Microsoft IIS (Internet Information Server) which uses port 80. If you do not use IIS, disable it or uninstall it to free this port.

An alternative solution would be to configure Abyss Web Server to use another web server or console port. For more information about changing the web server port, refer to "[General server configuration](#)".

If you have to change the console port, select **Settings...** in the main window, enter a new value in the **Port** field or press **Autodetect**, press **OK** and confirm starting the server.

*Appendix D. Solving Ports Listening Problems*

# Appendix E. Troubleshooting Guide

1. The console or the web site cannot be accessed from the computer on which Abyss Web Server is running.

If you cannot browse **127.0.0.1** or **localhost**, check if you have configured a HTTP proxy in the browser. If so, disable proxy usage for **127.0.0.1** and **localhost**.

2. Abyss Web Server reports the following error at startup: "Cannot start the web server (listening problem)"

This error arises when the web server port is already used by another application or if you do not have the privilege to use it. You have to choose another port for the web server using the console. Refer to "[Solving Ports Listening Problems](#)" for more information.

3. Abyss Web Server reports the following error at startup: "Cannot start the console (listening problem)"

This error occurs when the console port is already used by another application or if you do not have the privilege to use it. Refer also to "[Solving Ports Listening Problems](#)" for more information.

4. PHP CGI scripts display a security alert instead of being executed.

New versions of PHP activate by default a high security level to avoid problems that can occur when running them with Apache. Since these problems are specific to Apache and do not affect Abyss Web Server, you can safely deactivate the security alert as follows: open the console, select **Server Configuration**, press **Advanced...** and select **CGI Parameters**. Then press **Add...** in the **User CGI Environment Variables** table and add a variable which name is **REDIRECT\_STATUS** and which value is **200**.

*Appendix E. Troubleshooting Guide*