Contents

1 Legal issues **6**

1.1 Copyright 6

1.2 Disclaimer 6

1.3 Licence 6

1.4 Use 6

1.5 Distribution 7

2 How to register **8**

2.1 Registerable freeware 8

2.2 Payment 8

2.3 Registration form 9

3 Introduction to AWeb **10**

3.1 Why not use MUI ? 11

3.1.1 Visual feedback 11

3.1.2 Intuitivity 12

3.1.3 More intuitivity 12

3.1.4 Minor objections 13

3.1.5 Flames? 13

3.2 ClassAct 13

3.3 System requirements 13

3.4 Installation 14

3.4.1 Installing AWeb 14

3.4.2 Installer problems 15

3.4.3 Configuring the JFIF datatype 15

3.5 Tips for 2MB Amiga users 15

3.6 Starting AWeb 16

4.5 The network status window . . . 28
4.5.1 Purpose . . . 28
4.5.2 Opening the window . . . 28
4.5.3 Contents of the list . . . 28
4.5.4 Cancel transfers . . . 28
4.6 User authorization . . . 29
4.6.1 Authorization . . . 29
4.6.2 Save your authorization details . . . 29
4.7 Compatibility mode . . . 29
4.7.1 Incorrect HTML . . . 29
4.7.2 Differences . . . 30
4.8 Settings . . . 30
4.8.1 Controlling the settings requester . . . 30
4.8.2 Browser 1: Font settings . . . 30
4.8.3 Browser 2: Appearance . . . 30
4.8.4 Screen 1: Screen . . . 31
4.8.5 Screen 2: Windows . . . 32
4.8.6 Screen 3: Palette . . . 33
4.8.7 Network 1: General . . . 33
4.8.8 Network 2: Proxy . . . 35
4.8.9 Network 3: External programs . . . 35
4.8.10 Program 1: General . . . 36
4.8.11 Program 2: External programs . . . 37
4.8.12 MIME types and external viewers . . . 38
4.8.13 Example . . . 39
4.8.14 Closing the requester . . . 40
4.9 ARexx interface . . . 40
4.9.1 ARexx port names . . . 40
4.9.2 ARexx commands . . . 41
4.9.3 Return values from commands . . . 41
4.10 Shell command and ARexx macro interface . . . 42
4.10.1 Execute shell commands and ARexx macros . . . 42
4.10.2 Simple shell commands . . . 42
4.10.3 ARexx macros . . . 42
4.10.4 Parameters . . . 43
4.10.5 Load the result back into AWeb . . . 43

Chapter 1

Legal issues

1.1 Copyright

1.2 Disclaimer

This software is provided "as is". No warranties are made, either expressed or implied, with respect to reliability, quality, performance, or operation of this software. The use of this program is at your own risk. Yvon Rozijn assumes no responsibility or liability for any damage or losses resulting from the use of this software, even if advised of the possibility of such damage or loss.

1.3 Licence

This licence does not apply to parts of the distribution not covered by the AWeb copyright. For the licence of these parts, refer to the respective documentation.

1.4 Use

AWeb is registerable **freeware**. You may use this program for an unlimited period even without registration (see chapter 2).

Chapter 2

How to register

2.1 Registerable freeware

AWeb is what I call *registerable freeware*. This means

- You don't *have* to pay for AWeb. It is perfectly legal to use this version of AWeb for an unlimited period without paying. This is because I really don't want to force you to pay for a HTML-2 only browser, that can't handle tables, background pictures and other cool HTML-3 features.
- On the other hand, if you want to express your appreciation, you *can* pay for AWeb, and I will send you a personalized keyfile in return. With that keyfile, you will get a bonus function: multiple windows.

Note that future versions of AWeb will be proper shareware, or even commercial. The freeware registration file will be valid for future shareware versions.

2.2 Payment

Registering AWeb costs US $ 35, NLG 50, DEM 50, or GB £25.

You will receive your keyfile by e-mail. If you want your keyfile sent by snail-mail on a diskette, it costs you US $ 5, NLG 7, DEM 7 or GB £3 extra.

You can pay me in one of the following ways:

- Cash. This is the cheapest and easiest for both of us. Please wrap your banknotes in a sheet of paper so they are not visible from outside the envelope. Please send your cash together with a printed registration form. Please refer to the on-disk documentation for information on how to print a registration form. **Do not send coins!** If you want the keyfile on diskette, you may have to round up the amount.

Chapter 3

Introduction to AWeb

AWeb is a World Wide Web browser for the Amiga computer.

It offers the following features:

- AWeb doesn't use Magical User Interface (MUI). It uses BOOPSI classes instead, which results in less memory usage and increased speed. The BOOPSI classes are partially custom designed for AWeb, and partially from the ClassAct kit.

- AWeb can use a wide range of TCP-stacks: AmiTCP/IP, I-Net225, AS-225 or compatible. Without TCP stack running, you can still view local files.

- AWeb uses extensive internal multitasking, which leads to total asynchroneous and parallel network access. Images are starting to load while the document is still loading. It is possible to follow a link while the previous document is still loading. A separate network status window shows all pending network and local file accesses. All network accesses can be interrupted *immediately*.

- The HTML-2 standard is fully supported, including forms. For buggy pages not conforming to the standard, AWeb offers a compatibility mode.

- AWeb supports the use of proxies for HTTP, FTP, Gopher and Telnet. Because some proxies can't handle forms or other special cases, the use of proxies can be temporarily disabled from the menu.

- The HTTP and Gopher protocols are supported internally. By using external programs, FTP and Mailto can be used too. HTTP user authorization is supported.

- AWeb lets you load all images, delay all image loading, or load only clickable maps and delay other images. For delayed images, the ALT text is taken into account. Transparent GIFs are supported. The 24-bit picture datatype (picturedtV43) is supported.

To understand the consequences, imagine an application that is busy performing a task that takes a considerable amount of time. Meanwhile, it offers the user an interface to affect the current task (a common example is a cancel button). It checks the user input every second. Or imagine an application busy performing a short task, taking less than a second. The GUI isn't disabled, allowing the user to select another GUI element during this short time. This user action is than queued for a fraction of a second until the application is ready to process it. This is of course the way most applications work.

Both are examples of totally acceptable application response, providing that there is immediate visual feedback for the user action. There is a difference between the visual feedback of a user action, and the application response to that action. The former must be immediate, while the latter in general may take a short time, up to a second or so, without confusing the user. (Depending of the kind of application, of course.) MUI applications treat visual feedback as being an application response, thereby failing miserably.

It should be obvious that blocking user input entirely and showing a busy pointer, like some people suggest, is not an option at all in the above examples.

I realize some gadtools gadgets have the same behaveour as MUI gadgets, and that's one of the reasons why I don't like gadtools very much, either. But at least gadtools *buttons* give immediate feedback.

3.1.2 Intuitivity

Gadtools offers a cycle gadget. Click anywhere in the gadget, and the next option is selected. You may like or may not like this kind of gadget, but the behaveour exists and is part of the Amiga look and feel.

MUI offers a gadget that looks exactly the same, but behaves differently. Click in the text part of the gadget, and nothing happens. Just a quick flash of a pop-up menu. I have clicked many times on gadgets of this kind and wondered why nothing changes before I realized it was a MUI application so this kind of gadget behaves differently.

The most important rule in GUI design is: be consistent. Gadgets that look the same must behave the same. Gadgets that behave differently must look different. MUI fails to be consistent with the Amiga OS look and feel.

I am aware that there are commodities that change the behaveour of the Gadtools cycle gadget into a pop-up menu. Basically these commodities suffer from the same fault: it changes the behaveour of the gadget but not its appearance.

3.1.3 More intuitivity

Fortunately, MUI 3.x has left the silly behaveour of configuring all your MUI applications from one preference program. But still, configuring your MUI application can be very difficult and not obvious for the casual MUI user. The best example of this is the ever returning question in many Amiga newsgroups:

- OS 3.0 or better.
- At least 2 MB of memory, preferably more.
- Either AmiTCP/IP, I-Net225 or AS-225 to access the World Wide Web.
- Appropriate datatypes to view inlined images. At least a GIF and a JPEG datatype are needed to view most of the images on the World Wide Web.
- Some classes from the ClassAct kit (included).

3.4 Installation

3.4.1 Installing AWeb

Installing AWeb is straightforward. Just double-click the Install icon.

The archive contains three different Workbench icons for AWeb: for the standard Workbench, for MagicWB, and for NewIcons. The installation process will ask you which you want to install. The other icons are saved in a separate drawer so you can switch icons later.

AWeb needs some ClassAct classes. These are included in the archive, and the installation process will ask you if you want to install them. You are strongly encouraged to do so. If you do not install ClassAct, you have to make sure that AWeb can find its classes, for instance by executing the MakeAssign script before you start AWeb. If you *do* install ClassAct, you don't have to worry about this.

If you haven't saved your settings from within AWeb, the installation procedure will also ask you some questions to be able to install one of the predefined configurations. Note that you can always change the configuration afterwards, using the settings requester.

- First, it asks if you have 2 MB of memory in your Amiga, or more. If you have only 2 MB of memory, then a different setup is needed to get the most out of AWeb. Have look at section 3.5 to see how you should set up AWeb for a 2 MB Amiga.
- Also, you will be asked if you prefer small fonts or large fonts. The large fonts option will make AWeb look as closely as possible to the standard Netscape setup, using only standard Workbench fonts. Although the HTML standard doesn't impose specific fonts in any way, many pages on the World Wide Web are designed to look best using these fonts. If you plan to use AWeb on a small screen (less than 640 x 400), you might want to use the small fonts, or else the window will contain very little text. **Note:** The *large fonts* settings makes use of the *CGTimes* scalable font found on the Workbench Fonts diskette. Make sure you have this font properly installed if you choose *large fonts*.

If you follow these steps, websurfing shouldn't be a problem with only 2MB.

Of course, it will be more comfortable with more memory. Then you can run AWeb on its own screen, using more colours, using a bigger cache, etc.

3.6 Starting AWeb

3.6.1 From the Workbench

You can start AWeb by a double-click on its icon.

AWeb can be specified as *default tool* in a project icon. You can use extended selection (shift-click) to select one or more project icons. AWeb will load the projects selected as local documents.

In addition, AWeb supports the following *tool types*:

URL=*url_or_filename* Specify this tool type one or more times to open these documents when AWeb is started. If you specify a local filename, use the LOCAL tool type too.

LOCAL If this tool type is present, the names in the URL tool types will be interpreted as local file names, rather than network URLs. This tool type is not needed if you select documents by their project icon, as mentioned above.

CONFIG=*settings_filename* Use this file as settings file instead of the default, AWeb.prefs. If the file doesn't exist, some default settings are used. Saving the settings will save to this file name.

HOTLIST=*hotlist_filename* Use this file as AWeb's hotlist instead of the default, AWeb.hotlist. If the file doesn't exist, it will be created the first time you add an entry to the hotlist.

3.6.2 From the Shell

You can start AWeb from the Shell (or the CLI).

Format: AWeb [*url_or_filename*]**...** [**LOCAL**] [**CONFIG** *settings_filename*] [**HOTLIST** *hotlist_filename*]

Template: URL/M,LOCAL/S,CONFIG/K,HOTLIST/K

The documents in the URL argument are loaded when AWeb starts. If the LOCAL argument is present, the names will be interpreted as local file names, rather than network URLs.

The file in the CONFIG argument will be used as settings file instead of the default, AWeb.prefs. If the file doesn't exist, some default settings are used. Saving the settings will save to this file name.

Chapter 4

Using AWeb

4.1 The Graphical User Interface

In figure 4.1 an overview of the graphical user interface is presented. The items depicted in this figure are explained in sections 4.1.1 through 4.1.12.

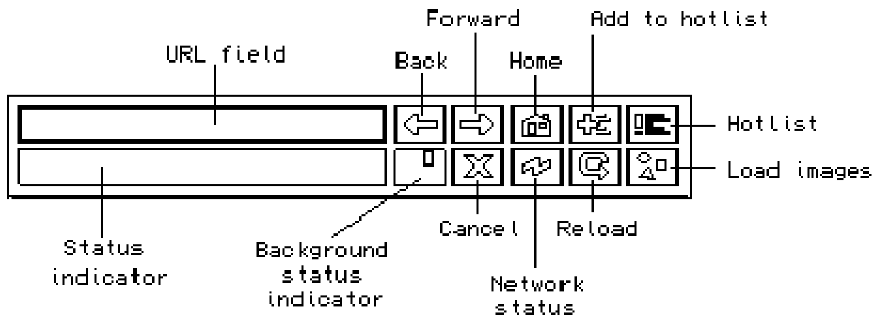

Figure 4.1: An overview of the graphical user interface.

4.1.1 URL field

This field shows the URL (network address) of the currently shown page. You can enter a new URL here, then ENTER will retrieve the page for that URL.

The `Project / Open URL` menu function or its shortcut, **A**`U`[1], will clear this field and activate it, so you can type the new URL right away.

The `Project / Open WWW` menu function or its shortcut, **A**`W`, will activate this field and preload it with "http://www." for even more convenience.

[1] **A** is used throughout this document to indicate the right amiga key.

The `Hotlist / Add to hotlist` menu function, or its shortcut, **AA**, will do the same.

4.1.8 Hotlist button

This button shows your hotlist.

The `Hotlist / Show hotlist` menu function, or its shortcut, **AH**, will do the same.

4.1.9 Cancel button

This button will interrupt (cancel) the load of a page in this window. Background loads cannot be cancelled by this button, use the cancel button in the network status window instead.

The `Control / Cancel load` menu function, or its shortcut, **AX**, and the **Esc** key, will do the same.

4.1.10 Network status button

This button will open the network status window, or bring it to front if it is already open.

The `Control / Network status` menu function, or its shortcut, **A?**, will do the same.

4.1.11 Reload button

This button will reload the current document. The page is deleted from the cache, and retrieved again.

The `Control / Reload current` menu function will do the same.

4.1.12 Load images button

This button will load all unloaded images in the current page.

The `Control / Load images now / All images` menu function, or its shortcut, **AI**, will do the same.

4.2 Menus

4.2.1 Project menu

The *Project menu* offers functions to open or close windows, fetch or save documents, or quit AWeb.

Next window Activates and brings up to front the next window. Only available in the registered version.

Previous window Activates and brings up to front the previous window. Only available in the registered version.

HTML mode Select the way HTML should be interpreted in this window. This menu item has 2 sub-items:

Strict Let AWeb follow the HTML standard.

Compatible Let AWeb be compatible with other browsers that are buggy. Useful for viewing a page that was composed with such a buggy browser.

Disable proxy? If you have proxies configured, you can select this menu item to toggle the use of proxies on and off. This feature is included because some proxies tend to handle forms and authorized requests incorrectly.

4.2.3 Cache menu

The *Cache menu* offers functions to flush the cache, and to save or fluch the cached authorizations.

Flush images in current Inlined images that appear in the current document are deleted from both the memory and disk caches. Note that if the image also appears in another document, it is undisplayed in that document, too.

Flush old images Inlined images that do *not* appear in any currently displayed document are deleted from both the memory and disk caches.

Flush all images The image cache is cleared completely. After this function, no images are left in the cache.

Flush documents All documents that are not currently displayed are deleted from the cache.

Save authorizations Save the current authorization details.

Flush authorizations Forget all the current authorization details. Note that this flushes the internal authorizations cache only. To flush the disk cache too, you have to select Cache / Save authorizations afterwards.

4.2.4 Navigate menu

The *Navigate menu* offers functions to navigate in the document history.

Back Walk back one document through the window history. The window history contains all pages viewed before *in this window*. Pressing the Back button, or using the **Alt + cursor left** key combination, will do the same.

4.2.7 Help menu

The *Help menu* offers you more information.

Documentation Shows the AWeb manual in this window.

AWeb home page Retrieves the AWeb Home page. A TCP stack must be running, and you must be connected to the Internet for this function to work.

AWeb FAQ Retrieves the AWeb FAQ. A TCP stack must be running, and you must be connected to the Internet for this function to work.

4.2.8 ARexx menu

The *ARexx menu* offers functions to start ARexx macros. Currently you can only start a macro from a file selector, but in the future you can configure your own ARexx macro menu.

Start ARexx macro... Opens a standard file requester. After you select an ARexx macro, that macro will be executed with the original window as default port.

4.3 Cache usage

AWeb uses a special caching system, partially in memory and partially on disk, to reduce the number of network accesses.

Note that the AWeb cache is a *cache*, not a *proxy* system. This means that the cache is cleared when AWeb finishes. Also, the cache doesn't take the HTTP expiration date into account. This might change in later versions of AWeb.

4.3.1 Documents

Documents that can be displayed by AWeb are initially loaded in memory. When the document cache is full, the least recently displayed document is swapped out to disk, thereby freeing memory. If a swapped-out page must be displayed again, it is swapped back into memory, thereby probably swapping out other pages.

The swapped documents are stored in AWeb's temporary directory. It should be clear that this would be only meaningful if the temporary directory isn't in RAM. So, if you are using a temporary directory in RAM, be sure to set the *document cache memory size* large enough.

When the total size of swapped out documents is larger than the *document cache disk size*, the least recently displayed documents are deleted.

4.4 The browser window

The browser window is the most important window of AWeb. It is the place where World Wide Web pages are displayed. On top of the window there are some gadgets.

4.4.1 Scrolling the page

You can scroll the window contents horizontally and vertically, provided the size of the document is larger than the window.

Of course you can use the scroll bars and arrow buttons to scroll, but AWeb also understands the following keys:

- **cursor up/down** (both cursor keypad and numeric keypad) scroll up and down over a short distance.
- **shift + cursor up/down** (cursor keypad) scroll up or down a page.
- **PgUp, PgDn** (numeric keypad) scroll up or down a page.
- **Space, Backspace** scroll up or down a page.
- **Home, End** (numeric keypad) jump to the beginning or the end of the document.
- **cursor left/right** (both cursor keypad and numeric keypad) scroll left and right. This is only possible if the document contains an image or preformatted text wider than the window, because otherwise AWeb will keep the text formatted to fit within the window width.
- **shift + cursor left/right** (cursor keypad) scroll left or right a full page width.

4.4.2 Following links

One of the most important features of the World Wide Web is the ability to include *hyperlinks* in documents. A hyperlink is displayed in another colour, and by default underlined. You can change the colour and the underlining in the Browser 2 page in the settings requester.

Images that are also links have a frame drawn around them. If you have deselected the link underlining in the settings requester, then this frame isn't drawn.

Click the text or image to follow the link, i.e. retrieve and display the document "behind" the link. The URL (network address) of the document that is linked to, is shown in the status indicator when the mouse pointer is over the hyperlink.

current from the menu, and shift-click the upper left half of the icon, or click the image to load and display the document, and then select `Project / Save as HTML` from the menu to save the document.

4.5 The network status window

4.5.1 Purpose

The Network Status Window shows all pending network and local file accesses. There are possibilities to cancel selected transfers, or all transfers.

4.5.2 Opening the window

You can open the window in three ways:

- By clicking the Network Status button.
- By selecting the `Control / Network Status...` menu item.
- By using the hotkey *A* `?`.

4.5.3 Contents of the list

The listview contains a line for each pending or queued transfer. The line consists of the filename retrieved (without hostname and path), and the current status. Current status can be one of the following:

- **Queued**, image will be loaded when network slots become available.
- **Started**, transfer process has started.
- **Looking up**, looking up host name.
- **Connecting**, making connection.
- **Waiting**, request sent; waiting for response.
- **99999/99999**, reading.
- **Processing**, processing image (remapping it to the screen colors).

4.5.4 Cancel transfers

You can cancel one specific transfer by first selecting the entry in the listview, then click the button marked with **X** (left). You can cancel all transfers by clicking the button marked with **XXX** (right).

4.7.2 Differences

In compatibility mode, quoted attribute values are terminated by any occurrence of ">". In addition, quoted attributes that contain URLs, like HREF, SRC and ACTION, are terminated by whitespace. Comments are terminated by any occurrence of "-->".

In strict mode, AWeb complies to the HTML 2.0 standard.

4.8 Settings

In the settings requester, you can change many aspects of AWeb. The changed settings can be saved, only used for this session, or cancelled. The settings requester is opened by selecting the `Settings / Change Settings...` menu item.

4.8.1 Controlling the settings requester

Because there are far too many parameters to change to fit in one window, the settings requester is organized in pages. You can select a page from the *chooser* gadget in the upper part of the window.

The various requesters are explained in the next sections.

4.8.2 Browser 1: Font settings

On this settings page, you can change the font and style AWeb should use for different types of text. The first column in the list contains the HTML tag for the type of text. The second column contains the current font and style selected. Note that *Normal* is not a HTML tag, but merely denotes normal text that is not subject to text style tags. A brief description of the meaning of the HTML tag in the selected row is displayed below the list. The Ff button pops up a standard font requester, where you can change the font, the size and the style.

4.8.3 Browser 2: Appearance

On this settings page, you can change the way links are displayed.

Change the colour of links

To change the colour of a link, select the appropriate row in the list. There are three different colours:

Unfollowed link This colour is used for links that you haven't selected yet.

Followed link Links to pages you have visited before are shown in this colour.

Selected link This is the highlight colour a link will have when you click it.

Clicking the button marked with a monitor symbol will pop up a standard screen mode requester. Here you can select a screen mode, width, height and number of colours.

Load spread palette

If this checkbox is selected, AWeb will load a palette for the screen. This palette contains an even spread of colours, that will be used by all images. Using this feature avoids distortion of the palette that happens if one image takes up all colours, leaving no reasonable colours for next images.

AWeb will never load such a palette for screens with more than 256 colours. Those screens will be on a graphics card that is assumed to accomodate "enough" colours for all possible images.

Please be aware that 256 colours (or less) is not many. Although AWeb tries to calculate an evenly spread palette, still many images will look suboptimal. If you don't load the palette, the first few images look great, but later images may look even worse.

4.8.5 Screen 2: Windows

On this settings page, you can specify the location and size of the windows, and the window refresh mode.

Location and size

There are two button groups, one for the initial browser window, and one for the network status window.

Specify the window coordinates in the X and Y gadgets, and the width and height in the W and H gadgets. Alternatively, you can click the *snapshot* button that will fill in these gadgets with the current values.

The values for the browser window only apply to the first window that is opened. The second and later windows will be opened shifted over a small distance to the bottom and right, provided there is enough room on the screen.

Properties

Currently, this group only contains the *Simple refresh* checkbox. This checkbox determines the window refresh mode. The refresh mode determines what happens if a window is (partially) obscured by another window.

Smart refresh For a smart refresh window, the obscured part is put away temporarily by the operating system. If part of the window is revealed, the system automaticcaly restores the uncovered portion. This is relatively fast, because the program need not to redraw the window contents. Major disadvantage is, that this method needs possibly large amounts of chip memory, especially on screens with many colours.

Off AWeb will never load images automatically. If you want to see an image, you have to click on its icon.

Max. network connections

AWeb is capable of handling an unlimited number of parallel network connections. You might want to limit this number to a reasonable maximum, to avoid overloading your network line.

Note that if the maximum is reached, *image* loading will be queued, but not *document*. A document will always be loaded immediately, so the actual number of connections could be somewhat higher if you are retrieving a page.

The status of all connections can be viewed in the network status window.

Home page

This is the URL of your *home page* as far as AWeb is concerned. It doesn't need to be the same as your home page on the WWW. It is merely the page that will be shown if you select `Navigate / Home document` from the menu, or click the Home button.

You can type in the address, or click the button marked with the = symbol to copy the current URL from your first (or only) browser window.

Local index

If the name of a local file requested ends in a slash (/), this name is appended to the file name. This allows setting up your own WWW pages locally without having to change the names.

Browse anonymously

Normally, if you follow a hyperlink, AWeb sends the address of the page that link appeared on to the server. This allows servers to generate lists of back-links for interest, logging, optimized caching, tracing obsolete or mistyped links, etc. Some servers, like chat servers, actually need this information to let you get the page.

Because the source of a link may be private information, or may reveal an otherwise private information source, AWeb allows you to disable this feature. If the checkbox is selected, AWeb doesn't send this address, and you are browsing anonymously.

Ignore server MIME type

As explained in the MIME types and external viewers section (see section 4.8.12), most servers send the MIME type along with the data. Unfortunately, some servers send the wrong MIME type if they fail to recognize the true type

Mailto:

A *mailto:* address is for sending e-mail. If your mail reader supports command line arguments to send a mail, you can use this feature. An example is the *Voodoo* mail reader (by Osma Ahvenlampi), that can be used like this: Voodoo MAIL TO someone@foo.bar

If you don't use such a mailer, you can use a script that calls your editor, and then a mail post program. An example script is included in the AWeb archive.

Use the Command and Arguments fields to specify your mail command. Argument parameters are:
first %s = e-mail address to send the mail to.
second %s = screen name that AWeb is running on, in case your mail program supports opening on a public screen.

Ftp:

A *ftp:* address is for retrieving files via the FTP protocol. If you own a FTP client that supports command line arguments, you can use this feature. A simple example is the *ncftp* program that comes with the AmiTCP/IP package.

Use the Command and Arguments fields to specify your FTP command. Argument parameters are:
first %s = host name to connect to.
second %s = full path and file name to fetch.
third %s = screen name that AWeb is running on, in case your FTP program supports opening on a public screen.

4.8.10 Program 1: General

On this settings page, you can change some general settings.

Temp path

This is the directory where AWeb will create its temporary files. Temporary files are needed for several reasons: retrieved files for an external viewer, downloaded files, source files for inlined images, and swapped documents.

Type the full path name, or click the drawer button to pop up a standard drawer requester.

Save path

AWeb will always ask where to save a downloaded file, or the HTML source when using the `Project / Save As` menu function. The default save path will be the initial drawer used in the save file requester.

4.8.12 MIME types and external viewers

On this settings page, you can configure the MIME types AWeb should recognize, and the external viewers to use for each MIME type.

MIME types

MIME (Multipurpose Internet Mail Extensions) is a mechanism for specifying and describing the format of Internet message bodies. It was primarily designed for e-mail, but MIME types are used also to identify the type of data in the HTTP protocol, the most widely used protocol on the World Wide Web.

For a browser like AWeb, the MIME type of a document determines whether the file should be displayed in the browser window, or be processed by some other program.

A MIME type consists of a *type* and a *subtype*. The *type* describes the major class of data, like text or image. The *subtype* is used for a subdivision of the major type into different formats, like GIF or JPEG images.

According to RFC 1521, the following official MIME types are defined:

TEXT/HTML This is a document in the HTML hypertext format. Virtually all pages on the Web are in this format.

TEXT/PLAIN This type is used for plain text documents (normally in ASCII).

APPLICATION/OCTET-STREAM This describes a binary file. The file could be processed by some application. An example of this would be an LHA archive.

APPLICATION/POSTSCRIPT The document is in PostScript format.

IMAGE/GIF, IMAGE/JPEG These are images, in GIF and JPEG format.

AUDIO/BASIC This type is used for audio data encoded using 8-bit ISDN mu-law [PCM].

VIDEO/MPEG This is an animation in MPEG format.

In addition to these official types and subtypes, it is allowed to define extension MIME types and subtypes. These should start with **X-** to avoid collisions with future official MIME types.

Changing MIME types

Select the MIME type you want to modify from the listview. Use the **Add** button to add a new blank row. Use the **Del** button to remove the selected row. Note that the TEXT/HTML and TEXT/PLAIN types cannot be removed.

IMAGE/GIF gif This row specifies that GIF files can be recognized from their .gif extension. You specify no viewer because you want to use the default image viewer, defined in the IMAGE/* row.

IMAGE/JPEG jpeg jpg jfif SYS:Utilities/VT %s This row defines the possible extensions .jpeg, .jpg and .jfif for JPEG images. It also specifies that JPEG images should be displayed using the VT program.

IMAGE/X-IFF iff ilbm ham ham8 This row defines an extension MIME type for IFF images. Note that the subtype starts with **X-** because it is not an official MIME type. This line is important when looking at IFF files on your local computer, as AWeb has no way to identify them as IFF images other than the extensions given here.

IMAGE/* SYS:Utilities/MultiView %s screen This row defines what viewer (MultiView) to use for all other images but JPEG. Even files with different subtypes than GIF or JPEG (but main type IMAGE) will be shown using this viewer. There are no extensions defined here, because all extensions are given in the different subtype rows. As an alternative, you could remove the IMAGE/GIF and IMAGE/X-IFF rows, and specify all extensions (gif iff ilbm ham ham8) here.

4.8.14 Closing the requester

The bottom region of the requester contains three buttons:

Save Apply all changes and save the settings. The next time you start AWeb the same settings will be used.

Use Apply all changes for this session only. Unless you save the settings later using the `Settings / Save Settings` menu item, they will be forgotten the next time you start AWeb.

Cancel Don't apply changes.

4.9 ARexx interface

This version of AWeb has ARexx support, although it is somewhat limited.

4.9.1 ARexx port names

Every AWeb window has its own ARexx port. This port is named **AWeb.#**, where # is a unique number.

The About requester shows the actual name of the ARexx port for the window from which you selected the About menu item.

Due to internal limitations, it is not possible to have more than 14 ARexx ports open simultaneously. When you open more than 14 windows, the 15th and later windows will not have an ARexx port.

Up to arexx, or on to commands

11 : You submitted an unknown command.

20 : There was an internal error. The command is not executed.

4.10 Shell command and ARexx macro interface

4.10.1 Execute shell commands and ARexx macros

AWeb offers a unique and powerful facility to execute Amiga DOS Shell commands and ARexx macros from a page, just by clicking on a hyperlink or by submitting a form. With some effort, you can create complex applications using AWeb as the user interface, starting scripts that dynamically compose new documents that are loaded into AWeb via ARexx, etcetera.

Although this feature can be very useful, it could also be very dangerous. Therefore this feature works only from local pages (with a URL starting with file://localhost/), and only if the Allow Shell commands setting is selected.

4.10.2 Simple shell commands

To include a simple command, just add a normal hyperlink in your document that points to a URL of the form x-aweb:command/*your_DOS_command* If the user clicks on the hyperlink, *your_DOS_command* is executed. The output of the command is directed to an auto opening console window, unless you specify another output redirection in your command.

Because compatible HTML mode stops the URL at a space, make sure you have escaped all spaces in the command by "** **" or else the command won't work if the user has selected compatible HTML mode.

Example: get dir would allow the user to execute the *dir sys all* command by a click on the words "get dir".

Note: The DOS command is executed in a separate shell, with a current directory set equal to the current directory of AWeb. You are advised to use only absolute path names in the DOS command, or else the result will depend on which directory happened to be the current directory when you started AWeb.

4.10.3 ARexx macros

Starting ARexx macros from your page works in a similar way. Just add a normal hyperlink that points to a URL of the form x-aweb:rexx/*your_ARexx_macro* If the user clicks on the hyperlink, *your_ARexx_macro* is started with the ARexx port for this window as the default command port.

4.11 Extension URLs

Internally, AWeb uses an extension to the URL scheme for some tasks. You can take advantage of this, by using the same URLs. These extension URLs always start with "**x-aweb:**". Note that you should only include such extension URLs in pages that will only be viewed by AWeb, because other browsers will not recognize these URLs.

A good place for extension URLs would be your hotlist. You can, for instance, access the AMosaic or IBrowse hotlist from within your hotlist, or even configure one of these as your home page within AWeb.

4.11.1 Hotlists

AWeb uses extension URLs to identify its hotlist, and other browser's hotlists.

x-aweb:hotlist Identifies AWeb's own hotlist.

x-aweb:amhotlist.rexx Identifies the ARexx based hotlist of AMosaic version 1.2. It uses the file ENV:mosaic/hotlist.html.

x-aweb:amhotlist.20 Identifies the hierarchical horlist of AMosaic 2.0 prerelease. It uses the file ENV:mosaic/.mosaic-hotlist-default.

x-aweb:ibhotlist/*path* Identifies the hotlist of IBrowse. Because this hotlist doesn't have a fixed location, you must specify the full path and file name in the URL.

4.11.2 Window history

The window history can be accessed through an extension URL:

x-aweb:windowhis Identifies the window history.

4.11.3 Shell commands and ARexx macros

Two extension URLs exist to start shell commands or ARexx macros.

x-aweb:command/*shell_command* Forms the interface to start Shell commands.

x-aweb:rexx/*ARexx_macro* Forms the interface to start ARexx macros.

notifying the author of the page that his page contains erroneous HTML, you can try to select Control / HTML mode / Compatible from the menu.

- *The colour requester from the Screen 3: palette settings page messes things up on a CyberGraphics screen.* This is a bug in CyberGraphics. It doesn't react properly to changes in the palette.
- *Closing the AWeb public screen while there is a visitor window open crashes if the screen is a CyberGraphics screen.* This is a bug in CyberGraphics V 2.15 and lower. The same thing happens on other CyberGraphics screens.
- *If the window is not active, a click in the scroller container (outside the knob) moves the scroller but doesn't scroll the window contents.* This is a bug in Intuition. The same thing happens sometimes with MultiView.
- *I use MagicMenu, and every now and then my system hangs if I open a menu.* This is a known problem in MagicMenu, caused by the fact that MagicMenu is not 100% compatible with the way Intuition handles menus.
- *I don't use MagicMenu, but my system hangs if I click on the wide part of the chooser gadget in the settings window.* Be sure to use the version of chooser.gadget included in this archive. Also, some animated mouse pointer commodities are known to cause this hang.

5.3 To do

The to-do list contains additional suggestions, received after the release. Please check both the todo list in the on-disk docs and the online page before sending me any suggestions.

Chapter 7

Acknowledgements

I especially wish to thank:

- **Osma Ahvenlampi** (”Tau”) for the TCP stack-adaptive design.
- **Jeroen Oudejans** for creating and maintaining the AWeb FAQ and for creating the PostScript printable version of the docs.
- **Thomas Tavoly** (”aTmosh”) for enhancing the logo and creating the Workbench icons.

I also wish to thank my beta testers, without them AWeb would never have become the great program it is.

- Osma Ahvenlampi (http://www.iki.fi/oa/)
- Jeroen Oudejans (jhwo@xs4all.nl)
- Thomas Tavoly (http://www.cistron.nl/~ttavoly/)
- Vincent Groenewold (http://www.zeelandnet.nl/people/supernov/)
- Donovan Janus
- Mike Meyer
- Paul Kolenbrander (http://www.iaehv.nl/users/paul/amiga.html)
- Donald Voogd
- Kristian Phillips
- Dale Currie
- Christopher Aldi