ERICSSON

Active Library Explorer Server, Installation Guide for UNIX® Platforms

INSTALLATION INSTRUCTION

1/1531-CNAH 118 010 Uen AA

Contents

1 Introduction

This document describes the installation procedure for Active Library Explorer Server, a web-based tool for viewing and searching electronic documentation.

Intended audience is local system administration personnel responsible for maintenance of file servers where Active Library Explorer (ALEX) document libraries are stored, and/or web servers where the web application itself is installed.

The installation procedure consists of the following steps:

1 Install the Active Library Explorer Server (see Section 2 for new installation or Section 3 for upgrading from old installation)

2 Verify that the installation works (see Section 2.5)

Once the system is configured and started as described in this document, very little further action is required to administer it. Thus, no separate user guide is provided.

Users of the Active Library Explorer also need the Adobe® Reader® program, which is used for viewing documents in PDF format. The program can be used as a plug-in in the Firefox® browser, or as an ActiveX® control in the Microsoft® Internet Explorer® browser.

This document covers the installation of Active Library Explorer Server on Unix® platforms. A separate document describes how to install the product on the Windows® platforms.

1.1 Functional Overview

The Active Library Explorer runtime architecture is described in Figure 1. The architecture contains five main components: a web browser, a web server (HTTPD), and the executables `alex`, `alexserv`, `alexhint` and `omnidaemon`.

Of these, the web browser and web server are not included in the Active Library Explorer Server product, and hence the installation of them is not described in this document.

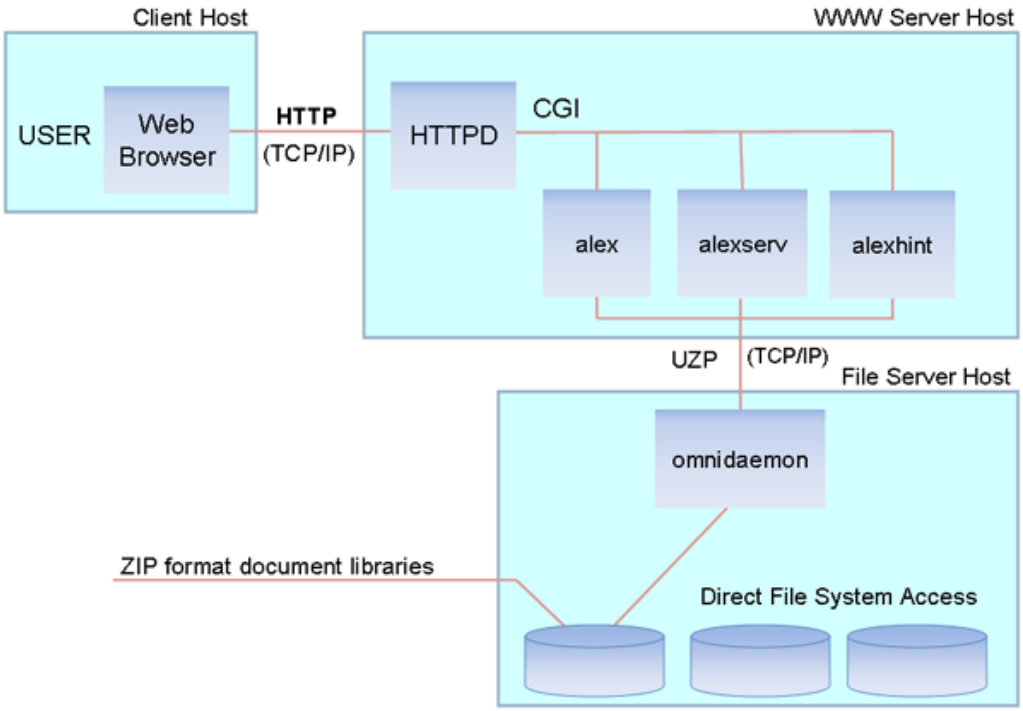

Figure 1 Active Library Explorer - Runtime Architecture

1.1.1 alex & alexserv & alexhint

The `alex`, `alexserv` and `alexhint` executables are installed on the web server as CGI applications. They act as gateway applications that read information from the document library server (`omnidaemon`) over TCP/IP, and transfer the information to the web browser. This is necessary since the web server cannot directly access documents in the compressed Active Library Explorer library files.

1.1.2 omnidaemon

`Omnidaemon` is a server process without user interface. The program may be compared with an *ftp* daemon. The program is typically started automatically on system boot, and should be running constantly while the system is operational.

The program listens to a local TCP port for incoming network connections, and interacts with clients by executing commands issued by the clients. The commands are defined by a proprietary protocol (UZP) similar to ftp.

Like ftp, the protocol allows transfer of file data from server to client. The `omnidaemon`, however, does not transfer ordinary disk files, but files from compressed ZIP archives. The program can also be thought of as an ‘‘unzip daemon’’, that is, an unzip program, but resident in memory.

1.2 System Requirements

Table 1 System Requirements

Operating systems(1)	Solaris™ 8, 9 and 10 SPARC Red Hat® Linux® 6.2 Solaris™ 10 x86
Recommended min. memory size	256 MB
Disk space	4 MB for Active Library Explorer programs App. 10-300 MB for each library file
Browser	Microsoft Internet Explorer 7 or later with Adobe Reader plug-in Firefox 1.0 or later with Adobe Reader plug-in Mozilla® 1.7 or later with Adobe Reader plug-in
Web server	A web server with CGI capabilities
Supported command handler	WinFIOL 4.0 or later

(1) These are the operating system versions on which Active Library Explorer has been tested. Other versions will probably also work well, but this has not been verified.

2 Installation

If you are upgrading an existing Active Library Explorer installation, see Section 3.

Installing Active Library Explorer on a new machine is done in the following five steps:

1 Unpack the installation package

2 Copy the files to assigned directories

3 Check the configuration parameters

4 Start the `Omnidaemon`

5 Verify the installation

Each step is described in subsequent sections:

2.1 Unpack the installation package

Copy the tar file for the selected platform to some suitable directory. Unpack the tar file with the command:

```
tar xvf alexserv_VVV_XXX.tar
```

(replace VVV above with version “13.2”, “14.3”, ... and replace XXX with “sol” or “linux” according to platform)

This will create a directory named ALEX containing all needed files. Follow the instructions in the subsequent sections in this document to install all components of Active Library Explorer Server.

2.2 Copy the files to assigned directories

2.2.1 alex, alexserv, alexhint & alex.conf

- Copy the `alex`, `alexserv` and `alexhint` binaries into the web server host location. Typically, the web server configuration contains a particular directory for CGI applications. Often this directory is named `cgi-bin`. This holds good for Apache™ web server default configuration settings also. Binaries can be copied directly into `cgi-bin` folder or some other sub-directory under it. All three binaries should have *execute* permission for all.

- Copy the extracted `alex.conf` file which contains the Active Library Explorer Server configuration settings into the folder in which it can be found by `alex`, `alexserv` and `alexhint` as described in the next step.

- `alex`, `alexserv` and `alexhint` will look for `alex.conf` file in current directory, followed by `/usr/local/etc`. In case this configuration file is not found, `alex`, `alexserv` and `alexhint` will assume default values for all the keywords in this file. More information on the keywords of `alex.conf` is provided in Section 4.1.

- Assuming that installation is being done under `cgi-bin` directory, the executables and configuration files should be installed according to the following example:

```
cp alex /usr/local/httpd/cgi-bin/alex
chmod 755 /usr/local/httpd/cgi-bin/alex
cp alexserv /usr/local/httpd/cgi-bin/alexserv
chmod 755 /usr/local/httpd/cgi-bin/alexserv
cp alexhint /usr/local/httpd/cgi-bin/alexhint
chmod 755 /usr/local/httpd/cgi-bin/alexhint
cp alex.conf /usr/local/httpd/cgi-bin/alex.conf
```

Apache configuration example

To enable shorter and more user friendly URL addresses, a script alias can be defined for the CGI applications in the web server configuration.

For the Apache server, go to the directory for configuration files, and edit the file `srm.conf`, or equivalent. The file `srm.conf` is specified in the main configuration file (`httpd.conf`) by the `ResourceConfig` keyword. Search the `srm.conf` file for a ScriptAlias keyword. Add a ScriptAlias for `alex`, `alexserv` and `alexhint`, as shown below:

```
ScriptAlias /cgi-bin/    /usr/local/httpd/cgi-bin/
ScriptAlias /alex        /usr/local/httpd/cgi-bin/alex
ScriptAlias /alexserv    /usr/local/httpd/cgi-bin/alexserv
ScriptAlias /alexhint    /usr/local/httpd/cgi-bin/alexhint
```

Optional configuration

Another issue to consider for Apache is the `'KeepAlive'` feature. Experience has shown that Internet Explorer sometimes works very slowly when this feature is enabled. The effect can be that the first page of Active Library Explorer takes a long time to load. If this problem is seen with the basic settings, then it is recommended to turn the `'KeepAlive'` feature off in Apache.

2.2.2 omnidaemon & omnidaemon.conf & sys_structure.conf

- Copy the extracted `omnidaemon` binary into a separate directory. `Omnidaemon` needs to be configured and started on a file server containing

all the library files. This can of course be the same machine in which `alex`, `alexserv` and `alexhint` are present.

- `omnidaemon` can be configured with an optional configuration file, usually named `omnidaemon.conf` where all the necessary configuration settings are present. See Section 4.2 for the description of various parameters present in the `omnidaemon.conf` file. A default configuration file is provided with the installation package. Copy the configuration file into a suitable location and customize it according to the requirements. The file can be copied into any location. The path of this configuration file should be provided to `omnidaemon` with `"-c"` option while starting.

- `Omnidaemon` can also use an optional system structure configuration file, usually named `sys_structure.conf`. This file defines a hierarchical structure of folders, where the library files are stored. See Section 4.2.10 for more information on the `sys_structure.conf` file.

- Assuming that `omnidaemon` is installed under `/usr/local/bin`, following is an example of set of commands to install the `omnidaemon files`:

```
cp omnidaemon /usr/local/bin/omnidaemon
chmod 755 /usr/local/bin/omnidaemon
cp omnidaemon.conf /usr/local/bin/omnidaemon.conf
cp sys_structure.conf /usr/local/bin/sys_structure.conf
```

2.2.3 alex_help.ahx

Copy the `alex_help.ahx` help file available in the installation package to the root library folder, that is either to the directory name as specified in `omnidaemon.conf`, or to the top level folder in the `sys_structure.conf` file. If `sys_structure.conf` is being used then ensure that the correct directory path is specified as the value for the `Path` attribute at `Level` "0". Please see Section 4.2.4 for how to specify these directories.

Note: If `alex_help.ahx` is not copied to the specified location, then a warning message **"The help file is not available! Please refer to the installation guide."** is given when entering Active Library Explorer.

2.3 Check the configuration parameters

In an installation where default values are used for all configuration parameters, the configuration files are not needed. If a default installation is not acceptable, any parameter value can be changed in the configuration files, as described in Section 4.

The following table lists the most important parameters and their default values:

Table 2 Parameters

Parameter	Default value	Notes
Web server port number (http)	80	See Section 4.1.3
Address of document server (omnidaemon)	localhost:9229	See Section 4.1.1 and Section 4.2.1
Path to directory containing the ALEX libraries	$HOME/libraries	$HOME = Home directory of user name under which omnidaemon is running. See Section 4.2.4. **Note:** If the libraries are to be shown in a folder structure in the GUI, configuration files are needed to define this. See Section 4.2.10.
Path to directory containing the annotation database	$HOME/tmp	Must be a writable directory. See Section 4.2.6
omnidaemon log file	$HOME/tmp/omnidaemon.log	See Section 4.2.7

For more information on configuring Active Library Explorer, see Section 4.

2.4 Start the omnidaemon

Sample commands for starting the `omnidaemon` with and without configuration file:

```
/usr/local/bin/omnidaemon
```

Starting this way, the `omnidaemon` looks for a configuration file with the name `omnidaemon.conf` in the current working directory. If not found, the default values for all the keywords in the configuration file will be assumed. To use an explicit configuration file, start the `omnidaemon` with the following command:

```
/usr/local/bin/omnidaemon -c /usr/local/bin/omnidaemon.conf
```

In this case, if the file is not found, the defaults are assumed automatically.

After starting the omnidaemon, if everything works as it should, no error messages will be displayed, and the shell prompt returns. omnidaemon starts by building a RAM cache of the library data. When this is ready, a second process is started which monitors the hard disk for new or updated libraries. Only then is omnidaemon ready to accept client requests. Each request spawns a new process, which dies when the request has been fulfilled. All processes share the same RAM cache.

See Section 4.2 for more information on the omnidaemon configuration settings. Section 4.2.11 describes a way to reduce omnidaemon startup time for large installations.

Errors and warnings detected during execution of omnidaemon are written to the log file (See Section 4.2.7). It can be viewed, for instance, with the more or less commands in a Unix shell window.

Note: Started this way, the omnidaemon will not automatically start on system boot. A suitable command has to be placed in the operating system startup files.

It is recommended to keep a backup copy of the default configuration files so that if any configuration mismatch happens, it is easy to start from the default configuration.

omnidaemon start on system boot

To avoid having to start the omnidaemon manually after every system reboot, the machine can be configured to start the program automatically. This can be done using the rc or rc.local file, or the /etc/init.d directory, for example. Consult your operating system documentation for the correct place to add local tools to be started at system boot.

Note: It is inadvisable to allow the program to run as the root user. It is recommended (but not mandatory) to define a dedicated user and group id under which omnidaemon will run. See Section 4.2.3 for more information on User and Group keywords to change the identity of the running process.

2.5 Verify the installation

2.5.1 Installing an Active Library Explorer Library

A library file is installed simply by copying it to a directory specified in the omnidaemon configuration file (LibraryPath keyword) or in the system structure configuration file (Path keyword). omnidaemon will detect the library automatically, typically within a couple of minutes. If the path specified in LibraryPath is not present, then omnidaemon will look for the libraries in $HOME/libraries. See Section 4.2.4 for more information on the Library Path.

It is also possible to force the `omnidaemon` to perform a check for new libraries, See Section 4.2.8.

2.5.2 Viewing the Libraries Using a Web Browser

To view documents in Active Library Explorer libraries, enter the web address (URL) of the alex program in the address (location) field of the web browser. The address has the form:

```
http://<address of the web server>/alex
```

When the above address has been entered into the web browser, the Active Library Explorer user interface is loaded into the browser window. To find a library, perform a search or open the relevant folder in the system structure. Clicking on a library name will show the main page of that library in a separate window.

For more information on Active Library Explorer features, see the online help text.

2.5.3 Trouble Shooting

In case of problems, the first step is always to check the log files, both the `omnidaemon` and the web server log files. For a new installation the most common error is that the `LibraryPath` keyword does not contain a valid file path.

3 Upgrading From Earlier Versions

If the new version of Active Library Explorer is replacing an existing installation of Active Library Explorer, and there is no need for a major reconfiguration, the following simple procedure can be followed:

1 Copy the new `alexserv` executable to replace the existing one. Copy the `alex` and `alexhint` executables to the same directory, replacing any existing files with the same names (in earlier versions of Active Library Explorer, one or both of these files may not exist).

2 Stop the running `omnidaemon` process.

3 Copy the new `omnidaemon` executable to replace the existing one.

4 Check that the keywords in the `omnidaemon` configuration file (typically called `omnidaemon.conf`) are still valid, and remove any obsolete keywords. Use Section 4.2 as reference. Check the `alex` configuration file `alex.conf` in a similar manner using Section 4.1 as reference.

5 Restart `omnidaemon`.

For more details about installing the necessary files, see Section 2. For a full description of how to configure Active Library Explorer, see Section 4.

4 Configuration File Reference

Active Library Explorer binaries can be configured using three configuration files. These files (typically `alex.conf`, `omnidaemon.conf` and `sys_structure.conf`) consist of space or tab separated name-value pairs on one line.

Following is a brief overview of each configuration file and its usage:

1 **alex.conf:** This file can be used to define various configuration settings for the Active Library Explorer Server. The `alex`, `alexserv`, and `alexhint` executables read this configuration file at the time of execution. The settings include the address of the `omnidaemon` document server, the name of the web server host, the web server port number, and so on. If this file is not present then `alex`, `alexserv` and `alexhint` will assume default values for all the settings.

2 **omnidaemon.conf:** This file contains various parameters needed by the `omnidaemon` document server. The parameters include the port in which `omnidaemon` should run, the server comment (will be displayed on the title bar of the browser), library path denoting the path to the library files, and so on. If this file is not present then `omnidaemon` will assume default values for all the parameters.

3 **sys_structure.conf:** This is an optional file, where a hierarchal structure of folders can be defined. Folders can be defined on several levels and normally the leaf folders contain the libraries. The defined tree structure will be displayed in the ALEX user interface, resembling the Windows file explorer. If used, the path to this file should be set as value of `LibraryPath` in “omnidaemon.conf” file.

The following sections describe the usage of the configuration files and keywords:

4.1 alex, alexserv & alexhint

Once the web server configuration has been modified and the `alex`, `alexserv` and `alexhint` executables have been installed in the CGI directory as described in the preceding sections, the configuration file can be edited.

4.1.1 Address of Document Server

The `omnidaemon` keyword specifies the hostname (or IP-address) and port number of the default document server, according to the following example:

```
# Address of the omnidaemon document server
Omnidaemon: localhost:9229
```

Modify the hostname above to match the address of the host on which the document server is installed. The port number (9229 in the example) must match the value of the `WellKnownPort` keyword of the `omnidaemon` configuration file.

In case `omnidaemon` keyword in `alex.conf` is not present or `alex.conf` itself is not present, then the default value `localhost:9229` will be assumed for `omnidaemon` keyword. This is also the value present in the `alex.conf` file provided in the installation package.

4.1.2 Name of Web Server Host

The `HttpdHost` keyword is used to provide the full domain name of the web server host where Active Library Explorer is being installed. This information is used by `alex`, `alexserv` and `alexhint` for URL translation.

If the `HTTP_HOST` or `SERVER_NAME` environment variable is correctly set by the web server, the `HttpdHost` keyword may be omitted from `alex.conf`.

The order of precedence followed by Active Library Explorer Server when using these parameters to determine the name of the web server host is:

1. HTTP_HOST
2. HttpdHost - If `HTTP_HOST` does not contain the host name.
3. SERVER_NAME - If `HTTP_HOST` and `HttpdHost` do not contain the host name.

As the `HTTP_HOST` or `SERVER_NAME` environment variable is set by almost all web servers including Apache, it is better to allow the Active Library Explorer Server to take the value from this variable. Hence, this keyword can normally be omitted from the `alex.conf` file.

4.1.3 Port Number of Web Server Host

The `HttpdPort` keyword is used to provide the TCP/IP port number of the web server. This information is used in URL translations in `alexserv`.

If the `HTTP_HOST` or `SERVER_PORT` environment variable is correctly set by the web server, or the web server is on default port 80, then the `HttpdPort` keyword may be omitted from `alex.conf`.

The order of precedence followed by Active Library Explorer Server when using these parameters to determine the port number of the web server host is:

1. HTTP_HOST - If the `HTTP_HOST` variable is set with the server name, but does not contain any port number, then Active Library Explorer Server assumes that the web server is on default port 80.
2. HttpdPort - If the `HTTP_HOST` does not contain the port number.

3. SERVER_PORT - If `HTTP_HOST` and `HttpdPort` do not contain the port number.

As the `HTTP_HOST` or `SERVER_PORT` environment variable is set by almost all web servers including Apache, it is better to allow the Active Library Explorer Server to take the value from this variable. Hence, this keyword can normally be omitted from the `alex.conf` file.

4.1.4 Path to Directory for Temporary Files

The `TempPath` keyword is used to specify a directory where `alexserv` can write temporary files (used in the Compare function). If this path is not specified in the configuration file, `alexserv` tries the path in the `TMPDIR` environment variable, and if that fails, a predefined directory path `/tmp` will be tried. The order of precedence is the following:

```
1. TempPath keyword in configuration file
2. $TMPDIR environment variable
3. /tmp -directory
```

Important:

`alexserv` must have write access to the specified directory. There must also be some disk space available, typically a few MB for each execution of the Compare function. The temporary files are very short-lived, so the need for permanent disk space is not very large.

4.1.5 Support for Switching Document Language

Active Library Explorer offers the user a function to switch from a library in one language to the same library in another language – and even to the corresponding document in this library. The prerequisite for this function is that the two libraries must have the same library identity, except for the language code.

The library identity is shown on the page listing all libraries, typically it starts with `LZN...`, or `XX/LZN...`, where `XX` is the language code. If the language code is missing, the language will be taken as ‘‘English’’.

Active Library Explorer displays the available language codes in a drop-down list in the toolbar, if this option is turned on. The list is built based on the actual language codes found in the libraries. The two-character language codes are not that user friendly, however. This can be improved by specifying a configuration parameter that maps language codes to plain text. The contents of the parameter are a series of value pairs separated by semicolons (‘;’). For example:

```
LanguageCodes:  es=Spanish (ES); ja=Japanese (JA);
fr=French (FR); pb=Portuguese (PB)
```

A predefined set of language definitions are provided in the sample configuration file, so normally the value of this parameter need not be changed. Note also that English need not be configured, it is defined by default.

If the `LanguageCodes` keyword is not given, the list of languages will not be shown in the toolbar.

4.1.6 Support for Japanese Character Encoding

When browsing documents in a Japanese environment, the character encoding of the XHTML pages must be explicitly set so that Japanese characters can be entered for example, in the search field. To do this, the following line must be entered into `alex.conf`:

```
CharEncoding:  EUC-JP
```

It is **not** recommended to use this setting when European specific characters are to be used. EUC-JP encoding is still good for English input but is fault sensitive to additional 8-bit coded characters (for example Spanish, Scandinavian).

4.1.7 Sample Configuration File

Below is a complete example of the `alex.conf` file included in the installation package:

```
#
# Active Library Explorer Server configuration file
#
# Note:  Lines starting with a hash (#) character are
# commented out (inactive) keywords.

# omnidaemon:      localhost:9229
# HttpdHost:       localhost
# HttpdPort:       :8080

# TempPath:        "/tmp/alextemp/"

# LanguageCodes:   fr=French(FR); ja=Japanese(JA);
# CharEncoding:    EUC-JP
#
```

4.2 omnidaemon

Before the `omnidaemon` program is started, a valid configuration file must be prepared by editing the sample file included in the delivery. The configuration file is read on startup, and each time the document server is reconfigured.

Reconfiguration happens when the omnidaemon process is sent a hang-up signal (Unix command: kill –HUP <pid>).

Note: Keyword values that contain blank characters must be quoted in the configuration file.

4.2.1 Port Number

The WellKnownPort keyword is used to specify the port number on which omnidaemon will listen for incoming connections. The default value for this keyword is 9229. This is the value present in the omnidaemon.conf provided along with the installation package. The same value will be taken for WellKnownPort, in case configuration file is not present.

```
WellKnownPort          9229
```

4.2.2 Server Comment

The ServerComment keyword is used to identify the site, for example the name of the department, or a common descriptive name for all available libraries, and so on. The default comment provided in the omnidaemon.conf file provided with the installation package is ‘‘Active Library Explorer’’. This can be edited to match the comment that is relevant for the respective Active Library Explorer Server installation. In case the omnidaemon.conf file is not present, the default comment ‘‘Active Library Explorer’’ will be taken.

```
ServerComment “Active Library Explorer”
```

This text string is shown in the browser window title.

4.2.3 User and Group

The User and Group keywords are used to specify the user and group under which omnidaemon should run. If the omnidaemon is running under the current Unix user, these keywords can be commented out.

```
# User          daemon
# Group         daemon
```

If the User and Group keywords are enabled, the omnidaemon process will try to change its user identity accordingly during startup. This operation usually works only if the process is started as the root user.

4.2.4 Library Path

The LibraryPath keyword can specify one of two things: a system structure configuration file (see Section 4.2.10) or the directory where the library files are stored. The default value present in the omnidaemon.conf file provided with the installation package is /data/alex/libraries. In case this

path is not present or the directory is not accessible, the directory will be taken as `$HOME/libraries`. This is the directory assumed in case the `omnidaemon.conf` file itself is not present.

```
LibraryPath     "/data/alex/libraries/sys_structure.conf"

or

LibraryPath     "/data/alex/libraries/"
```

The use of local disks is recommended for performance reasons.

4.2.5 Allowing Connections from remote clients

For security reasons, `omnidaemon` by default will deny connections from servers other than the local host.

The following statement need to be added in the configuration file to enable `omnidaemon` to allow connections from other servers.

```
IPMask          "Allow [ip-address]"
```

In the *[ip-address]* specify the actual ip-address of the client from which connections should be accepted. For each client a separate entry needs to be added. You can also use wild-card character '*' in the ip-address (for example, 172.16.*.*).

If a client that doesn't have an IPMask entry in the configuration file tries to access `omnidaemon`, then a warning message in the following format will be logged into the log file (see Section 4.2.7).

```
Warn:  client denied based on the address, [ip-address]
```

4.2.6 Annotation Database

Annotations provide a means for the reader of a document to add their own comments and additions to a document. Annotations are public (all users of the same Active Library Explorer installation can see them) and are stored into a separate database.

The location for this database is specified with the `AnnotationDb` keyword. The specified directory must have suitable write privileges assigned to it, and it may not contain any document libraries. The annotation database itself is created when the first annotation is added. The value present in the `omnidaemon.conf` file provided with the installation package is `/data/alex/writable`. In case this path is not present or is not accessible, it will be taken as `$HOME/tmp`. This is also the value assumed in case the `omnidaemon.conf` file itself is not present.

```
AnnotationDb     "/data/alex/writable/"
```

4.2.7 Log File

Omnidaemon logs events and errors to a log file during execution. The location of this file is specified with the LogFile keyword. The value present in the omnidaemon.conf file provided along with the installation package is /usr/tmp/omnidaemon.log. In case this directory is not present or is not accessible, the directory will be taken as $HOME/tmp/omnidaemon.log. This is the path assumed, in case the omnidaemon.conf file itself is not present.

```
LogFile          "/usr/tmp/omnidaemon.log"
```

When troubleshooting, this file should always be consulted. It can be viewed, for instance, with the *more* or *less* commands in a Unix shell window.

4.2.8 Adding New or Updated Libraries

Omnidaemon repeatedly scans the specified disk directories to check if libraries have been added or updated, so typically you only have to wait for a few moments before the new library is available. This applies also to changes in the system structure configuration file (see Sections 4.2.4 and 4.2.10). Changes to the omnidaemon configuration file (omnidaemon.conf) are, however, not automatically detected, so omnidaemon must be explicitly ordered to update itself in this case.

There are two ways of ordering an omnidaemon update:

1 By sending the Unix USR1 signal. This causes a rescan for new or changed libraries, and the system structure information will also be updated. This signal is needed only if the automatic update has failed to notice a change.

2 By sending the Unix HUP signal. This causes omnidaemon to discard all information in the RAM cache, then reread the configuration file, and finally rebuild the RAM cache from scratch. This signal is needed when you want to apply changes in the configuration file (omnidaemon.conf).

The example below shows how to send an USR1 signal to the running omnidaemon process. By executing the commands outlined below, the omnidaemon will immediately rescan the file-system to update the information in the RAM cache.

```
cat /tmp/omnidaemon.pid

2548

kill -USR1 2548
```

To verify that the omnidaemon performed the action, the log file can be viewed:

```
tail -2 /usr/tmp/omnidaemon.log
[2548] 2000-01-05 14:41:19 Note:
       Update ordered by child or external program
```

```
[2548] 2000-01-05 14:41:19 Note:
       Refresh finished, 0 changed, 0 new databases
```

Note that while `omnidaemon` is refreshing its RAM cache, it will not respond to user (`alex`, `alexserv` or `alexhint`) requests for a short time period, depending on the number and the size of the libraries.

4.2.9 Sample Configuration File

Below is a complete example of the `omnidaemon.conf` file included in the installation package:

```
#
# omnidaemon configuration file
#
# Note: Lines starting with a hash (#) character are
# commented out (inactive) keywords.

# WellKnownPort           9229
# ServerComment           Active Library Explorer

# User                    daemon
# Group                   daemon

# LibraryPath             "/data/alex/libraries/"

# AnnotationDb            "/data/alex/writable/"

# LogFile                 "/usr/tmp/omnidaemon.log"
```

4.2.10 System Structure Configuration

If a flat list of libraries is not desired, it is possible to organize document libraries in a hierarchical structure. To use this feature, a system structure configuration file has to be specified in `omnidaemon.conf` with the `LibraryPath` keyword.

For example, libraries can be organized into a system structure as below:

```
Ericsson, Finland
   GSM System Library, R7
           gsm_sys1.alx
           gsm_sys2.alx
           Switching System R7
               switch_s1.alx
               switch_s2.alx
               MSC
                  msc_lib.alx
           OSS
               oss_lib.alx
```

```
GSM System Library, R8
        Switching System R8
            switcha.alx
```

Note: Entries marked with normal text in the example above are document libraries, bold text indicates folders in the library structure.

The system structure configuration file has one entry, consisting of five name-value pairs, for each folder. Each entry must specify the following information in this order:

- Name of the folder (for example, "`Ericsson, Finland`").
- Unique label for a folder (not currently in use).
- Level of the folder. This is a numerical value starting with 0 for the root folder and increasing with one for each sub-level (In the example above; 0 for "`Ericsson, Finland`", 1 for "`GSM System Library, R7`" and 2 for "`Switching System R7`").
- The disk directory where the libraries of the folder are stored. *Note: A folder can have libraries in only one directory.*
- Main page for folder (not currently in use).

The entries have to be organized hierarchically, that is, after each system entry come **all** its subsystem entries before the next entry on the parent's level. The information for each entry must be in the order as above. Text strings and file paths **must** be enclosed within double quotes if spaces are used.

Below is a complete listing of a system structure configuration file, based on the information in the example above.

```
#
# omnidaemon system structure configuration file
#
# Note: Lines starting with a hash (#) character are
# commented out (inactive) keywords.

Name:         "Ericsson, Finland"
Label:        "-"
Level:        0
Path:         /data/alex/libraries
IndexFile:    -

Name:         “GSM System Library, R7”
Label:        “GSM R7”
Level:        1
Path:         /data/alex/libraries/GSMR7
IndexFile:    -

Name:         “Switching system R7”
Label:        “SSR7”
Level:        2
Path:         /data/alex/libraries/GSMR7/SwitchS
IndexFile:    -

Name:         “MSC”
Label:        “MSC”
Level:        3
Path:         /data/alex/libraries/GSMR7/SwitchS/MSC
IndexFile:    -

Name:         “OSS”
Label:        “OSS”
Level:        2
Path:         /data/alex/libraries/GSMR7/OSS
IndexFile:    -

Name:         “GSM System Library, R8”
Label:        “GSM R8”
Level:        1
Path:         /data/alex/libraries/GSMR8
IndexFile:    -

Name:         “Switching System R8”
Label:        “SSR8”
Level:        2
Path:         /data/alex/libraries/GSMR8/SwitchSR8
IndexFile:    -
```

4.2.11 Optimizing the omnidaemon startup time

In a typical installation `omnidaemon` will start up in a matter of seconds (depending on processor speed and load, of course). Also, since `omnidaemon` does not normally have to be restarted very often, the startup time is of little significance. This section describes a special case where the startup time may increase, and gives a solution for how to avoid it.

The tree synchronization feature was introduced in Active Library Explorer version 8. This feature automatically opens and highlights the folder in which a document resides, as the document is opened in the browser. Thus it greatly helps the user to navigate in the document library folder structure. This feature works in the Internet Explorer browser, only.

This feature needs a special index file in each library. All new libraries will have this index file automatically included. For older libraries, `omnidaemon` can generate this index file at runtime. This is done when `omnidaemon` is started, and it will of course somewhat slow down the startup time. For small libraries the effect is negligible, but for a very large library (300-400 MB) the indexing may take up to a minute (again, depending on processor speed and load).

When generated during startup, the index files are kept in memory, only, since `omnidaemon` does not have write access to the library files in normal operation. Thus, if `omnidaemon` is restarted, the index files need to be regenerated.

It is possible to update the library files with this index file, once and for all, by running `omnidaemon` as a command line utility. The syntax for running the utility program is:

```
omnidaemon -i [-c omnidaemon.conf]
```

assuming `omnidaemon` and `omnidaemon.conf` reside in the current directory. In case the -c option is omitted, `omnidaemon` will update the libraries in the default directory `/data/alex/libraries/`. If the -c option is used, `omnidaemon` will update the libraries according to the configuration specified in the file. Only the libraries lacking the index file will be updated.

It is recommended to stop the normal `omnidaemon` service while this update run is being executed. After the libraries have been updated, the `omnidaemon` utility will exit automatically.

If some users are using the Active Library Explorer for Windows, the “standalone” browser that can be used on a PC without a web connection, it is strongly recommended to update all libraries using the `omnidaemon` command line utility. Active Library Explorer for Windows can not generate the needed index file at runtime, so the tree synchronization feature will not work for libraries where this index file is missing.

Glossary

CGI
Common Gateway Interface, a method for web servers to dynamically generate web pages via external software.

FTP
File Transfer Protocol

HTML
Hyper Text Markup Language, a mark-up language for online content.

IP
Internet Protocol

PDF
Portable Document Format. A PostScript-based document format defined by Adobe®.

URL
Uniform Resource Locator. An address on the World Wide Web.

ZIP
File compression format, originally created by PKWare Inc.