AWeb

Version 1.1, April 1996

The World Wide Web Browser for the Amiga computer

Contents

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).

1.5 Distribution

Distribution of AWeb is permitted on a limited basis as described by this licence. Permission to distribute unmodified copies of the AWeb archive on disk is granted provided no money is charged beyond the cost of the storage media. Commercial distributors of freely distributable software are explicitly **NOT** allowed to distribute AWeb. Permission to distribute the unmodified AWeb archive electronically on computer networks and bulletin board systems is granted provided that no money is charged explicitly for accessing or downloading the archive.

Distribution of the unmodified AWeb archive on magazine cover disks and similar media requires that a copy of the issue is sent to the author. Contact me before placing AWeb on a cover disk. Refer to chapter 6 for information on contacting me.

Distribution of AWeb that violates this license is prohibited without the prior written permission of the author.

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.

- Bank transfer.
  - From within The Netherlands:
    Transfer to Postbank account 4948774 of Y. Rozijn, Zuideinde 9, Meppel.
  - From outside The Netherlands:
    Transfer to account 4948774 of Postbank, Amsterdam, The Netherlands, SWIFT: INGBNL2A of Y. Rozijn, Zuideinde 9, Meppel, The Netherlands.
    Please make sure to mention the correct address and SWIFT number, or else it will cost me too much.
- Eurocheque. Please make payable to: Y. Rozijn, Zuideinde 9, Meppel, The Netherlands.
- International Postal Money Order. Please send to:

  Y. Rozijn
  Zuideinde 9
  NL-7941 GA Meppel
  The Netherlands

  Note that this method is relatively expensive for you, and may take a very long time (over 6 weeks).

Other payment methods are **not** possible. Especially bank checks are not acceptable for me due to the high costs involved.

If none of these methods are acceptable for you, please e-mail me.

2.3 Registration form

If you want to register, please fill out this registration form. Please refer to the on-disk documentation for information on how to print a registration form.

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 improve network speed, host names and network addresses are cached so addresses are looked up only once during a session.
- AWeb can open it's windows on the default public screen, on a named public screen or open its own screen.
- In extension to its own hotlist, AWeb can read other hotlists, like those of AMosaic.
- An advanced settings requester is integrated in the program. This requester includes:
  - Fonts and styles AWeb uses for different types of text.
  - Link colors and underlining.
  - Screensettings.
  - Screencolors for AWeb's own screen.
  - Location and size of the AWeb windows.
  - General network settings (image loading and home page).
  - Proxy settings.
  - External ftp and mail commands.
  - Save and temporary paths, cache size.
  - External editor and HTML source viewer.
  - MIME types AWeb should recognize and the external viewers to use.
- AWeb has an ARexx interface, and a unique and powerful shell command interface.

3.1 Why not use MUI ?

I must admit that MUI is a very clever piece of programming. And basically, I like the idea of a fully user-configureable GUI very much. But in my opinion, visual feedback and intuitivity are more important features of a GUI than configurability. And it is in these areas that MUI scores very badly.

3.1.1 Visual feedback

Intuition and BOOPSI gadgets give immediate visual feedback when the user plays with them. This is due to the fact that the gadget imagery update is done by the input task, not by the application. The only circumstance in which there is no immediate visual feedback, is when some program has switched off multitasking, but that's not a normal condition. With MUI, all feedback is done in the application's context. This effectively means cooperative multitasking instead of preemptive multitasking. And this cooperative "multitasking" is exactly why most Amiga owners dislike MS-Windows. For the very same reason many Amiga owners (me included) don't like MUI.

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:

How do I get AMosaic to run on its own screen? I believe that the 18 (eighteen!) steps to perform to accomplish this, are far too many.

And even if the way configuring MUI aspects of an application is improved, there are still many MUI applications offering *two* settings requesters - one for the MUI aspects, and one for the application's own parameters. This is also confusing. For the user, a system like MUI should be transparent. The user should not need to know which aspect is controlled by MUI, and which by the application itself.

3.1.4 Minor objections

Size and speed

Many people complain that MUI is big and slow. Many other people say you can't expect a 1 MB, 8 MHz 68000 machine to run modern applications, and the things MUI offers is well worth the extra resource cost.

Fact is, MUI takes up memory and CPU time, which makes it bigger and slower than no MUI at all. If you don't give much about fancy looking gadgets, MUI rapidly becomes *too* big and *too* slow.

Shareware

MUI is shareware. And MUI applications *need* MUI, it is not an option.

I want the user to have a choice which extras he or she wants to buy.

3.1.5 Flames?

Don't flame me about this. I won't read it. If you are a MUI lover, well, there are at least three MUI based browsers available. Use one of them instead of AWeb.

3.2 ClassAct

AWeb uses the ClassAct GUI toolkit. ClassAct is a set of BOOPSI gadget and image classes, designed to speed up and simplify the development of GUI applications. Free to users of the applications, ClassAct is available as shareware and commercial licences for Amiga developers. The licence includes the development kit, support, and distribution rights to the shared ClassAct class libraries. For more information, refer to http://www.warped.com/~timmer/classact.html or http://www.nai.net/~caldi or send mail to caldi@nai.net.

3.3 System requirements

AWeb needs the following to run:

- 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*.

3.4.2 Installer problems

In the past, Commodore has released several versions of the *Installer* program with quite different capabilities. Unfortunately, some misbehaving installation procedures of other applications might have copied an earlier version of the Installer program onto your hard disk. This can result in error messages like: Unable to compile script when you try to install AWeb.

If you get such error message, you should delete SYS:Utilities/Installer if it exists in that drawer, and make sure the file SYS:C/Installer is the same as the one on your Workbench 3.x Install diskette.

3.4.3 Configuring the JFIF datatype

If you use the JFIF datatype (by Christoph Feck, TowerSystems), then please read this:

The JFIF datatype doesn't seem to handle shareable pens on public screens correctly under all circumstances. You have to install AWeb in the datatype or else AWeb will not be able to show inline JPEG images.

In the JFIF preference editor, you must add an application named **AWebIP** (AWeb Image Processing), and then select *Single-Pass Quantization* (in the GadTools version of the preference editor) or *One-pass* (in the MUI version).

3.5 Tips for 2MB Amiga users

You can use AWeb on a 2MB Amiga, if you configure it properly. Here are some suggestions:

- Use AWeb on the default public screen.
- Select simple refresh for your windows.
- Set image loading *Off*, and only load images you really want to see by clicking the icons.
- Set maximum number of network connections to 1. Every connection takes up precious memory.
- Set your temporary path to a directory on your hard disk, not in RAM.
- Set your cache as follows:
  - document memory: about 40% of the amount of free memory when AWeb and your TCP stack are running
  - document disk: as large as you like and have room on your hard disk
  - image memory: the same as *document memory*
  - image disk: as large as you like and have room on your hard disk

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.

The file in the HOTLIST argument will be used 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.

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.

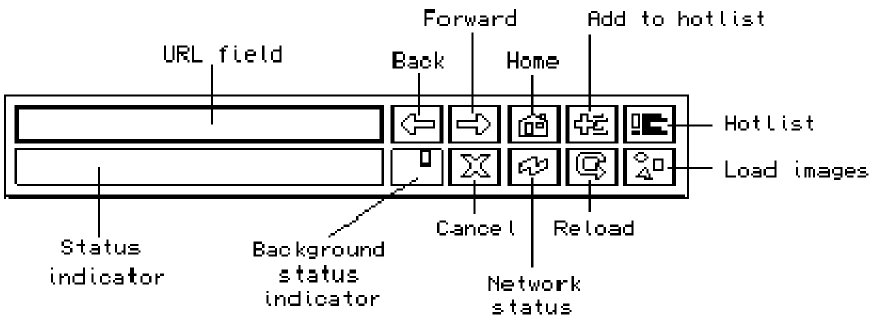

Figure 4.1: An overview of the graphical user interface.

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, **AU**[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, **AW**, 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.

4.1.2 Status indicator

This field serves two purposes.

First, when browsing through a page, it shows the URL "behind" the link currently pointed to with the mouse. That is, when you click this URL will be retrieved.

Second, when a page is being loaded for this window, is shows the current state. If actual data is retrieved, a progress bar will appear showing how far the load process is. The progress bar will not appear if the final size of the document is not known on forehand.

4.1.3 Background status indicator

When one or more load operations are in progress, this indicator will show a little square. On every connection made, and on every block retrieved, the square will advance one step.

This indicator lets you know if there are still things loading in the background, and how rapidly they progress. If you want more detail, the network status window will tell you everything.

4.1.4 Back button

This button lets you walk back through the window history. The window history contains all pages viewed before *in this window*.

The `Navigate / Back` menu function, or its shortcut, **A**`B`, will do the same.

4.1.5 Forward button

This button lets you walk forward through the window history. The window history contains all pages viewed before *in this window*. Note that the window history is currently linear, without branches. That means that if you walk back a few steps, then retrieve another page, the window history "after" that page will be forgotten.

The `Navigate / Forward` menu function, or its shortcut, **A**`F`, will do the same.

4.1.6 Home button

This button retrieves the URL that is configured as your home page.

The `Navigate / Home document` menu function, or its shortcut, **A**`D`, will do the same.

4.1.7 Add to hotlist button

This button adds the current document to your hotlist.

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

4.1.8 Hotlist button

This button shows your hotlist.

The `Hotlist / Show hotlist` menu function, or its shortcut, **A**`H`, 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, **A**`X`, 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, **A**`I`, 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.

New window Open a new window. Only available in the registered version.

Close window Close the current window. Only available in the registered version.

Open URL Clear the URL field and activate it so you can type a new URL.

Open WWW Preset the URL field with "http://www." and activate it for maximum convenience if you want to load a WWW page.

Open local... Opens a standard file requester. After you select a HTML file, the file will be loaded in the current window.

View source... Show the HTML source of the current page, using the viewer program that was installed as HTML source viewer.

Save source... Opens a standard file requester. After you type a file name or select a file, the HTML source of the current page is saved. If the selected file already exists, you have the choice to:

- overwrite the old file,
- append the source to the old file,
- select another name, or
- cancel the save alltogether.

About... Opens a window with version information. If the current window has an ARexx port, the name is shown here.

Quit Quit AWeb after confirmation. If you confirm, all pending network operations are cancelled.

4.2.2 Control menu

The *Control menu* offers functions to control the operation of AWeb, other than cache functions.

Load images now Initiates the load of images in the current document. This menu item has 2 sub-items:

  All images Initiates the load of all images in the current document that aren't loaded yet. Pressing the Load Images button will do the same.

  Maps only Initiates the load of all clickable maps in the current document that aren't loaded already.

Network status... Open the network status window, or bring it up to front if it is already open. Pressing the Network Status button will do the same.

Reload current Reload the current document. The page is deleted from the cache, and retrieved again. Pressing the Reload button will do the same.

Cancel load Interrupt (cancel) the load of a page in this window. Background loads cannot be cancelled by this menu function, use the cancel button in the network status window instead. Pressing the Cancel button, or pressing the **Esc** key will do the same.

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.

Forward Walk forward one document through the window history. The window history contains all pages viewed before *in this window*. Note that the window history is currently linear, without branches. That means that if you walk back a few steps, then retrieve another page, the window history "after" that page will be forgotten. Pressing the Forward button, or using the **Alt + cursor right** key combination, will do the same.

Home document Retrieve the URL that is configured as your home page. Pressing the Home button will do the same.

Window history Show the window history. The current document is marked.

4.2.5 Hotlist menu

The *Hotlist menu* offers functions to use the hotlist, or to read foreign hotlists.

Add to hotlist Add the current document to your hotlist. Currently, the hotlist is linear, not hierarchical.

Show hotlist Show the hotlist in the browser window.

AMosaic (ARexx) Show the ARexx-based hotlist of AMosaic 1.2. AWeb expects it to reside under the name ENV:mosaic/hotlist.html.

AMosaic (2.0) Show the hierarchical hotlist of AMosaic 2.0 (pre-release). AWeb expects it to reside under the name ENV:mosaic/.mosaic-hotlist-default.

Other... Opens a standard file requester, from which you can select other hierarchical hotlists. These include hotlists saved by older versions of IBrowse, and other hotlists created by AWeb.

4.2.6 Settings menu

The *Settings menu* offers functions to configure AWeb.

Image loading This menu item provides a quick way to set the image loading. It has 3 sub-items:

- All images
- Maps only
- Off

with the same meaning as the choices in the image loading chooser.

Change settings... Brings up the settings requester.

Save settings Save the current settings. This function will take a snapshot of your window positions first.

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.

4.3.2 Images

Inlined images are treated completely differently. Images are stored in the temporary directory when they are loaded. If the image is received completely, the datatype will *process* the file, resulting in a displayable image in *chip* memory (the image cache). When the image cache is full, undisplayed images in memory are removed, but the disk file remains. If the image must be displayed later, the disk file is processed again, but no network access is needed.

Because the datatype locks the file, the disk file must remain while the image is in memory.

When the total size of image files is larger than the *image cache disk size*, the least recently displayed images are deleted.

4.3.3 Low memory

To avoid crashes when running out of memory (caused by the datatypes or other external programs), AWeb tries to leave at least 100 kB of chip memory free. Whenever there is less than 100 kB total memory free, AWeb will swap out documents, even if the document cache isn't full yet. Displayed documents will never be swapped out.

When there is more than 100 kB total free memory, but less than 100 kB free chip memory, AWeb will flush images from memory, even if they are being displayed. As long as the image disk cache isn't full, they remain on disk.

4.3.4 Setting the cache sizes

You can set the cache sizes from the Program 1: General page in the settings requester.

If you are using a 2MB Amiga, have a look at the 2MB tips (section 3.5). Otherwise, you will probably want to use a

- temporary directory in RAM;
- a *document memory cache* large enough,
- a *document disk cache* of size 0;
- a reasonably sized *image memory cache* (this will take up chip memory),
- a large *image disk cache*.

Of course, you can use other settings if you like. You can fine-tune the memory and disk usage by changing these settings.

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.

4.4.3 Inlined images

A document can contain inlined images interspersed with the text. If an inlined image is not (yet) loaded, AWeb displays an icon for that image. You can select if you want images to be loaded immediately or not using the image loading chooser in the settings requester.

AWeb displays different icons under different circumstances. Figure 4.2 shows these different icons and their meaning.

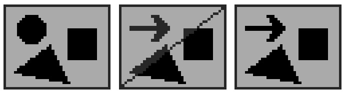

Figure 4.2: The first icon depicts an unloaded image. Click it to load the image. The second icon depicts an unloaded image that is also a link to another document. Click in the upper left half of the icon to follow the link directly, or in the bottom right half to load the image. The third icon depicts an unloaded clickable map. Click the icon to load the image. Once it is loaded you can pick a spot from the map.

For an inlined image, a so-called ALT-text can be defined. This is a text that can be displayed if the browser doesn't display the image. Of course, AWeb understands this ALT-text and will display it instead of the icon imagery. With ALT-text, unloaded images look like as depicted in figure 4.3

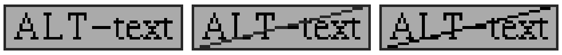

Figure 4.3: The first icon depicts an unloaded image. The second icon depicts an unloaded image that is also a link. The third icon depicts an unloaded clickable map. The difference between the second and third icon is minimal. Please refer to the on-disk documentation for a more clear view of these icons.

4.4.4 Downloading

Instead of following a link and display the new document, or loading and displaying an inlined image, you can *download* a document or an inlined image. To do so, hold the **Shift** key while clicking the link or image icon. The document or image is retrieved, and a standard save requester will pop up to let you specify a file name.

If the document or image is already in cache, it will only be saved, not retrieved again over the network.

Note you can also save a displayed image in this way. Just press the **Shift** key and click the image. If the image is also a link, this could be abmiguous; therefore AWeb will save the image in this case. If you want to download the document "behind" the link, you can either select `Cache / Flush images` in 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).

4.6 User authorization

4.6.1 Authorization

User Authorization is a way to protect certain documents on the net by a user ID and password. To access such documents, you generally have to register yourself first.

Whenever you attempt to retrieve a document for which user authorization is necessary, AWeb will pop up a requester where you can type in your user ID and password. If these are valid, you will get the document.

On subsequent accesses to documents within the same *realm* on the same server, you don't need to enter your user ID and password again.

4.6.2 Save your authorization details

AWeb remembers all authorization details within a session. To save these data, select the `Cache / Save authorizations` menu item. The next time you start AWeb, your authorization details are read so you won't need to enter these again.

The menu item `Cache / Flush authorizations` lets AWeb forget all your authorization details. Select `Cache / Save authorizations` afterwards to wipe out the disk file too.

Note: the authorization file is in an internal format. Do not attempt to modify this file.

4.7 Compatibility mode

4.7.1 Incorrect HTML

By default, AWeb complies fully to the HTML-2 standard. Unfortunately, there are many pages on the Web designed using an inferior browser on another hardware platform. That browser has several bugs, so that even an ill-formed HTML-page looks good when using that particular browser. When you view such a page using a decent Web browser, like AWeb, the page might look distorted. You can expect large parts of the page missing, links to URLs that seem to contain HTML tags, and other strange things.

If you encounter such problems, try using the compatibility mode in AWeb. In this mode, AWeb tries to mimic the bugs in this widely-used PC-browser, showing as much of the erroneous page as it possibly could.

To show a page in compatibility mode use the `Control / HTML mode / Compatible` menu item.

To view the page using strict HTML-2 again, use the `Control / HTML mode / Strict` menu item.

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.

Then press the **Change colour** button, this will pop up a colour requester. Note that this colour requester (as opposed to the Palette colour requester) does *not* change the screen's palette. Instead, it picks the colour from the available palette that fits best to the selected colour values. Internally, the 24 bit colour values are stored, and for every screen AWeb opens on the best fitting colours are determined.

Change the underlining

The checkbox determines whether links should be displayed underlined or not. If underlined is selected, *unfollowed links* are underlined with a solid line, and *followed* links will have a dashed line.

If this checkbox is selected, images that are links have a border in the appropriate colour.

Change the cycle field

Many people use a commodity that turns cycle gadgets into popup menus. Because a cycle gadget certainly has its drawbacks, AWeb offers the possibility to display a cycle field in a form as a list. Note that selection fields with more than 5 selections, or with multiple selections, are always turned into a list.

If this checkbox is selected, all selection fields are displayed as lists, even those with less than 5 selections.

4.8.4 Screen 1: Screen

On this settings page, you can specify on which screen AWeb should open its windows. Using the chooser, you can make AWeb to open on the default public screen, a named public screen, or let AWeb open its own public screen.

Default public screen

If this is selected, AWeb will open its windows on the default public screen. Usually this is the Workbench screen.

Named public screen

AWeb will open its windows on a public screen that is not necessarily the default. You can enter the screen name you want AWeb to open on. If the screen doesn't exist when AWeb starts, the default public screen is used instead.

Own public screen

Using this selection, AWeb will open its own public screen. The name of this screen is **AWeb**.

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.

Simple refresh No parts of the window are saved for simple refresh windows. If part of the window is revealed, the program must redraw the uncovered parts. This will take up no extra chip memory at all. The drawback is, that scrolling and refreshing the window will take more time.

4.8.6 Screen 3: Palette

On this settings page, you can change the colour settings for the AWeb own public screen, pretty much like the Workbench Palette preferences program does for the Workbench screen.

The gadgets on this settings page are only enabled if AWeb runs on its own screen. When using another screen, you should use the colour settings method provided by the owner of that screen.

The palette settings are only effective when AWeb opens its own screen. When using another screen, it respects the settings for that screen.

This settings page has two major parts.

Pen settings

The list, and the upper palette button row, specify the screen pen settings, comparable to the right-hand part of the Workbench Palette program. To change a pen, first select the pen from the list, then select the color from the palette.

Change palette

The lower palette button row allows you to change the actual color of the pens. This is comparable to the left-hand part of the Workbench Palette program. Press the **Change colour** button to pop up a colour requester for the selected colour from the palette.

4.8.7 Network 1: General

On this settings page, you can change some general settings regarding the network access.

Image loading

This chooser lets you select if you want AWeb to load images:

All images AWeb will start loading every image as soon as it is encountered in a page. This could consume a lot of bandwidth, especially if you are using a slow connection.

Maps only This option doesn't load all images, but only clickable maps. This can save a lot of traffic, and still lets you use clickable maps, as these are often essential navigation tools.

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 of the data. This may lead to a file being saved instead of passed to an external viewer as expected.

In those cases, it is best to ignore the MIME type as reported by the server, and let AWeb identify the MIME type by the file name extension.

If this checkbox is selected, the MIME type as reported by the server is ignored.

Allow Shell commands in links

AWeb offers a powerful facility to excute Shell commands just by clicking a hyperlink or submitting a form.

Although this feature can be very useful, it could also cause severe damage if an undesired command like FORMAT would be executed. Therefore this feature is disabled by default. Select this checkbox to enable it.

4.8.8 Network 2: Proxy

On this settings page, you can configure proxy servers.

Proxy servers

A proxy server is a special server, that acts as a gateway between your computer and the Internet. Instead of having to establish a connection to a server possibly on the other end of the world, a browser only connects to the proxy. The proxy server has a cache of the most popular pages, so there is a chance the page you requested is already there. If not, the proxy server retrieves the document for you. This will decrease the traffic on the network, thus speeding up websurfing.

Sometimes, the proxy server is the *only* way to connect to your provider, thereby acting as a firewall.

Configuring a proxy

Just type in the address of the proxy in the appropriate field. Make sure it is in one of these two forms: **http://proxy.foo.bar**, or **http://proxy.foo.bar:8080**, where the name and port number may differ, of course.

If the address doesn't start with "http://", AWeb will prepend this to the address.

4.8.9 Network 3: External programs

On this settings page, you can configure some external programs for network accesses that AWeb doesn't support directly yet.

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.

Scroll overlap

If you scroll up or down by a page, there is some overlap. Bacause you might want to have a larger overlapping area when you are using a larger font, you can change the overlap size.

Set this gadget to the desired overlap size, measured in pixels.

Caches

Use these four gadgets to fine-tune the amount of cache AWeb should use. All sizes should be given in kB. See also section 4.3.

4.8.11 Program 2: External programs

On this settings page, you can configure some external programs to be used by AWeb.

Editor

This is the editor command invoked when you click the edit gadget in text area form fields.

Use the Command and Arguments fields to specify your editor command. Argument parameters are:
first %s = file name to edit.
second %s = screen name that AWeb is running on, in case your editor supports opening on a public screen.

Make sure the command will **not** return until you leave the editor. For some editors, this will need a STICKY or KEEPIO argument.

HTML source viewer

Currently, AWeb relies on an external viewer for the Project / View source menu function.

Use the Command and Arguments fields to specify your source viewer command. Argument parameters are:
first %s = file name to edit.
second %s = screen name that AWeb is running on, in case your viewer supports opening on a public screen.

Note that the default setting, *MultiView*, will not produce the expected results if you happen to have a HTML datatype installed on your system. In that case, the datatype will show the source as HTML again. If this happens, you should configure another viewer.

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.

MIME type and subtype

In these string gadgets, you specify the MIME type and subtype.

You can use an asterisk to specify a wildcard subtype. AWeb will use the external viewer defined in this row for files with the same type but a subtype for which no external viewer is defined. See the example.

Extensions

Most servers send the MIME type together with the data. AWeb will then use this MIME type, unless Ignore server MIME type is selected. If the server doesn't specify the MIME type (or if it is ignored), AWeb tries to determine the MIME type from the file name extension. If that fails, AWeb looks at the data to see if it is HTML text or plain text.

The extensions are especially important when looking at local files. As there is no server for local files, there is only the extension that tells AWeb about the type of the file.

In this string gadget, you type the extensions that could identify this MIME type. Separate multiple extensions by spaces or commas. The extensions are not case sensitive.

Processing

Files of types TEXT/HTML and TEXT/PLAIN will be shown in the browser window. Files of other types are processed by an external viewer. In spite of the name *viewer*, this is not limited to graphical files. The external "viewer" for an audio file, for example, will play the audio file.

Use the Command and Arguments fields to specify the viewer command to execute for this MIME type. Argument parameters are:
first %s = file name to "view"
second %s = screen name that AWeb is running on, in case your external viewer supports opening on a public screen. Use this only if you want it to open on the same screen as AWeb.

If AWeb can't determine the MIME type, or if the MIME type is known but not in the list, or if the MIME type is in the list but there is no external viewer defined, AWeb will pop up a save requester. You can than save the file, and try to process it later.

4.8.13 Example

Suppose you want to see JPEG images using the VT program, and other images using the MultiView program on its own screen. You know that JPEG files can have extensions **jpeg**, **jpg**, or **jfif**, and that GIF files have an extension **gif**. IFF images can be recognized by **iff**, **ilbm**, **ham** or **ham8**. You want AWeb to recognize other image formats you don't know of.

Then you would configure the following MIME types:

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

4.9.2 ARexx commands

Currently a very rudimentary command set is implemented. More commands will be added in the future.

Available commands are:

OPEN URL/A,RELOAD/S Retrieve and show the document for this URL. The RELOAD switch will reload the document even it is still in the document cache.

RELOAD Reload the current document.

GET ITEM/A Get information from the document in this window. The *ITEM* argument determines the information to return:

URL Retrieve the URL of the document.

SOURCE Retrieve the HTML source of the document.

TITLE Retrieve the title of the document. If no title was defined in the document, the document's URL is returned.

The information is returned in the reserved variable RESULT.

ACTIVATEWINDOW Make this window the active window.

WINDOWTOFRONT Move this window in front of all other windows on the screen.

WINDOWTOBACK Move this window to the back of all other windows on the screen.

CLOSE FORCE/S Close this window. The FORCE switch suppresses the "Are you sure" requester if this was the last window.

QUIT FORCE/S Quit AWeb. The FORCE switch suppresses the "Are you sure" requester.

Back to port names, up to arexx, or on to return codes

4.9.3 Return values from commands

Every ARexx command returns a completion code in the reserved ARexx variable RC.

ARexx commands return the following codes:

0 : Command executed succesfully.

1 : Command executed succesfully, but there is some condition that might be of interest.

5 : The command was syntactically valid, but could not be completed for some reason.

10 : You supplied invalid arguments for this command.

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.

4.10.4 Parameters

You can use a HTML *form* or a *clickable map* to pass parameters to your DOS command or ARexx macro.

Forms

Supply a ACTION="x-aweb:command/*your_command*" attribute in your <FORM> tag to execute the command if the user submits the form. Similarly, you can include a ACTION="x-aweb:rexx/*your_macro*" attribute to start the ARexx macro.

Form parameters are converted to Amiga DOS style parameters: the field name will be used as the argument name, and the field value will be used as argment value. The value will be quoted, with the *escape*, *newline* and *quote* characters in the value escaped as required by Amiga DOS.

Note: *switch arguments* (/S) cannot be passed in this way. You could use a script instead, like the example in the on-disk documentation.

Clickable maps

When using a clickable map, the x and y coordinates of the mouse pointer within the image are passed to the command as parameters without keyword.

ARexx arguments

Parameters for ARexx macros are passed in the same format as for DOS scripts. The argument string will contain the name, an equal sign, and a quoted value for each form parameter. Have a look at the second example in the on-disk documentation for one possible way of parsing this.

4.10.5 Load the result back into AWeb

If your script or macro has created a HTML document (or just a plain text file), you can automatically load this file back into AWeb. Use the ARexxOPEN command for this purpose. If you re-use the name of your file for different responses, be sure to add the RELOAD switch to prevent AWeb from showing the previous (cached) document again.

Of course, this will work better from within an ARexx macro than from within a DOS script. In a DOS script, you have no way of determining to which ARexx port you should address the OPEN command.

4.10.6 Examples

Please have a look at the on-disk documentation for two examples.

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.

Chapter 5

Problems, comments and suggestions

5.1 Known AWeb bugs

Although AWeb is thoroughly tested, it is very likely that there are some bugs left. At the time of release no bugs were known. For the latest information, check the online bugs page: http://xs4all.nl/~yrozijn/aweb/bugs.html.

- A line break can occur at a tag that doesn't imply a word or line break.

5.2 Common problems

Listed below are some common problems you might encounter, and their solution.

- *I can't install AWeb. The installer reports errors like: Unable to compile script.* Most likely some misbehaving application installation has copied an obsolete version of the Commodore Installer utility to your hard drive. Make sure the Installer version in the SYS:C directory is "installer_2 2.17 (1993-02-13)" or better. This version can be found on the Amiga Workbench 3.x Install floppy disk. Also, delete any copies of the Installer program from your SYS:Utilities drawer.

- *AWeb crashes when loading a GIF image* The commonly used ZGif datatype version 39.16 had a bug. Be sure to use version 39.18 or better of the ZGif datatype, or another GIF datatype.

- *AWeb doesn't display all of the page. Other browsers display the page correctly* Probably the page contains some erroneous HTML. One of the common errors is the use of something like <!————————-> as a dividor line in the HTML source. This can lead to large parts of the page commented out if the number of dashes is not exactly right. Apart from 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 6

How to contact the author

You can contact me

- by e-mail: yrozijn@xs4all.nl
- by snail-mail:
  Yvon Rozijn
  Zuideinde 9
  NL-7941 GA Meppel
  The Netherlands

Please don't e-mail me unless you have a serious question or suggestion. I expect a lot of e-mail, so please don't be mad if I don't reply your mail. I will try to reply at least to all serious questions. When I receive a lot of questions on the same topic, I could add the question and answer to the FAQ, instead of replying to everyone individually. So, if you don't get a reply within a few days, have a look at the AWeb FAQ (http://huizen.dds.nl/~aweb).

If you have questions about AWeb, you can also mail to the FAQ: aweb@dds.nl.

Before sending me any questions, have a look at the AWeb Frequently Asked Questions (http://huizen.dds.nl/~aweb), and the AWeb home page (http://www.xs4all.nl/~yrozijn/aweb/).

Before sending me any bugs, have a look at both the local and online *known bugs list*.

Also, before sending me any suggestions, have a look at both the local and online *to-do list*.

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