AWeb

Version 2.1, November 1996

The Amiga Web Browser!

Contents

Chapter 1

Introduction

1.1 General introduction

AWeb is a fast and powerful World Wide Web browser for the Amiga computer. It offers the following features:

- AWeb can use a wide range of TCP-stacks: AmiTCP/IP, I-Net225, AS-225 or compatible. The TCP stack can be started automatically when it is needed. 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-3.2 standard is (almost) fully supported, including tables, image alignment, colours, etc. For buggy pages not conforming to the standard, AWeb offers a compatibile 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, Mailto, Telnet and News 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.
- AWeb supports background sounds on pages, provided you have a datatype for the sound installed.

- 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

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.

1.2.4 Distribution

The AWeb-II package, containing the AWeb browser, is distributed by AmiTrix Development.

Distribution otherwise than via AmiTrix Development is not allowed without prior written permission from both the author and AmiTrix Development.

1.3 Registration

Please be sure to register your copy of *AWeb-II*. Support, upgrades, and any updates made available by AmiTrix will only be provided to registered purchasers of the product. Registration may be done by completing the online registration form in the on-disk docs, or by completing and returning the enclosed registration card.

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.

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.

2.2.2 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).

2.3 Tips for 2MB Amiga users

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

- Close all unnecessary windows, applications, etc.
- Use AWeb on the default public screen.
- Set image loading*Off*, and only load images you really want to see by clicking the icons.
- Set backgrounds and background sound*Off*.
- Do *not* install custom GUI buttons or transfer animation but use the built-in defaults.
- Set maximum number of network connections and maximum number of disk reads both 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:
  - cache size memory: about 50% of the free memory when AWeb is running.
  - cache size disk: as much as you like.
  - minimum free chip: about 50kB.
  - minimum free fast: 0 kB.

2.3 Tips for 2MB Amiga users

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.

Chapter 3

Getting started

3.1 Overview

Figure 3.1 presents an overview of the AWeb graphical user interface (GUI). Note that the actual size of the buttons and their imagery can vary depending on your set-up. The relative location of the buttons remains the same. The various components are described in more detail in the next sections.

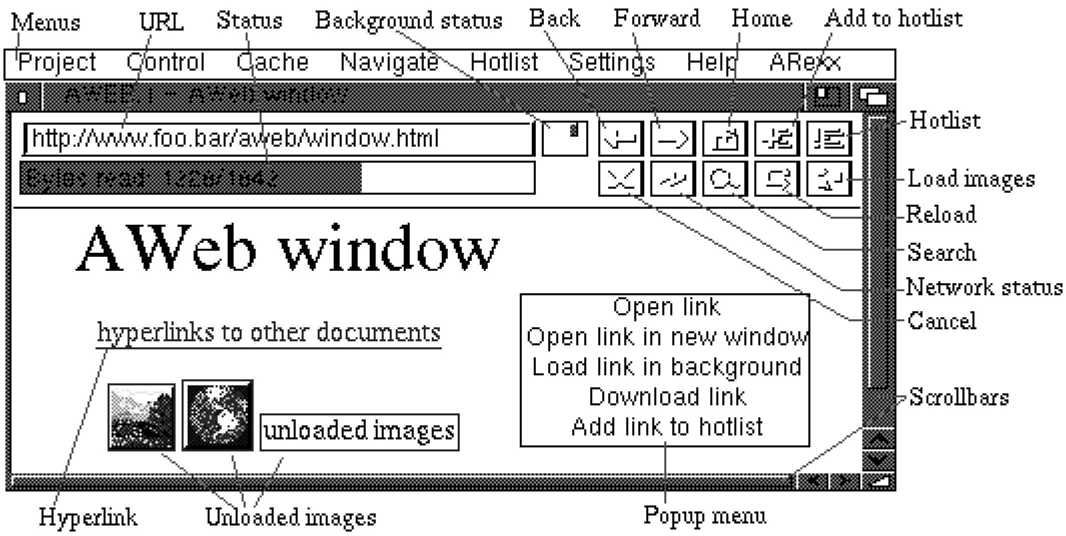

Figure 3.1: An overview of the AWeb graphical user interface (GUI).

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

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, **AF**, or the **Alt + cursor right** key combination, will do the same.

Home button

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

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

Add to hotlist button

This button adds the current document to your hotlist.

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

Hotlist button

This button shows your hotlist.

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

You can configure AWeb so that this button opens the hotlist maintenance window instead.

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 (see section 4.5) instead.

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

Network status button

This button will open the network status window (see section 4.5), or bring it to front if it is already open.

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

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:

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

Print... Opens a print parameters requester (see section 4.4) to allow a graphical print of the current document.

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

Quit Quit AWeb. All pending network operations are cancelled.

3.3.2 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.

Play background sound Plays the background sound attached to this page. If the sound is still playing, it will be stopped now and played again from the start.

Network status... Open the network status window (see section 4.5), or bring it up to front if it is already open. Pressing the *network status* button will do the same.

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

> **Images in current** Reloads the images in the current document. The images are retrieved again from their original source.

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.

Previous window Activates and brings up to front the previous window.

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.

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.

3.3.5 Hotlist menu

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

Add to hotlist Add the current document to the end of your hotlist. Pressing the *add to hotlist* button will do the same.

Show hotlist Show the hotlist in the browser window. Pressing the *hotlist* button will do the same.

Maintenance... Opens the hotlist maintenance window. You can configure AWeb so that the *hotlist* button will do the same.

Save hotlist Save any changes made to the hotlist now. Changes will be saved when you quit AWeb automatically, but sometimes you might want to save changes earlier.

Restore last saved Reverts the hotlist to the last saved version.

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.

3.3.6 Settings menu

The *Settings menu* offers functions to configure AWeb. For more information on settings please refer to chapter 5.

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.

Backgrounds This menu item provides a quick way to set the background colors and images setting.

Background sounds This menu item provides a quick way to set the background sounds setting.

Browser settings... Brings up the browser settings requester.

Program settings... Brings up the program settings 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.

3.4.1 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.

3.4.2 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 Options page in the browser settings requester.

Images that are also links have a frame drawn around them. If you have deselected the link underlining in the browser 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.

3.4.3 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 3.2 shows these different icons and their meaning. You can also tell AWeb to use your own images here.

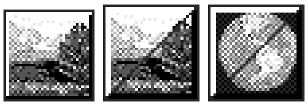

Figure 3.2: The first icon depicts an unloaded image. Click it to load the image. The second icon depicts an unloaded image that is also a link to another document. Click in the upper left half of the icon to follow the link directly, or in the bottom right half to load the image. The third icon depicts 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

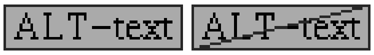

Figure 3.3: The first icon depicts an unloaded image. The second icon depicts an unloaded image that is also a link or a clickable map.

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.

3.4.5 Pop-up menu

If you hold the **ALT** or the **Ctrl** key while you click on a link or an image, AWeb opens a pop-up menu with several very useful choices that were not easily accomodated otherwise.

You can configure AWeb to open the popup menu also if you hold down the **SHIFT** key. The popup menu is described in more detail in section 3.5.

3.5 The popup menu

3.5.1 Purpose

Given a link or an image, there are a number of things you might like to do with it. You might want to follow a link, download it, open it in a new window, load or flush an image, etc. The *popup menu* offers a way to choose from all these different functions in a clear and easy fashion.

3.5.2 Using the popup menu

To open the popup menu, hold the **ALT** or the **Ctrl** key and click on a link or an image. Or, you can click on a link or an image using the middle mouse button if you own a three-button mouse. The menu opens and presents you a list of choices depending on the type of object you clicked on. Then you can click on the desired function.

You can configure AWeb to open the popup menu also if you hold down the **SHIFT** key, or not to open the popup menu if you hold down the **ALT** or the **Ctrl** key.

To close the menu without choosing a function, you can click the right mouse button, or click anywhere outside the popup menu.

3.5.3 Image functions

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

- **Load image** - Load the image, just like if you had clicked on the image.
- **Reload image** - Reload this image.
- **Save image** - This will save the image, just like if you had shift-clicked on the image.
- **Download image** - This will download the image, just like if you had shift-clicked on the unloaded-image icon.
- **Flush image** - Flush this image from memory.

- **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.

Chapter 4

Working with AWeb

4.1 Searching for text

You can open the search requester by clicking the *search* button in the button bar, or with the Project / Search... menu item.

Note that each window can have its own search requester. Each requester will have the same window title as the browser window.

Search for

Type the text to search for in the **Search for** field.

Ignore case

If this checkbox is selected, AWeb will search for the text disregarding upper and lower case. If it is unchecked, upper and lower case characters will be treated as different.

Search

The **Search** button finds the *next* occurrence of the search string in the current document. The window contents will be scrolled so that the text found is on top of the window. Note that the document cannot scroll beyond the last full window height, so if there is matching text in the last part of the document it isn't scrolled to the top of the window.

You can use the **Enter** key as an alternative for the **Search** button.

From top

The **From top** button finds the *first* occurrence of the search string in the document.

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

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 name in the CONFIG argument will be used as subdirectory name relative to ENV:AWeb to look for the settings. By default the settings in ENV:AWeb (no subdirectory) are used. If the directory doesn't exist, default settings are used. Saving the settings will create the directory (also in ENVARC:AWeb).

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.

4.3 Starting your TCP connection

Before you can access documents from the World Wide Web, you must start your *TCP connection* with the Internet. The connection is maintained by the TCP package you are using, also known as *TCP stack*.

You can start your TCP stack yourself before you start AWeb, or while AWeb is running. To make life easier, you can even let AWeb start your TCP stack when it is needed for the first time. For example, if you are browsing through local documents, and then follow a hyperlink to some document out there on the World Wide Web, then AWeb could start the TCP stack automatically.

To use this feature, you have to configure the start script (or program) for your TCP package.

If you have configured the stop script (or program) for your TCP stack, AWeb can terminate the TCP stack automatically after you quit AWeb. Note that it does so only if AWeb has started the TCP stack before, and only after asking for your permission to do so. If you started the TCP stack yourself, AWeb will not terminate it.

4.4 Print a document

4.4.1 The print parameters requester

Before AWeb starts printing the current page, you can alter some parameters. You open the *print parameters* requester with the Project / Print... menu item.

Printer parameters that cannot be controlled in the print parameters window can be changed using the standard Workbench *PrinterGfx* preferences utility.

Scale (%)

Here you can change the scale for printing the document. Printing at 100% takes up the full paper width. The document is always printed with an 1:1 (square) aspect ratio, because many if not all pages are designed for this ratio.

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 visible screen instead of the off-screen bitmap AWeb uses now (datatypes need a Screen to remap themselves to), and would of course make the executable bigger. I don't think applications should kludge around a limitation in other software this way.

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:

1. By clicking the *network status* button
2. By selecting the **Control / Network Status...** menu item
3. 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 *cancel* button at the bottom left.

You can cancel all transfers by clicking the *cancel all* button at the bottom right.

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.

Group entries

Use the *Add a group* button to create a group, then change its name. Now you can add entries to the group, or move existing entries into it. Entries within a group will be displayed indented in the list. You can include other groups within a group, to any level.

Click on the little arrow next to the group title in the list to open or close the group. If the group is closed, its members are not visible, making the list clearer. Also, you can use the **Enter** key to open and close a group if the title is selected.

Remove an entry

Select the entry you want to remove, either with the mouse or with the cursor up and down keys. Now you can remove the entry with the *Remove* button. If you remove a group, all members will be removed too.

Follow a link

Select the entry for the document you want to load, either with the mouse or with the cursor up and down keys. Now you can load the document with either the *Follow* button or with the **Enter** key. Another way to follow a link is to double-click on it.

If you have selected the Hotlist requester auto close setting, the requester will close if you follow a link. Otherwise the requester stays open.

4.6.4 Save and restore

If the hotlist was changed, it will be saved automatically when you leave AWeb. You can save it earlier with the Hotlist / Save hotlist menu item. The hotlist is also saved after you have added a link.

Use the Hotlist / Restore last saved menu item to revert the hotlist to the state it was in when it was last saved.

4.6.5 Using other hotlists

You can specify a different hotlist to use with the HOTLIST tooltype or argument.

You can load another AWeb hotlist using the Hotlist / Other... menu item. The hotlist will be shown as document, not in the hotlist maintenance requester.

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.

4.7.5 Filtering

If the *Filter* checkbox is selected, the list will only show entries for the window number selected in the *window* field.

If you deselect the *Filter* checkbox, the list will show entries for all windows.

4.7.6 Ordering

With the *Order* chooser, you can select how the entries in the list should be ordered.

Natural Natural ordering shows all displayed pages in the order you have displayed them. If you go back a few pages, then go to another page, the latest page is added at the end of the list. If you redisplay a page that was displayed before, it is also added again to the end of the list.

Mainline The list shows only the pages on the mainline. That is, entries for which the *mainline indicator* is off will not be shown.

Retrieved The list contains the pages in the order as they were retrieved for the first time. The list will show every page only once.

Title The entries in the list are sorted by their title.

URL The entries in the list are sorted by their URL (network address). The list will show URLs instead of titles if you select this ordering.

4.8 User authorization

4.8.1 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.

4.8.2 Save your authorization details

AWeb remembers all authorization details and saves them to disk when you quit. The next time you start AWeb, your authorization details are read so you won't need to enter these again.

To save these data during your session, select the Cache / Save authorizations menu item.

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.

4.9.6 Open an object

If you have selected an object from the list, you can use the **Open** button to load the page in the current browser window. If it is an image, the external viewer for that image (as configured in the viewers settings) will be started.

Instead of the **Open** button, you can double click on the entry, or use the **Enter** key.

4.9.7 Save an object

If you have selected an object from the list, you can use the **Save** button to save the object to another file. You will see a save requester where you can enter the file name.

4.9.8 Delete an object

If you have selected an object from the list, you can use the **Delete** button to delete the object from the cache. Note that if the object is currently being displayed, it will not be deleted until it is no longer displayed.

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*.

5.1.3 Different settings

By default, AWeb stores its settings in the ENVARC:AWeb drawer. If you used the CONFIGtooltype or argument, the settings are stored in the ENVARC:AWeb/*config_name* drawer instead. If this drawer doesn't exist, it will be created.

5.1.4 Controlling the settings requesters

Menus

Each settings requester has the following menu items:

- Project menu
  - Open... - Read settings from a different file.
  - Save as... - Save the settings in a different file.
  - Quit settings - Close this settings requester.
- Edit menu
  - Reset to defaults - The settings in this requester are reset to their default values.
  - Last saved - The settings in this requester are reset to the values as you have saved them the last time.
  - Restore - The settings in this requester are reset to the values as they were
- Windows menu
  - Browser settings... - Open or activate the browser settings requester.
  - Program settings... - Open or activate the program settings requester.
  - Network settings... - Open or activate the network settings requester.
  - ClassAct settings... - Open or activate the Classact preference window.
  - Snapshot - Store the current window positions.

Buttons

Each settings requester has some standard buttons:

- *Clicktabs* - From the row of so-called *clicktabs* you can pick a *page* with related settings.
- Save - Save the current settings to disk, and close the requester.
- Use - Close the requester and use the current settings, but don't save them. After a boot, the last saved settings are active again.

- 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.

Example

Look at the *HTML source viewer* setting (in the Program: Programs settings). Suppose you want to install MultiView as viewer, and let it open its window on the same screen as AWeb.

First, you click on the popfile button and navigate to the **SYS:Utilities** drawer. Then you select **MultiView**. Now the *Command* is set up correctly.

Then click on the dropdown gadget and hold the mouse button. You will see:
%f = path/file name
%n = screen name

MultiView can take the arguments: FILE,SCREEN/S,PUBSCREEN/K. Therefore you enter **%f PUBSCREEN %n** in the *Arguments* field. The **%f** will be replaced by the real file name when the command is executed by AWeb. Likewise, **%n** will be replaced by the actual screen name.

So your settings would look like this:
Command:SYS:Utilities/MultiView
Arguments:%f PUBSCREEN %n

If you like MultiView to run on its own screen, you would set it up this way:
Command:SYS:Utilities/MultiView Arguments:%f SCREEN

5.2 Browser settings

The browser settings requester can control the following aspects:

- Options - Some general options that control the behaveour and looks of the browser.
- Fonts - The fonts that should be to AWebs disposal.
- Styles - The font type, size and style to use for various elements on a HTML page.
- Colours - The colours to be used for various elements of a document.
- Viewers - The external viewers to use for various types of objects.

These various aspects are explained in the section 5.2.1 through 5.2.5.

5.2.1 Browser settings - Options

HTML mode

Use this chooser to select the HTML mode. It is recommended that you leave this set to *tolerant* unless you encounter problems with a page.

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.

5.2.2 Browser settings - Fonts

Font type

AWeb uses two sets of fonts, a normal set (usually a set of proportional fonts), and a set of fixed-width fonts. Use this chooser to switch between these two sets.

Font list

Each font set contains fonts in seven different sizes. Size 1 is the smallest, size 7 is the largest size. Normal text is usually rendered in size 3.

Select a font size from the list, and use the Ff button to bring up a standard font requester. You then can change the font name and size.

Set all to this font name

If you press this button, all font names are changed to the name of the currently selected font. The sizes remain unchanged.

5.2.3 Browser settings - Styles

Styles list

For each HTML style tag you can configure which font type (normal of fixed-width) and size should be used, and the style to use.

the list contains all relevant tags and the current styles for that tag. Below the list is a brief description of the meaning of the currently selected tag.

Fixed width

Select this checkbox if you want this tag to be rendered using the fixed-width font instead of the proportional font.

Abs size

For header tags (<H1> through <H6>) you can enter here a font size from 1 to 7 here. Size 1 is smallest, 7 is largest.

Rel size

For all other tags, you can enter a relative size here. Valid relative sizes are -6 through +6. The actual size used for this type of element will be the current normal size in the document, adjusted by this number. Note that the resulting size will never be less than 1 or greater than 7.

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.

For each MIME type you want to be recognized by AWeb, there should be an entry on this page with the appropriate viewer named.

About MIME

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 and 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. 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:

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.

5.3 Program settings

The program settings requester can control the following aspects:

- Screen - The screen that AWeb should open on: default public screen, another public screen, or let AWeb open its own screen.
- Palette - The screen colours for AWeb's own screen.
- Options - The save and temporary paths, and various other general options.
- Programs - The external editor and HTML source viewer to use.
- ARexx - The entries to appear in the ARexx macro menu.

These various aspects are explained in the section 5.3.1 through 5.3.5.

5.3.1 Program settings - Screen

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

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.

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.

5.3.3 Program settings - Options

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.

Temp path

This is the path used for temporary files. These include files passed to external viewers, and images that are not cached.

Scroll overlap

If you scroll up or down by a page, there is some overlap. Because 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.

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.

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.

HTML source viewer

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

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

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

5.3.5 Program settings - ARexx

Changing menu entries

The list contains the menu entries that should appear in the ARexx menu.

Select the entry you want to modify from the listview. Use the **Add** button to add a new blank entry. Use the **Del** button to remove the selected entry.

Title

Type the title as you like it to appear in the ARexx menu.

Shortcut

You can select a shortcut from the chooser, or leave it to *none* if you don't want this entry to have a shortcut. Available shortcuts are **A**1 through **A**0.

You can assign different entries the same shortcut, but the shortcut will actually work only for the topmost menu item.

Macro

In this gadget, you can type in the name of the ARexx macro to execute. You can click on the button to pop up a file requester, so you can easily select the macro.

Rearranging menu entries

You can move an entry in the menu. Select the entry, 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 **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).

Max. disk reads

Although you may have set the number of network connections to a reasonable maximum, it could cause disk trashing when you try to load local files.Disk trashing is the phenomena you may encounter when two or more files from the same device are read simultaneously. Most of the time the disk is busy moving its read heads to and from, causing the speed of data transfer to drop dramatically.

To avoid this, you can limit the number of local disk reads independently.

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.

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.

Start with homepage

If this checkbox is checked, AWeb will retrieve the page defined as your home page immediately when you start the program. If this option is not checked, you will initially get an empty window.

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 external viewers section, 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.

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.

- **FTP**
  An *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.

  Argument parameters are:

  – %h - host name to connect to
  – %f full path and file name to fetch
  – %n screen name that AWeb is running on, in case your FTP program supports opening on a public screen.

- **Telnet**
  A *telnet:* address is used to start a telnet session. If you own a telnet client that supports command line arguments. you can use this feature. A telnet address can contain: telnet://username:password@hostname:port

  Argument parameters are:

  – %l - login name (user name) to use for logging in to this host
  – %w - password to use with this user name
  – %h - host name
  – %p - port number
  – %n - screen name that AWeb is running on, in case your telnet program supports opening on a public screen.

  If your telnet client does not support all these parameters, you can use the example script included with the on-disk documentation. It will pop up a requester with the user name and password to use, if these were given in the address.

- **News**
  A *news:* address points to a newsgroup or an article in a newsgroup. If you own a news reader that supports command line arguments or ARexx commands, you can use this feature.

  Argument parameters are:

  – %a - newsgroup name or article identification
  – %n - screen name that AWeb is running on, in case your news program supports opening on a public screen.

Start and stop your TCP stack

AWeb is able to start your TCP connection automatically when it is needed.

Use the chooser to select one of the entries below. Then use the Command and Arguments fields (see section 5.1.5 to specify the necessary command and arguments.

- **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.

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

Limited proxy usage

Some proxies can't handle submitting forms with METHOD=POST, or authorized pages correctly. If you have problems with such pages, try selecting this checkbox. It makes AWeb handle these pages directly, without going through the proxy.

No proxy for these sites

If this list contains *site names* (also known as *locations* or *host names*), then documents and images from these sites will never be retrieved through the proxy.

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*.

5.4.4 Network settings - Cache

Cache path

AWeb uses a cache on your harddisk to store objects (for more information please refer to section 6.2.1). It will use the directory you configure here as its cache directory. Note that you should create a new directory, or use an empty one for AWebs cache. Don't use the cache directory for anything else but AWebs cache.

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

Note that if you change this setting it does not become active when you **Use** or **Save** the settings. In order to avoid an inconsistent cache, AWeb will continue to use the current cache directory until you quit AWeb. The next time you start AWeb the new directory will be used.

Verify cache copy

Use this chooser to set the verification mode. *Once per session* is the recommended setting.

Fast response

Check this checkbox if you want fast response from the cache.

Cache size

Use these two gadgets to set the maximum size to use for the cache in memory and on your harddisk.

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.

NOTE: You cannot install new buttons while AWeb is running.

5.5.3 Creating your own button file

Use an ordinary paint program to design your buttons. Create one picture that contains all button imagery. Make sure all button images are equal in size. If you want different *selected* imagery (shown when the button is depressed), draw those images too. Save the picture in the Storage/Buttons 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 which button image. Use the following tooltypes:

- **BUTTON_SIZE=***width,height*
  This defines the width and height of each image, in pixels.
- **BACK=***nx,ny,sx,sy*, **FORWARD=***nx,ny,sx,sy*, **HOME=***nx,ny,sx,sy*, **ADDHOTLIST=***nx,ny,sx,sy*, **HOTLIST=***nx,ny,sx,sy*, **CANCEL=***nx,ny,sx,sy*, **NETSTATUS=***nx,ny,sx,sy*, **SEARCH=***nx,ny,sx,sy*, **RELOAD=***nx,ny,sx,sy*, **LOADIMAGES=***nx,ny,sx,sy*
  These tooltypes define the location of the button imagery in the picture, for each button.*nx,ny* are the x and y positions of the top left corner of the *normal* image.*sx,sy* are optional. If present, they tell the location of the *selected* image for this button. If a button definition is missing from the tooltypes, the built-in default image is used for that button.
- **NSBUTTON_SIZE=***width,height*
  This defines the width and height of each network status button image, in pixels. Default is the same width and height as the other buttons.
- **NSCANCEL=***nx,ny,sx,sy*, **NSCANCELALL=***nx,ny,sx,sy*
  These two tooltypes define the location of images for the *cancel* and *cancel all* buttons in the network status window.
- **TRANSPARENT**
  If this tooltype is present, and the picture was saved as a transparent brush, the images will be rendered with a transparent background. Note that this only works if the datatype supports transparency. Most regular datatypes do, one important exception is the ILBM datatype that comes with the V43 picture datatype.

5.6 Using your own transfer animation

5.6.1 Configurable animation

The transfer animation in AWeb's main window is fully user-configurable. It is read from the file Images/def_transferanim. 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.

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.

- **REST=***left,top*
  Defines the top left corner of the *rest* image, that is displayed if no transfer is going on. This tooltype is optional. If you don't specify it, the animation gadget will be empty at rest.

5.7 Using your own default images

5.7.1 Purpose

There are moments where just displaying text is not sufficient, but some imagery is needed. For some of these imagery, AWeb uses external images that you can replace with your own.

AWeb uses these external images in the following places:

- As default icons for unloaded images.
- For displaying WWW icon entities like &document.

All external images are located in the Images drawer. These are image files, that can be in any format for which a datatype is available. Transparent GIF's are also supported.

5.7.2 Unloaded images

For unloaded images, AWeb uses the following files:

- filename : def_image, used for unloaded normal image.
- filename : def_imagemap, used for unloaded image map.
- filename : def_errimage, used for images that couldn't be loaded.

5.7.3 Icon entities

For all supported icon entities, there is a file with the same name. So for &document; there is a file named document in the Images drawer.

Look at the overview in the on-disk documentation for all supported entities.

5.7.4 Installing your own images

To install your own image, just copy your image over the old one. Be sure to use the same name, and don't use a suffix like .iff or .gif.

The Storage drawer contains some alternatives for the icon entities, but you can use any image of course.

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.

6.1.2 HTML modes

AWeb does its best to display all pages correctly, but sometimes you have to set the way HTML should be interpreted to get the best results.

You can choose out of three *HTML modes*:

- **Strict**
  In *strict* HTML mode, AWeb understands only the proposed HTML 3.2 standard.
- **Tolerant**
  In *tolerant* HTML mode, AWeb understands also other extensions. These are both extensions specific to other browsers, and HTML 3.0 extensions that can't be found in HTML 3.2. Also, in tolerant mode some commonly made HTML errors are correctly understood (like leaving out <TD> in a table or putting <FORM> between <TABLE> and <TR>).
- **Compatible**
  In *compatible* HTML mode (a nice way of formulating "buggy"), AWeb tries to interpret buggy HTML. Below is a description of the deviations of the standard when compatibility mode is used.

Set one of these in the Options page in the browser settings.

The overview of supported HTML in the on-disk docs mentions with each HTML tag or attribute in which mode it is recognized.

6.1.3 Compatible mode

As mentioned above, some pages contain bad HTML. When you view such a page, it can 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 *compatible* HTML mode of AWeb. **Warning:** Using compatible HTML mode, documents containing *valid* HTML might look distorted in turn.

In compatible mode, AWeb exposes the following deviations from the SGML standard:

- quoted attribute values are terminated by any occurrence of ">"
- quoted attributes that contain URLs, like HREF, SRC and ACTION, are terminated by whitespace
- comments are terminated by any occurrence of "->"

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

started AWeb, it will check with the server. The second and later times the object is used without verification. This allows AWeb to update its cached objects regularly, but does not produce excessive network overhead. This is the default verification mode.

- **Verify never**
  AWeb will just use cached copies of objects if they are available. It will never check with the server if the object has changed. If you know or suspect that the cached copy is no longer up-to-date, you should use the reload function.

  When you browse *off-line* through cached pages with AWeb set to one of the other response modes, AWeb silently falls back to *verify never* mode. So it won't complain that it cannot connect to the server, and won't try to start your TCP stack just to verify cached objects. Of course it *will* do so when you try to fetch an object that is not in cache.

6.2.4 Fast response

Most browsers implement the verification mentioned above so that they first ask the server, and when the object turns out to be unmodified, then they use the cached copy. As a consequence, with *every* link you follow, you have to wait a few seconds until the network connection is set up and the server responds. You have to wait with every verification, even if the object is still in cache and up-to-date.

In most cases, the cached copy *will* still be valid. After all, that is one of the reasons for having a cache in the first place. So slow verification will let you wait many times totally unnecessary.

When using **fast response** mode, AWeb *always* uses the cached copy if one is available. The cached document or image is displayed immediately. In the same time that the object is displayed, AWeb connects to the server and verifies if the object is indeed not modified. In most cases the copy will still be up-to-date, and these connections disappear silently after a few seconds. This way you will gain a few seconds with every link you follow to a cached document.

In the few cases where the object was modified, you will first see the old version that was in cache. Then after a few seconds, when the server responds, the new updated version will be shown in the window, much like as if you had hit the reload button. If this behaveour confuses you or annoys you, you should not use fast response mode.

6.2.5 Uncached objects

All documents and inlined images *from the net* are stored in the cache. The following objects will *not* go into the cache:

- Objects from your own machine (file://localhost). Because these objects already reside on your harddisk, there is no reason to store them again in the cache.

- 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:

ports = SHOW('P')
PARSE VAR ports dummy 'AWEB.' portnr . /* note the trailing period! */
ADDRESS VALUE 'AWEB.' —— portnr

Once you have got a port, you can use the GET ACTIVEPORT command to get to the active window:

OPTIONS RESULTS
'GET ACTIVEPORT'
ADDRESS VALUE RESULT

6.3.2 ARexx commands

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

6.3.3 Transfer commands

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 cache.

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

The reserved variable RESULT will contain the port name of the new window.

RELOAD Reload the current document.

WAIT DOC=DOCUMENT/S,IMG=IMAGES/S,ALL/S Wait for transfer completion.

- **ALL** Wait for transfers in all windows and background. Without the ALL switch this command waits only for transfers related to the addressed window.
- **DOCUMENT** Wait for document transfer only, no inlined images or downloads.
- **IMAGES** Wait for transfer of inlined images only. DOCUMENT and IMAGES may be combined. With no DOCUMENT or IMAGES specified, the command waits for all transfers.

SAVEAS NAME,APPEND/S Save the HTML source of the document.

If *NAME* is given, the source is saved under this name. If the *APPEND* switch is set, the source is appended to the file, otherwise the file will be overwritten.

If no *NAME* is given, a save requester will pop up.

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.

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

SCREENTOFRONT Move the screen that AWeb is using in front of all other screens.

SCREENTOBACK Move the screen that AWeb is using to the back of all other screens.

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.

6.3.6 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.

6.3.7 Include ARexx macros in the menu

See section 5.3.5.

6.3.8 Start ARexx macros from a hyperlink

See section 6.4.

6.4 Shell command and ARexx macro interface

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.

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 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 below.

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 below for one possible way of parsing this.

6.4.4 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 ARexx OPEN 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.

6.4.5 Examples

For examples please refer to the on-disk documentation.

6.5 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 other hotlists from within your hotlist, or even configure one of these as your home page within AWeb.

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.

Chapter 7

Miscellaneous

7.1 Common problems

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

- *AWeb won't start. It says it can't find xxx.image or xxx.gadget.*
  Probably you haven't installed the latest ClassAct classes. You can find them on ftp://ftp.warped.com/pub/amiga/classact/

- *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.

- *When I try to print the document it just does nothing*
  You probably try to print from a screen with more than 256 colours. Most printer drivers can't handle these kind of screens. You have to change your screen mode used first. **Tip:** Create a separate configuration (using the CONFIG tooltype or argument) with a 256 colour screen mode. When you like to print a page, first continue surfing, and after you have quit AWeb start it again with the other configuration. Now you can fetch the document from cache and print it off-line.

- *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 use HTML mode compatible.

- *AWeb becomes terribly slow if I run it on its own screen.*
  This has something to do with chip memory usage and DMA bandwidth. Apparently, the problem becomes worse when you have an accelerated

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.

7.3 Known bugs

Although AWeb is thoroughly tested, it is very likely that there are some bugs left. At the time of release the following bugs were known. For the latest information, check the AWeb support page at http://www.networkx.com/amitrix/aweb.html.

7.4 Things to do

For the to-do list, please refer to the on-disk documentation.

7.5 Contact

You can contact AmiTrix Development

- by e-mail: support@amitrix.com
- by snail-mail:
  AmiTrix Development
  5312 - 47 Street
  Beaumont, Alberta
  T4X 1H9
  CANADA
  Phone or fax: 1+ 403-929-8459
- through the Web: http://www.networkx.com/amitrix/index.html

For more information on ordering AWeb-II or other AmiTrix products, contact us at the above address, or by e-mail: sales@amitrix.com.

Dealer inquiries are welcomed.

7.6 Acknowledgements

I especially wish to thank:

- **Osma Ahvenlampi** ("Tau")
  for the TCP stack-adaptive design
- **Josef Faulkner**
  for writing many useful ARexx add-ons
- **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

- **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

7.6 Acknowledgements

- Christopher Aldi
- Josef Faulkner
- David Göhler
- Jason G Doig

AWeb-II was written by Yvon Rozijn.