AWeb

Version 2.1, November 1996

The Amiga Web Browser!

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.

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.

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.

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.

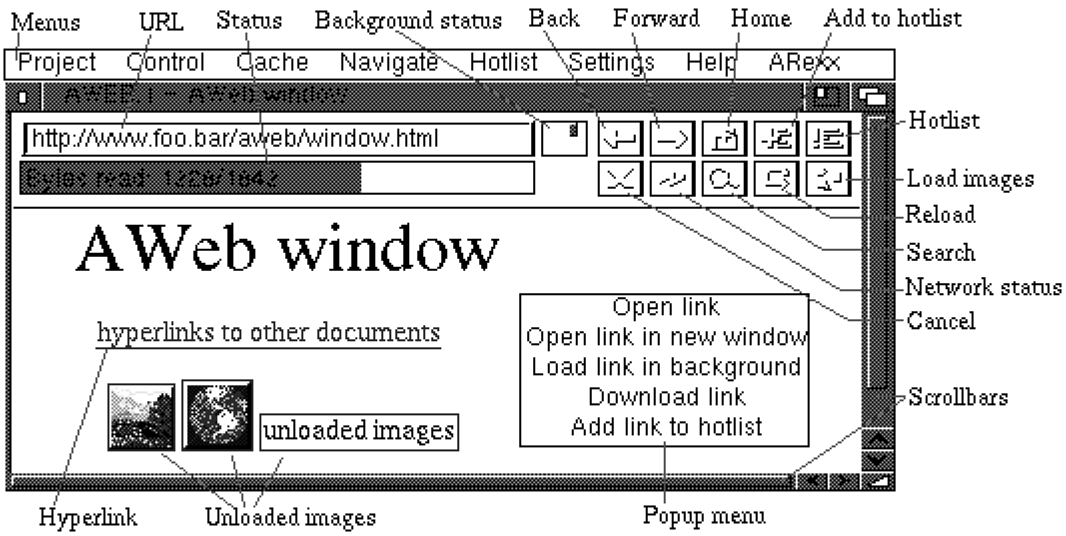

Figure 3.1: An overview of the AWeb graphical user interface (GUI).

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.

- 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.

3.3.5 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.

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.

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.

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.

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.

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.

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.

4.7.5 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.

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.

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.

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.

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.

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.

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.

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.

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

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

- **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.

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.

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

- **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.

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 "->"

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.

```
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.

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 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.

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

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

7.6 Acknowledgements

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

AWeb-II was written by Yvon Rozijn.