



Version 1.1, April 1996

The World Wide Web Browser for the Amiga computer

©1996 by Yvon Rozijn

---

---

3.6.1	From the Workbench . . . . .	16
3.6.2	From the Shell . . . . .	16
<b>4</b>	<b>Using AWeb</b>	<b>18</b>
4.1	The Graphical User Interface . . . . .	18
4.1.1	URL field . . . . .	18
4.1.2	Status indicator . . . . .	19
4.1.3	Background status indicator . . . . .	19
4.1.4	Back button . . . . .	19
4.1.5	Forward button . . . . .	19
4.1.6	Home button . . . . .	19
4.1.7	Add to hotlist button . . . . .	19
4.1.8	Hotlist button . . . . .	20
4.1.9	Cancel button . . . . .	20
4.1.10	Network status button . . . . .	20
4.1.11	Reload button . . . . .	20
4.1.12	Load images button . . . . .	20
4.2	Menus . . . . .	20
4.2.1	Project menu . . . . .	20
4.2.2	Control menu . . . . .	21
4.2.3	Cache menu . . . . .	22
4.2.4	Navigate menu . . . . .	22
4.2.5	Hotlist menu . . . . .	23
4.2.6	Settings menu . . . . .	23
4.2.7	Help menu . . . . .	24
4.2.8	ARexx menu . . . . .	24
4.3	Cache usage . . . . .	24
4.3.1	Documents . . . . .	24
4.3.2	Images . . . . .	25
4.3.3	Low memory . . . . .	25
4.3.4	Setting the cache sizes . . . . .	25
4.4	The browser window . . . . .	26
4.4.1	Scrolling the page . . . . .	26
4.4.2	Following links . . . . .	26
4.4.3	Inlined images . . . . .	27
4.4.4	Downloading . . . . .	27

4.10.6 Examples . . . . .	43
4.11 Extension URLs . . . . .	44
4.11.1 Hotlists . . . . .	44
4.11.2 Window history . . . . .	44
4.11.3 Shell commands and ARexx macros . . . . .	44
<b>5 Problems, comments and suggestions</b>	<b>45</b>
5.1 Known AWeb bugs . . . . .	45
5.2 Common problems . . . . .	45
5.3 To do . . . . .	46
<b>6 How to contact the author</b>	<b>47</b>
<b>7 Acknowledgements</b>	<b>48</b>

## 1.5 Distribution

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

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

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

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

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

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

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

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

## 2.3 Registration form

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

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

### 3.1 Why not use MUI ?

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

#### 3.1.1 Visual feedback

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

HOW DO I GET AMOSAIC TO RUN ON ITS OWN SCREEN? I believe that the 18 (eighteen!) steps to perform to accomplish this, are far too many.

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

### 3.1.4 Minor objections

#### Size and speed

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

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

#### Shareware

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

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

### 3.1.5 Flames?

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

## 3.2 ClassAct

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

## 3.3 System requirements

AWeb needs the following to run:

### 3.4.2 Installer problems

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

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

### 3.4.3 Configuring the JFIF datatype

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

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

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

## 3.5 Tips for 2MB Amiga users

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

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



---

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.

### 4.1.2 Status indicator

This field serves two purposes.

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

Second, when a page is being loaded for this window, 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.

### 4.1.3 Background status indicator

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

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

### 4.1.4 Back button

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

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

### 4.1.5 Forward button

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

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

### 4.1.6 Home button

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

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

### 4.1.7 Add to hotlist button

This button adds the current document to your hotlist.

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

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

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

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

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

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

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

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

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

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

### 4.2.2 Control menu

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

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

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

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

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

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

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

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

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

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

#### 4.2.5 Hotlist menu

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

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

**Show hotlist** Show the hotlist in the browser window.

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

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

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

#### 4.2.6 Settings menu

The *Settings menu* offers functions to configure AWeb.

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

- All images
- Maps only
- Off

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

**Change settings...** Brings up the settings requester.

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

### 4.3.2 Images

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

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

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

### 4.3.3 Low memory

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

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

### 4.3.4 Setting the cache sizes

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

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

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

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

### 4.4.3 Inlined images

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

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



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

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

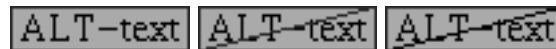


Figure 4.3: The first icon depicts an unloaded image. The second icon depicts an unloaded image that is also a link. The third icon depicts an unloaded clickable map. The difference between the second and third icon is minimal. Please refer to the on-disk documentation for a more clear view of these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

### 4.4.4 Downloading

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

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

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

## 4.6 User authorization

### 4.6.1 Authorization

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

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

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

### 4.6.2 Save your authorization details

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

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

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

## 4.7 Compatibility mode

### 4.7.1 Incorrect HTML

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

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

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

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

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

### Change the underlining

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

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

### Change the cycle field

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

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

## 4.8.4 Screen 1: Screen

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

### Default public screen

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

### Named public screen

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

### Own public screen

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



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

#### 4.8.6 Screen 3: Palette

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

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

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

This settings page has two major parts.

##### Pen settings

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

##### Change palette

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

#### 4.8.7 Network 1: General

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

##### Image loading

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

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

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

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

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

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

### Allow Shell commands in links

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

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

## 4.8.8 Network 2: Proxy

On this settings page, you can configure proxy servers.

### Proxy servers

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

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

### Configuring a proxy

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

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

## 4.8.9 Network 3: External programs

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

### 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.

### Caches

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

## 4.8.11 Program 2: External programs

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

### Editor

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

Use the Command and Arguments fields to specify your editor command. Argument parameters are:

**first %s** = file name to edit.

**second %s** = screen name that AWeb is running on, in case your editor supports opening on a public screen.

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

### HTML source viewer

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

Use the Command and Arguments fields to specify your source viewer command. Argument parameters are:

**first %s** = file name to edit.

**second %s** = screen name that AWeb is running on, in case your viewer supports opening on a public screen.

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

## MIME type and subtype

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

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

## Extensions

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

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

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

## Processing

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

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

**first** %s = file name to "view"

**second** %s = screen name that AWeb is running on, in case your external viewer supports opening on a public screen. Use this only if you want it to open on the same screen as AWeb.

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

### 4.8.13 Example

Suppose you want to see JPEG images using the VT program, and other images using the MultiView program on its own screen. You know that JPEG files can have extensions **jpeg**, **jpg**, or **jiff**, 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:

### 4.9.2 ARexx commands

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

Available commands are:

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

**RELOAD** Reload the current document.

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

**URL** Retrieve the URL of the document.

**SOURCE** Retrieve the HTML source of the document.

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

The information is returned in the reserved variable **RESULT**.

**ACTIVATEWINDOW** Make this window the active window.

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

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

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

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

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

### 4.9.3 Return values from commands

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

ARexx commands return the following codes:

**0** : Command executed successfully.

**1** : Command executed successfully, 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.

#### 4.10.4 Parameters

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

##### Forms

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

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

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

##### Clickable maps

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

##### ARexx arguments

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

#### 4.10.5 Load the result back into AWeb

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

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

#### 4.10.6 Examples

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

## Chapter 5

# Problems, comments and suggestions

### 5.1 Known AWeb bugs

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

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

### 5.2 Common problems

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

- *I can't install AWeb. The installer reports errors like: Unable to compile script.* Most likely some misbehaving application installation has copied an obsolete version of the Commodore Installer utility to your hard drive. Make sure the Installer version in the SYS:C directory is "installer\_2 2.17 (1993-02-13)" or better. This version can be found on the Amiga Workbench 3.x Install floppy disk. Also, delete any copies of the Installer program from your SYS:Utilities drawer.
- *AWeb crashes when loading a GIF image* The commonly used ZGif datatype version 39.16 had a bug. Be sure to use version 39.18 or better of the ZGif datatype, or another GIF datatype.
- *AWeb doesn't display all of the page. Other browsers display the page correctly* Probably the page contains some erroneous HTML. One of the common errors is the use of something like `<!------->` as a divider 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

## Chapter 6

# How to contact the author

You can contact me

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

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

If you have questions about AWeb, you can also mail to the FAQ: [aweb@dds.nl](mailto:aweb@dds.nl).

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

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

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