Contents

- AWeb can print the visited pages on a graphical printer.
- An easy popup menu allows you to perform many different functions on any link (like load in background, open in new window) or image (like save to disk, flush, view with external viewer).
- AWeb has a fast, persistent, disk-based cache. The cachebrowser allows easy off-line browsing and cache management.
- 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.
- AWeb has a hierarchical hotlist, with options to group entries. In extension to its own hotlist, AWeb can read other hotlists, like those of AMosaic.
- A set of advanced settings requesters is integrated in the program. You can control many aspects of the program in these requesters.
- AWeb has an ARexx interface, and a unique and powerful shell command interface.

1.2 Legal issues

1.2.1 Copyright

The AWeb browser, and all files included in the distribution are, unless otherwise noted,**Copyright ©1996 by Yvon Rozijn. All rights reserved.**

The ClassAct gadget system is Copyright ©1995 Phantom Development.

1.2.2 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 and AmiTrix Development assume 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.

1.2.3 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.

Chapter 2

Before you begin

2.1 System requirements

AWeb needs the following to run:

- OS 3.0 or better
- At least 2 MB of memory as an absolute minimum. 3 MB or more is highly recommended.
- Either AmiTCP/IP, I-Net225 or AS-225 to access the World Wide Web, or a TCP stack that is compatible with one of these. Examples of compatible products are Miami and TermiteTCP.
- 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. To hear background sounds on pages, a WAV and an AU datatype are recommended.
- The ClassAct gadget kit (included in the distribution). The latest ClassAct archive is always available from:
  ftp://ftp.warped.com/pub/amiga/classact/.

2.2 Installation

2.2.1 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.

If you follow these steps, websurfing is possible with only 2MB. Be prepared however that complicated pages (with large tables) may not fit completely in memory. They will be truncated.

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

3.2 The GUI

On top of the browser window there are a number of gadgets. The exact size and imagery of the buttons may vary depending on your set-up, but their location is fixed (see figure 3.1). This section explains these gadgets.

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**[1]`U`, 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.

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, it 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.

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 (see section 4.5) will tell you everything.

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`, or the **Alt + cursor left** key combination, will do the same.

[1]**A** is used throughout the document as a way to denote the right Amiga key

Search button

This button opens the search requester (see section 4.1), or bring it to front if it is already open.

The Project / Search... menu function, or its shortcut, **AF**, will do the same.

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.

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.

3.3 Menus

3.3.1 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.

Close window Close the current window.

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.

Search engines Opens the local page extras/search.html which contains a quick interface to several search engines on the Internet.

Search... Opens the search requester (see section 4.1). Pressing the *search* button will do the same.

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:

3.3.3 Cache menu

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

Cache browser... Opens the cache browser.

Flush from memory Flush one of the following from the memory cache. If the objects were also on disk, they will remain there.

- **Nondisplayed images** All images that are not used in the currently visible page(s) are flushed from memory.
- **All images** All images, including those on the current page, are flushed from memory.
- **Nondisplayed documents** All documents that are not currently visible are flushed from memory.

Delete from disk Delete one of the following from the disk cache and also from memory if the objects are not in use. You will be asked for confirmation first.

- **All images** Delete all images from the disk cache.
- **All documents** Delete all documents from the disk cache.
- **Erase cache** Delete all images and documents from the disk cache.

Fix cache... Synchronizes the cache registration with the files actually found on disk. Also, any files in the cache directory that don't belong to AWebs cache are deleted.

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.

3.3.4 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*. 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 requester.

Network settings... Brings up the network settings requester.

ClassAct settings... Brings up the ClassAct preferences requester.

Save current settings Saves all current settings.

Snapshot windows Saves the current positions of the first browser window and all requester windows.

Snapshot as alternate size Saves the current position and size of the first browser window as the alternate size you get when you hit the *zoom* gadget in the window border.

3.3.7 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.

Registration Retrieves the AWeb-II online registration form at the AmiTrix web site. *You are strongly encouraged to register your copy of AWeb-II.* A TCP stack must be running, and you must be connected to the Internet for this function to work.

3.3.8 ARexx menu

The *ARexx menu* offers functions to start ARexx macros.

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.

User-configurable items The remainder of this menu contains a list of ARexx macros you can configure yourself. Use the ARexx macro menu settings page for this. Please refer to chapter 5 for more information.

3.4 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.

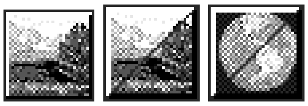

Figure 3.2: The first icon depicts an unloaded image. Click it to load the image. The second icon depicts an unloaded image that is also a link to another document. Click in the upper left half of the icon to follow the link directly, or in the bottom right half to load the image. The third icon depicts an unloaded clickable map. Click on the bottom right half to load the image. Once it is loaded you can pick a spot from the map. You can click in the upper left half of the icon to follow the link without map coordinates. Servers should recognize this as a request to a text-only version of the page.

If an image cannot be loaded for some reason, the error icon is shown.

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 3.3

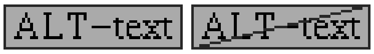

Figure 3.3: The first icon depicts an unloaded image. The second icon depicts an unloaded image that is also a link or a clickable map.

3.4.4 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. This will work even for background images (only if background images are displayed): just shift-click somewhere in the background and you will be asked for a filename to save the background image.

If the image is also a link, shift-clicking the image could be ambiguous; therefore AWeb will save the image in this case. If you want to download the document "behind" the link, you should use the pop-up menu.

- **Show image** - This function shows the image using the external viewer for this image type.

3.5.4 Link functions

If you opened the popup menu for a *link*, it can contain the following functions as long as they are applicable:

- **Open link** - This will show the document behind the link in the current window, just like if you had clicked on the link.
- **Open link in new window** - Open a new window, and display the document in the new window.
- **Load link in background** - This function retrieves the document, but doesn't display it in any window. You will use this function to load a page into the cache, to view it later.
- **Save link** - Save the document behind this link to disk, just like if you had shift-clicked on the link.
- **Download link** - Download the document behind this link and saves it to disk, just like if you had shift-clicked on the link.
- **Add link to hotlist** - This adds an entry to the hotlist for the URL this link points to.

If you open the popup menu for an image that is also a link, the popup menu will contain both sets of functions.

Backwards

This button finds the *previous* occurrence of the search string in the current document.

Cancel

The **cancel** button closes the search requester.

4.2 Starting AWeb

4.2.1 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_name* Look in ENV:AWeb/*settings_name* for the settings. By default the settings in ENV:AWeb are used. If the directory doesn't exist, default settings are used. Saving the settings will create the directory (also in ENVARC:AWeb).

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.

4.2.2 From the Shell

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

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

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

This means that if you use AWeb on a aspect ratio 1:2 screen (like 640 x 256) the printout is about half in height compared to the screen.

Center

When printing with a scale less than 100%, this checkbox allows the printout to be horizontally centered on the paper.

Formfeed

If this is checked, a formfeed is sent to the printer after printing the document. Without formfeeds, you can print several small documents on one sheet of paper.

Print backgrounds

Some documents have a background colour or a background image that may make the printout quite unreadable. This is especially the case when printing in black and wite or greyscale.

If this checkbox is **not** selected, the background of the page will not be printed, but is left white instead. Furthermore, to gain maximum readability all text is printed in black. The printout of embedded images is not affected.

4.4.2 Print in progress

While AWeb is printing the document, the *print in progress* requester is shown. You can cancel the current print with the **Cancel** button. Note that it can take a few seconds before the print is actually cancelled and the requester disappears.

While the document is being printed, you can go on and visit other documents. You cannot start another print though, you will either have to wait until the current printout is ready, or cancel it.

4.4.3 When it doesn't print...

If you run AWeb on a screen with more than 256 colours, printing probably doesn't work. This is not the fault of AWeb, but of the printer driver you are using. Most printer drivers today cannot handle *bitmaps* with more than 256 colours (8 bits).

You will have to change the screenmode that AWeb is using, and optionally select to load a spread palette. Then printing shouldn't be a problem.

As soon as printer drivers become available that can handle more than 256 colours, you can print from screens with more colours directly.

Technically AWeb could convert everything down to 8-bit, but that would be slow (all images on the page have to be remapped), would require more options (like *load spread palette*, and the option to disable the conversion if the user has a >8 bit capable printer driver), the remapping of datatypes would need a real

4.6 The hotlist

4.6.1 Adding a page

The AWeb hotlist can be used to remember the addresses of interesting pages. Whenever you find an interesting page that you think you would like to visit again in the future, you should add the page to the hotlist. Do this by pressing the *add to hotlist* button, or using the Hotlist / Add to hotlist menu item.

4.6.2 Using the list

Press the *hotlist* button, or use the Hotlist / Show hotlist menu item to load a page with all links to all remembered pages.

With the Hotlist button gives requester setting, you can make the *hotlist* button pop up a requester. This is the same requester as for the Hotlist / Maintenance... menu item. The functions in this requester are described below.

4.6.3 Maintenance

When you choose the Hotlist / Maintenance... menu item, a requester will open. In this requester you can change entries in the hotlist, move them around, group entries or delete them.

Use the window close gadget, or the **Esc** key to close the requester.

Change an entry

Select the entry you want to change, either with the mouse or with the cursor up and down keys. Now you can change the name of the entry and the URL that it points to.

Move an entry

Select the entry you want to move, either with the mouse or with the cursor up and down keys. Now you can move the entry up and down in the list with either the arrow up and arrow down buttons, or with the **Ctrl** + cursor up/down keys. The entry will step into open groups, but will skip closed groups.

If you move a group title, the entire group will move.

Add an entry

Use the *Add a link* button in case you want to add an entry to the list and type the name and URL yourself.

4.7 The history window

4.7.1 Purpose

In the history window, you can check all the pages you have visited in the current AWeb session. You can select old pages to display again. The history window offers filtering and ordering options.

4.7.2 Opening the window

Select Navigate / Window history from the menu to open the history window. The display will be filtered automatically for the window in which you selected the menu item.

If the history window is already open, it will pop up to front, and will be activated. Also, the selected window will change to the window for which you opened the history.

4.7.3 The history list

In the list, all visited pages are shown.

The **first column** contains the *window number*. This is the same number as appears in the title bar of the window.

The **second column** contains the *mainline indicator*. This is a little arrow: . Suppose you retrieve pages **A** - **B** - **C** - **D**, then go back to **C**, back to **B**, and then retrieve page **E**. In this case the mainline is **A** - **B** - **E**. These are the pages that the back and forward buttons will walk along.

The **third column** contains the *title* of the page, or the URL if no title is known.

If you selected *natural* or *mainline* order, the current document will be highlighted. The two fields below the list show the title and URL for the highlighted page.

4.7.4 Redisplay a page

Select the page you want to see again from the list. You can click on the entry, or use the arrow keys to move the highlight. Then click the *display* button, or hit the **Enter** key.

Another way to redisplay a page is to doubleclick on the entry.

If you have the History window auto close setting selected, the window will close automatically.

The menu item Cache / Flush authorizations lets AWeb forget all your authorization details.

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

4.9 Cachebrowser

4.9.1 Purpose

With the *cachebrowser*, you can check which files are (still) in the cache. You have options to view the cached objects, save them to another file, or delete them from the cache.

4.9.2 Opening the window

Select Cache / Cachebrowser... from the menu to open the cachebrowser window. If the cachebrowser window is already open, it will pop up to front, and will be activated.

4.9.3 The cache list

The list shows all cached objects. The following information is shown for each object:

- **URL** - the Url (address) of this object.
- **Date** - the date when it was retrieved.
- **Size** - the size of the object in bytes.
- **Type** - the MIME type of the object.
- **File** - the file name of the object in the cache directory.

4.9.4 General information

Below the list there is a fuelgauge showing how full the cache is, compared to the configured cache size.

Also the number of files and the total occupied disk space is shown.

4.9.5 Sort by

Use the **Sort by** chooser to sort the entries in the list by their URL, their date (most recent will be shown first), or by their MIME type.

Chapter 5

Configuring AWeb

5.1 Settings Requesters

5.1.1 Purpose

In the settings requesters, you can change many aspects of AWeb. The changed settings can be saved, used until the next reboot, temporarily tested, or cancelled.

There are three settings requesters, each controlling a different aspect of AWeb:

- **Browser settings** - Everything directly related to the way HTML is shown, including the fonts used and the external viewers
- **Program settings** - Things general for the program, including screen mode, paths, and the ARexx menu
- **Network settings** - Everything related to the network, including the loading of images, plug-ins and the cache

The on-disk documentation contains a useful index to quickly find where a specific setting is located.

5.1.2 Opening the Settings Requesters

A settings requester is opened by selecting the appropriate Settings / *aspect* settings... menu item.

The Settings / ClassAct settings... menu item will open the ClassAct preference requester. You can control a number of aspects for the different requesters here.

NOTE: When opened on a NTSC medium resolution screen (200 pixels in height), some requesters may not be visible completely, depending on the screen font used. If this happens, you should open the *ClassAct* preference requester, go to the *Misc* page, and decrease the number given for the *layout spacing.*

- Test - Use the settings now, but don't save them to disk. The requester remains open. If you cancel the requester, the original settings become active again.
- Cancel - Close the requester without changing the current settings.

5.1.5 Command and arguments

On several locations within the settings requesters, you can enter some sort of external command. For convenience, there are separate string gadgets for the command itself and its arguments.

In the *command* string gadget, you can type in a command. It is advised that you specify the full path, or else AWeb might not be able to execute the command. You can click on the popfile button to pop up a file requester, so you can easily pick the command.

In the *arguments* string you specify the arguments for the command. Any text you type here will be passed to the external program, but some special character combinations will be replaced by parameters from AWeb. Generally the following parameter specifiers are used:

- **%a** = news article or newsgroup name
- **%e** = e-mail address
- **%f** = path/file name
- **%h** = host name
- **%l** = login name
- **%n** = screen name
- **%p** = port number
- **%u** = URL
- **%w** = password

Not all parameters are applicable in every case. You can click the dropdown gadget to pop up a short descriptive list with the possible parameters.

Instead of a program, you can specify a DOS script. If you do, make sure that the *script* bit for this file is set. You can set this bit either by the DOS command *protect script_name +s* or by the Workbench Information program (select the *Script* checkbox).

Note that an external program setting will usually *only* be recognized if *both* fields, command *and* arguments, are supplied.

Change the underlining

The checkbox determines whether links should be displayed underlined or not. If underlined is selected, *new links* are underlined with a solid line, and *visited 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 this changes the behaveour of the gadget but not the appearance, and because it only affects GadTools gadgets which are not always used (because of the limitations of the gadtools library), using such a commodity is not a good idea.

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.

Use background and colours

Many pages contain background images or background colours. AWeb will display these only if this checkbox is checked. If it is unchecked, AWeb will use only the colours defined in the Browser colours settings page.

Use background sound

Some pages have a background sound attached. If this checkbox is selected, the sound plays every time you display the page. If it is unselected, the sound will not be played automatically. You can hear the sound only after selecting the Control / Play background sound menu item.

Popup menu activation

These checkboxes determine the ways to open the popup menu.

If you check one or more of the "SHIFT key", "ALT key" and "CTRL key" checkboxes, the popup menu will open if you click on a link or image while holding one of these keys.

If you check the "middle mouse button" checkbox, the popup menu will also open if you click a link or image with the middle mouse button.

If you deselect all four checkboxes, the popup menu is disabled.

Normal, Bold, Italic, Underlined

Use these checkboxes to pick the rendering style for the current type of element.

5.2.4 Browser settings - Colours

Change the default colours

To change the colour of a link, text or background, select the appropriate row in the list. There are different colour settings for:

- New link - This colour is used for links to pages that you haven't visited yet.
- Visited 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.
- Background - The background colour of a page without a background colour or background image.
- Text - The default colour of normal text on a page.

Then press the **Change colour** button, this will pop up a colour requester. Note that this colour requester (as opposed to the Program 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 that AWeb opens on, the best fitting colours are determined.

These settings are ignored if a page defines its own colours.

Use screen text and background colours

Suppose you don't want any special colour for your background and text, but just the normal screen colours. Instead of trying to get the palette exactly right to match the screen colours, you can simply select this checkbox. Then the palette settings for browser background and text are ignored, and those of the screen are used (unless the page defines its own colours).

5.2.5 Browser settings - Viewers

For objects that cannot be displayed directly in the browser window, AWeb starts an external program, called a *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.

The type of the object, and thereby the particular viewer to start, is determined by the MIME type of the object.

MIME type and subtype

In these string gadgets, you specify the MIME type and subtype. See the About MIME types chapter for more information on MIME types.

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

Use the Command and Arguments fields to specify the viewer command to execute for this MIME type. Argument parameters are:

- %f - file name to "view"
- %n - 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.
- %u - original URL of this object.

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.

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:

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

AWeb is able to pre-load a palette if it opens its own screen. This palette contains an even spread of colours (or shades of gray), 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.

Whith this *chooser* gadget, you have the choice of loading a colour palette, a grayscale palette, or no palette loading.

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. A grayscale will give more acceptible results when there are little colours available.

If you select a grayscale palette, you will probably want to configure your JPEG (JFIF) datatype to use a grayscale mapping for the "AWebIP" task. Otherwise JPEG images might not be displayed.

5.3.2 Program settings - 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.

Hotlist requester auto close

If this checkbox is selected, the hotlist maintenance window will close automatically if you follow a link in that window. If this checkbox is not selected, the window remains open until you close it yourself.

Hotlist button gives requester

If this checkbox is selected, the *hotlist* button will open the hotlist maintenance window. Otherwise, the hotlist is displayed in the browser window.

History window auto close

If this checkbox is selected, the history window will close automatically if you pick a document from the window to display. If this checkbox is not selected, the window remains open until you close it yourself.

Create icons

If this checkbox is checked, AWeb will create an icon for each file that was saved or downloaded. If it is a text file (HTML or plain text), it sets AWeb as the default tool so you can easily view the text with a doubleclick on the icon.

Auto open

These checkboxes determine if the hotlist window, the history window and the network status window should automatically be opened when you start AWeb.

You can use the Settings / Snapshot windows menu item to save the positions where these windows should open.

5.3.4 Program settings - Programs

Editor

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

Use the Command and Arguments fields (see section 5.1.5) to specify your editor command. Argument parameters are:

- %f - file name to edit.
- %n - 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.

in the list with either the arrow up and arrow down buttons, or with the **Ctrl** + cursor up/down keys.

The **Sort** button will sort the entries. Entries with shortcut are moved to the top, in sequence of their shortcut. Entries without shortcut are moved to the bottom and are sorted alphabetically by their title.

5.4 Network settings

The network settings requester can control the following aspects:

- Options - Various general network related settings, including the loading of images.
- Programs - The external programs or *plug-ins* to use, e.g. for FTP or mailto.
- Proxy - The proxies that AWeb should use.
- Cache - The cache size, cache mode and other cache related settings.

These various aspects are explained in the section 5.4.1 through 5.4.4.

5.4.1 Network settings - Options

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.

The status of all connections can be viewed in the network status window (see section 4.5).

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.

Use cookies

If this checkbox is selected, the use of HTTP cookies is enabled.

With the HTTP *cookie* mechanism, the server can ask the browser to store some data that was generated by the server. Then the browser sends this data with every request for a page with an URL that falls within a certain range of URLs. This enables complex and powerful transaction sequences. For a definition of cookies, see http://home.netscape.com/newsref/std/cookie_spec.html.

5.4.2 Network settings - Programs

Protocol "plug-ins"

Of all possible internet protocols, AWeb understands only HTTP: and Gopher: by itself. There are browsers available (mostly for the PC, but some for the Amiga) that handle other protocols internally, like FTP:, mailto: and news:. However, handling these protocols internally would make the executable bigger, and will never offer the same ease of use as dedicated software. Therefore AWeb offers the possibility to configure external programs that should be started when you follow a hyperlink using one of these protocols.

Use the chooser to select from one of the protocols listed below. Then use the Command and Arguments fields to specify your plug-in command and arguments.

You can configure plug-ins for these protocols:

- **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 on-disk documentation.

  Argument parameters are:

  - %e - e-mail address to send the mail to.
  - %n - screen name that AWeb is running on, in case your mail program supports opening on a public screen.

- **Start TCP** This command is used to start your TCP stack automatically.

  For example, if you use the AmiTCP/IP package, you would set *command* to: AmiTCP:bin/startnet and leave the *arguments* field blank.

  Argument parameters are:

  – %n - screen name that AWeb is running on, in case your TCP script or program supports opening windows on a public screen.

  Important note: The start TCP command should return when the TCP connection is active or fails. If your TCP program remains active during the session, you cannot use it directly. You have to use a script instead, like the example script in the on-disk documentation.

- **End TCP** This command is used to stop your TCP connection after your confirmation.

  There are no argument parameter substitutions. **Important notes:** If you used the example script as *Start TCP* command, the on-disk documentation provides another example script to stop your TCP program.

5.4.3 Network settings - Proxy

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

First, choose the protocol you want to define the proxy for. Use the chooser for this, it has 4 options:

- HTTP
- FTP
- Gopher
- Telnet

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

All sizes should be given in kB.

Minimum free memory

These two gadgets determine the minimum amount of chip and fast memory that AWeb should leave free. If there is less memory free than what is set here, AWeb will flush objects from memory. Usually these objects remain on your harddisk.

If you specify more memory than can possibly be freed (e.g. a non-zero fast memory limit on a machine with no fast memory), AWeb will continuously flush all documents and images. So be careful what you enter here.

Sizes are in kB.

Do not cache these sites

If this list contains *site names* (also known as *locations* or *host names*), then documents and images from these sites will not be stored in the cache.

Use the **Add** button to add a new entry, and the **Del** button to delete the currently selected entry. Double-click on an entry to edit the name *in place*.

5.5 Use your own GUI buttons

5.5.1 Configurable buttons

The buttons in AWeb's main window are fully user-configurable. They are read from the file Images/def_buttons. This must be a picture file, but the exact format isn't relevant as long as you have a datatype installed that can read the file.

The icon that goes with the file is important. The tooltypes in the icon contain vital information, without this information AWeb cannot use the file.

You can also use your own transfer animation shown in the *background status* gadget.

5.5.2 Installing your own buttons

The Storage/Buttons drawer contains a few alternate sets, and it is a good place to store new button sets.

To install new button images, open AWeb's drawer on your Workbench. Now you can double-click the *Install_Buttons* icon. It will open a file requester from which you can pick your button set. The install procedure will then copy the selected set to your Images drawer.

Alternatively, you can click the *Install_Buttons* icon once, hold the shift key and double-click on the icon of the button set you want to install.

exact format isn't relevant as long as you have a datatype installed that can read the file.

The icon that goes with the file is important. The tooltypes in the icon contain vital information, without this information AWeb cannot use the file.

The animation is shown in the *background status* gadget. this gadget will be sized according to the size of the animation. For the best look, you should use an animation that is about twice as high as your button images.

5.6.2 Installing your own animation

The Storage/Animations drawer contains a few alternate animations, and it is a good place to store new animations.

To install a new animation, open AWeb's drawer on your Workbench. Now you can double-click the *Install_Animation* icon. It will open a file requester from which you can pick your animation. The install procedure will then copy the selected file to your Images drawer.

Alternatively, you can click the *Install_Animation* icon once, hold the shift key and double-click on the icon of the animation you want to install.

NOTE: You cannot install a new animation while AWeb is running.

5.6.3 Creating your own animation file

Use an ordinary paint program to design your animation. Create one picture that contains all animation frames. Make sure all button images are equal in size. Also every next frame must be at the same horizontal and/or vertical distance from the previous. Save the picture in the Storage/Animations drawer. Save it with icon, or add an icon later.

It is a good idea to keep the picture file as small as possible. The larger the file is, the longer it will take to load at startup time of AWeb.

Then add some tooltypes to the icon. The tooltypes tell AWeb where in your picture to look for the animation frames. Use the following tooltypes:

- **SIZE=***width,height*
  This defines the width and height of each animation frame, in pixels.

- **FIRST=***left,top*
  Defines the top left corner of the first animation frame.

- **FRAMES=***number*
  This is the total number of frames in the animation. After the last frame the first one is used again.

- **DELTA=***delta_x,delta_y*
  This is the amount to add to the current x and y coordinates to get to the next frame.

Chapter 6

Advanced topics

6.1 HTML modes

6.1.1 About HTML

Most of the documents ("pages") found on the World Wide Web are written in *HTML* (HyperText Markup Language). HTML was originally designed as a standard hard- and software independent way of formatting documents. It is an application of *SGML* (Standard Generalized Markup Language).

The only official standard is HTML 2.0, which contains very limited possibilities. The W3 consortium was developing a new, extended standard, HTML 3.0.

Meanwhile, in the recent boom of Internet and the World Wide Web, some browser manufacturers have introduced several ad-hoc extensions to HTML, of which many didn't fit in the new HTML 3.0 standard. Probably because this would make HTML 3.0 an academic standard without any practical use, the development of HTML 3.0 was abandoned. AWeb supports some of the extensions found in HTML 3.0.

In the first half of 1996 the W3 group proposed a new HTML 3.2 standard, which contains many of the widely used NetScape and Microsoft Internet Explorer specific extensions. AWeb fully supports HTML 3.2.

The large browser manufacturers have introduced other tags, that aren't included in the HTML 3.2 standard. AWeb will try to support most of these non-standard extensions.

To make things even more inconvenient, some earlier versions of popular PC browsers didn't stick to the SGML rules. And even recent versions of those browsers still have problems with SGML comments. Because many people design their pages using these browsers, there are many documents on the web that just are bad HTML.

6.2 Cache usage

6.2.1 Why a cache is useful

When you are web-surfing, you often find that there are pages that you visit every time again. Most of the time these pages haven't changed, so it is really a waste of network bandwidth to fetch the page (and possibly the images on it) every time. The *cache* is used to store the most recently visited pages and images on your hard disk. The next time you visit the same page, it is already on your computer. There is no need to retrieve it again over the network, thus saving time and bandwidth.

Another benefit of having a cache is, that after you have disconnected from the network, the pages remain on your computer. You can then browse through them again *off-line*.

6.2.2 How the cache works

AWeb uses a two-stage caching system, both in memory and on disk. The main cache is on disk, and for the most recently used objects there is also a copy in memory. This way you will always get the maximum response speed.The term "object" is used here to indicate documents (pages) and inlined images shown within documents.

Whenever you click a link (or type an URL), or an embedded image on a page is needed, AWeb first looks in its cache to see if it is already there. If the object is in memory, it is used immediately. If it is still on disk but not in memory, it is loaded back into memory. If it is not on disk, it is fetched over the network.

A *reload* of an object never uses the cached copy, but fetches the object from the original location again.

6.2.3 Verifications

Of course, if the original object has been updated since it was stored in the cache, the cache copy should be updated as well. Therefore, when AWeb uses a cached copy of an object, it also sends a verification request to the server, to see if the object was modified. If not, AWeb will use the cache copy, and if the object *is* modified, AWeb will fetch the new version.

You can control how often AWeb will check with the server:

- **Verify always**
  Every time AWeb uses an object from the cache, it will also send out a verification request. This guarantees that you always see the most recent version. Obviously it also generates a lot of network activity, so normally you won't use this. Only if you visit sites with many documents that are updated very frequently you may want to use this option.

- **Verify once per session**
  Only when AWeb uses an object from the cache for the first time after you

- Downloaded objects, they are stored at the location where you save them.
- Objects that cannot be displayed by AWeb itself, but are passed to an external viewer. These objects are saved in the temporary path, and are deleted when the viewer exits.
- Objects from sites listed in the do not cache these sites list.

6.2.6 Cache directory

To take maximum advantage of AWeb's cache, the cache directory should be located on your hard disk. It is possible to configure AWeb to use a drawer in your Ram disk for its cache, but that would take up much memory and everything will be lost when you turn off your machine.

Should your machine crash (or be turned off) while AWeb was still active, then the next time you start AWeb the cache will be recovered automatically.

AWeb stores its cache files in several subdirectories. That way the number of files in each directory is kept low, which will result in a much faster cache when used with certain file systems (like AFS).

Note that you should not use the cache directory for anything else. Also, don't modify or delete any files. If you accidentally deleted some files from the cache, or if you suspect that it has become corrupt, you should use the Cache / Fix cache... menu item. It will synchronize the internal registration with the files actually on disk. **NOTE:** This function will delete any files from the cache directory that are not belonging to AWebs cache.

6.2.7 Cachebrowser

You can use the cachebrowser (see section 4.9) to see what files are in the cache, and optionally view, save or delete them.

6.3 ARexx interface

This version of AWeb has ARexx support.

6.3.1 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.

Note that the first window can be closed while later windows are still open, so it is not guaranteed that the port AWEB.1 exists. Use the following code fragment to determine a valid AWeb port:

6.3.4 Information retrieval commands

GET ITEM/A,VAR/K 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. Note that due to an ARexx limitation only the first 65535 bytes are returned.
- **TITLE** Retrieve the title of the document. If no title was defined in the document, the document's URL is returned.
- **SCREEN** Retrieve the name of the screen that AWeb uses to open its windows on.
- **ACTIVEPORT** Retrieve the name of the ARexx port associated with the most recently activated AWeb window. Useful in ARexx macros started via a shell script or shell command to find the active window.

The information is returned in the reserved variable RESULT, unless the *VAR* argument is used to specify the variable name.

CACHE ITEM/A,FROM/A Get information about AWeb's cache. The *ITEM* argument determines the information to return, and the *FROM* argument determines the object to get information for. *ITEM* can be one of the following:

- **URL** Get the URL for a file in cache. *FROM* must be the name of a cache temporary file, with or without path.
- **NAME** Get the file name for a file in cache. *FROM* must be the name of a cache temporary file, with or without path. The file name returned is the last part of the URL, after the last slash.
- **TEMP** Get the temporary file name for a URL. *FROM* must be a URL of a document or image in cache. Note that no URL parsing is done, so the URLs must match exactly. Also note that many documents won't have a corresponding cache file.

The information is returned in the reserved variable RESULT.

6.3.5 Control commands

ALLOWCMD Temporarily allow shell commands and ARexx macros to be started from hyperlinks regardless of the Allow shell commands setting. This allows easier usage of ARexx plug-ins without need for the user to explicitly change the settings.

Commands are allowed only in the current document, or in the document that is being loaded. After a new document is loaded in the window, the normal settings apply.

ACTIVATEWINDOW Make this window the active window.

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

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.

6.4.1 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.

6.4.2 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.

6.4.3 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

6.5.1 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 hotlist of AMosaic 2.0 prerelease. It uses the file ENV:mosaic/.mosaic-hotlist-default.

x-aweb:ibhotlist/*path* Identifies othet hierarchical hotlists, like other AWeb hotlists or the format used by early prereleases of IBrowse. Because this hotlist doesn't have a fixed location, you must specify the full path and file name in the URL.

6.5.2 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.

6.6 Overview of supported HTML

For a complete overview of supported HTML, please refer to the on-disk documentation.

machine. The solution is to use either a lower or a higher speed ("baudrate") to communicate with your modem. Start at the modem speed (so if you own a 14k4 modem, start at 14k4 communication speed). Then increase the speed until you find a dramatic slowdown. Then go one step back. Changing the communication speed with your modem might involve adapting the settings for your dialler, TCP stack and/or your SANA driver. Refer to the respective documentations.

- *Backgrounds in AWeb are terribly slow, much slower than competitive Amiga browsers.*
  This might be caused by an unsuited picture datatype. Use the 24-bit picture datatype **only** on CyberGraphics screens; on original Amiga screens this datatype becomes very slow. Use the original picture.datatype from your Workbench diskette.

- *The colour requester from the program settings: Palette page messes things up on a CyberGraphics screen.*
  This is caused by the palette-orientated design of the Amiga OS. This design works fine on Amiga chipset screens, but isn't really suited for graphics cards.

- *I run AWeb on a small screen, and the save, use, test, cancel buttons in this settings window fall outside the screen.*
  Start the ClassAct preference program, go to the "Misc" page, and set "layout spacing" to a lower number.

- *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.

7.2 Release history

For the complete release history, please refer to the on-disk documentation.

- **Jamie Mears**
  for further enhancing of the logo

I thank for their translations and/or proofreadings:

- Anders Bakkevold (Norwegian)
- Jean-Michel Bezeau (French)
- Pavel Bures (Czech)
- Gabriele Favrin (Italian)
- Ole Friis (Danish)
- Jose Roberto Gonzalez Rocha (Spanish)
- Ulf Holm (Swedish)
- Pantelis Kopelias (Greek)
- Richard Marti (German)
- Marcin Orlowski (Polish)
- Lasse Pedersen (Danish)
- Vit Sindlar (Czech)
- Miloslaw Smyk (Polish)

Some of the above people are member of the Amiga Translators' Organization (ATO) (http://www2.dk-online.dk/users/Ole_Friis/Trans/Index.HTM). (Although some did the translation after a direct contact with me, not as member of ATO.)

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

- Osma Ahvenlampi ("Tau")
- Jeroen Oudejans
- Thomas Tavoly ("aTmosh")
- Vincent Groenewold ("supernov")
- Donovan Janus
- Mike Meyer
- Paul Kolenbrander
- Donald Voogd
- Kristian Phillips
- Dale Currie