

amuw

COLLABORATORS

	<i>TITLE :</i> amuw		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 6, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	amuw	1
1.1	AmigaUW Documentation	1
1.2	AmigaUW/Legal Information	2
1.3	AmigaUW/Begware	2
1.4	AmigaUW/Author Information	3
1.5	AmigaUW/Description	3
1.6	AmigaUW/Installation	3
1.7	AmigaUW/Operation	4
1.8	AmigaUW/uwserverd	5
1.9	AmigaUW/Auw	6
1.10	AmigaUW/uw.device	7
1.11	AmigaUW/TTY Resize Support	8
1.12	AmigaUW/Technical Information	9
1.13	AmigaUW/The Future	9
1.14	AmigaUW/Credits	10
1.15	AmigaUW/TroubleShooting	10

Chapter 1

amuw

1.1 AmigaUW Documentation

Amiga UnixWindows (AmUW)

by Ezra Story

(version 3.0 Copyright (c) 1993,1994)

Mumbo, Jumbo (but please read it anyway)

Description

Ok, What the heck is it?

Legal Information

Distribution Information

Beware

Please? :-)

Basic documentation (how to get started)

Installation

Setting it up.

Operation

How to use it?

Argh! What Now!?! (common complaints and problems)

TroubleShooting

Problems?

Specific documentation (All the cool neat options)

Uwserverd

The Amiga side server.

Auw
The UNIX side server.

Uw.device
serial.device replacement

Extraneous Information (for those interested)

TTY Resize support
For comm program authors.

Technical Information
How it all works, bugs, etc.

Future Enhancements
The *REAL* release.

Author Information
How to contact me, etc.

Credits
Who to blame... :-)

1.2 AmigaUW/Legal Information

Legal Information
=====

The AmigaUW package is Copyright (C) 1993,94 by Ezra A. Story. The package may be freely redistributed subject to the limitations given below. The AmigaUW package may not be redistributed in any form other than the original archive released by Ezra A. Story.

The user of this package accepts the responsibility for any damage, loss of productivity, or loss of money that may occur through or during its use.

1.3 AmigaUW/Beware

Beware Information
=====

The AmigaUW package is beware. What does this mean? It means that if you really like and use this package, you can send me US \$10, but I won't complain too hardily if you don't. (Consider it a donation to the "MultiLink fund" (see Future).

You may send money in the form of cash, or a check or money order payable in US dollars only.

Author Information
Where to send registrations

1.4 AmigaUW/Author Information

If you want to contact me, just use the addresses given below. Email is the preferred method and will receive the quickest response. Drop me a line if you find a bug, have a suggestion, or just want to tell me how cool (or uncool) AmigaUW is. :-)

Email: ezy@panix.com
IRC: Ezy

Snail Mail:
Ezra A. Story
P.O. Box 741
JAF building
New York, NY 10116

1.5 AmigaUW/Description

Description
=====

Amiga UnixWindows (or AmigaUW, or Amuw) is a package designed to allow a person to have multiple concurrent sessions open on a UNIX host over a single serial connection. If you have ever used DNet, Screen, MacLayers, or Term, then you are familiar with what this package does.

*** AmigaUW requires at least AmigaDOS version 2.04 (V37) or better. ***

The package consists of a server on the UNIX host, a server on the Amiga, and a serial device emulator which uses the server on the Amiga to open up sessions on the UNIX host using ordinary terminal programs.

Please read this WHOLE documentation file before starting to use AmigaUW, I have tried to anticipate many of the problems one might encounter when setting it up when writing the documentation.

1.6 AmigaUW/Installation

Installation
=====

Compile the UNIX server on your UNIX host by typing "make" (see the README_FIRST file). If you get any compilation errors, check the makefile, and if you can't figure it out after that, send me a message about it. So far, auw has been successfully compiled on SunOS, NeXTStep, Ultrix, HP/UX, Sgi, and Solaris2. It should work with most BSD and SYSVR4 systems. There is no guarantee it will work with your system however. :-)

Copy the uwserved and client binaries somewhere into your AmigaDOS search path. Copy the uw.device to your DEVS: directory. Set your serial preferences in sys:Prefs/serial to the EXACT same settings that you usually use in your terminal programs. For high speed modems, check the documentation on

uwserved
for options relating to bps
rate and high speed operation.

Please see the
operation
section for information on how to
use AmigaUW now that you've set it up.

1.7 AmigaUW/Operation

Operation

=====

Here's a step by step description of what you need to do to use AmigaUW properly.

#1. Install the programs.

See the
Installation
section. Make sure your serial
preferences (Prefs/serial) are set correctly!!!!

#2. Start the Amiga background server

Type "run uwserved" followed by the options you wish to use (such as -m, for 8bit, -B for a high bps rate..etc). See the documentation for

uwserved
for more information on the options.

#3. Start the first session.

Start your favorite communications program and make it use "uw.device" unit 0. If you're not already on your UNIX host, call up and login.

#4. Start up multisession mode.

Start up AmigaUW by typing "auw" plus any options you may require. See the

Auw

section for information on the options. Make sure to explicitly specify the "-b" option followed by the bps rate you're using to connect to the UNIX host. The bps rate option is critical for good interactive performance. Most users will also want to use the "-m" option for 8bit mode (see the section on auw for more details).

#5. Open up a second session.

Start up another copy of your favorite terminal program, and make it use "uw.device" unit 1. A shell prompt should appear in a few seconds. You are now using multiple sessions! :-). To open up more sessions, just start more copies of your terminal program using different unit numbers of uw.device (unit 2,3,..etc). The limit is currently 7 sessions. See the

uw.device

documentation for more information about the unit numbers.

#6. After you're done, Close down.

To exit multisession mode, just close all the terminal programs you have open, then reopen another terminal program after you've shut all the other ones down (i.e. Go to step #3 :-)). You can logoff, or perhaps restart AmigaUW from this point (Step #4). The "-d" option of uwserverd can be used to autorestart a terminal program when you have closed all the sessions.

#7. Remove uwserverd.

Just send it a control-c by using the "break" command in AmigaShell. This will cause uwserverd to exit and remove itself from memory. The "-t" option can automatically do this for you when you quit as well.

1.8 AmigaUW/uwserverd

Uwserverd

=====

Uwserverd is the program which handles the Amiga side of the Amuw protocol. It is meant to run as a background task which communicates with the UNIX host over the serial.device and with the uw.device client on the Amiga.

Initially, uwserverd allows only one session open. Only when the AmigaUW server is started up on the UNIX host will it allow multiple sessions. This allows you to use uwserverd and uw.device without having to load up a separate communications program in order to start auw on the UNIX host.

When all of the sessions have been exited in multi-session mode, uwserverd causes the UNIX side to exit, and returns to the single session mode. To make uwserverd exit, you must send it a ^C (usually using the "break" command in AmigaShell), or use the option "-t" to have it automatically quit for you.

Uwserverd accepts the options listed below.

- B Default buffer size for each channel. Uwserverd allocates a buffer to hold incoming data for each channel that it opens. The buffer size defaults to 8000 bytes.
- b Bps rate uwserverd will use to communicate with your modem. This is typically used for baud rates you cannot set in preferences, (38400, 56800..etc). The default is whatever you have set in the serial preferences. Note that this sets the MAXIMUM bps rate that your modem can acheive including compression. (ie. for 14400 v42bis you would prolly use 38400).
- d Default client to launch on exit from multiple session mode. If this option is specified with a program/script name, uwserverd will attempt to execute the program/script when it starts up, and whenever it exits multiple session mode into single session mode. This is very useful when you exit all of your comm programs to exit multiple session mode.
- D Uwserver defaults to using the serial.device. If you specify this option followed by a device name, uwserverd will use that device instead.
- H Turns on high speed mode. This is equivalent to the "Rad Boogie" mode of the serial.device. It turns off XON/XOFF checking, sets 8-bit characters on and does not check parity. It is recommended you set this option if you can. The default is NOT to use high speed mode.
- U Uwserver defaults to unit number 0. Use this option to change which device unit uwserverd opens.
- m Turns on eight bit sends to the UNIX host. If the connection you're using can handle 8-bit, is it recommended that you turn this option on. Uwserverd defaults to 7-bit.
- t Quit on last client close. Specifying his option will cause uwserverd to automatically exit when the last uw.device client is closed in the normal, non-multisession mode. Useful for people who are tired of typing "break 6" all the time. :-)
- v Causes uwserverd to print status messages to its standard output.
- ? prints a summary of all the options it accepts.

1.9 AmigaUW/Auw

Auw
===

Auw is the UNIX host side server for AmigaUW. It MUST be run when

uwserverd is ALREADY running on the Amiga. It accepts the two options listed below.

- m Turns on eight bit sends to the Amiga. If the connection you're using can handle 8-bit, it is recommended that you switch this option on. Auw defaults to 7-bit.
- b Specifies the bps rate of your connection to the UNIX host. This option is very important for interactive response. If it is much higher than your actual connection speed, things like Zmodem dloads and long directory listings will effectively lock up all input. Because it is impossible to get the timing exact, you may want to tweak the value you use. For example, on a slow UNIX machine, "auw -mb 2500" may be used instead of "auw -mb 2400" to account for protocol overhead. Auw defaults to a setting of 19200.

In general, the setting / 10 will be the average cps you typically get plus 10-20 for overhead. Here are some typical settings for typical connections. NOTE: connection types do NOT always correspond to modem types. (ie. a v.32bis modem connects at only 9600 to a v.32 modem, not 14.4k.) Plus, the upper limit is set by the UNIX terminal server. You may connect at 2400 MNP 5, but your terminal server might only communicate with the UNIX host at 2400, for example.

connection type	amuw bps setting
-----	-----
2400	2500
2400 MNP 4	2600
2400 MNP 5	2600
2400 v.42bis	2800
9600	9800
9600 v.42	10000
9600 v.42bis	10200
14400 v.42	16000
14400 v.42bis	17000

Once run, Auw does not exit until told to do so by the Amiga. Usually, this is done by the uwserverd, but if you are communicating with it directly for some reason (after the Amiga crashed, or if you accidentally started it without uwserverd running), you can cause it to exit by typing Control-A followed by DEL.

1.10 AmigaUW/uw.device

uw.device

=====

Uw.device is a serial.device replacement which allows an ordinary terminal program to take advantage of the AmigaUW multiple session capability. Uw.device uses uwserverd to do its work, so uwserverd must be running in order to use uw.device!

To open up a session using a terminal program, just use "uw.device". To close a session, just close the device by exiting the terminal program or by using the terminal program's serial port settings to close the port. The unit number you use with uw.device does not make a difference, and many times you may want to vary it to get around things such as OwnDevUnit library (see

TroubleShooting
).

To open up more sessions after "auw" has been started, open more copies of your terminal program using uw.device. To exit AmigaUW, just close all your open sessions, including the one you used to start AmigaUW. Uwserverd will reset, and auw will exit. You may then reopen uw.device to logoff or whatever.

1.11 AmigaUW/TTY Resize Support

TTY Resize Support

=====

Uw.device also supports AmigaUW's automatic TTY resize by using a special IOREquest command called UWCMD_TTYRESIZE. This IOREquest should be sent to uw.device whenever the dimensions of the communications window change.

```
#define UWCMD_TTYRESIZE (CMD_NONSTD)+20
```

```
uw.device/UWCMD_TTYRESIZE
```

```
uw.device/UWCMD_TTYRESIZE
```

NAME

Resize -- Change the remote's idea of the terminal dimensions.

FUNCTION

This command causes uwserverd to tell the remote UNIX host (auw) the new terminal dimensions. If the UNIX host supports the proper ioctl(), the terminal will be automatically resized.

IO REQUEST

```
io_Message      mn_ReplyPort initialized
io_Device       set by OpenDevice
io_Unit         set by OpenDevice
io_Command      UWCMD_TTYRESIZE
io_Data         high word set to the width in characters.  Low word
                set to the length in characters.
io_Length       high word set to the width in pixels.  Low word set
                to the length in pixels.
```

RESULTS

1.12 AmigaUW/Technical Information

Technical Information

=====

AmigaUW is based on the UnixWindows protocol designed by John D. Bruner for a multiseession UNIX connection. It will actually work with the original uw program in protocol 1, but it is not recommended as even this early release of AmigaUW has some enhancements that the normal UW protocol doesn't, such as TTY resize.

AmigaUW's protocol is stream based, does NO error detection or correction, and has no priority system. It is not a networking (link) protocol, and it is not meant as such. It is meant as a simple low-weight protocol to give multiple terminal sessions over a single serial line to a UNIX host, thats it.

AmigaUW only supports 7 simultaneous channels.

AmigaUW greatly slows down I/O to/from windows when a long continous send is done from the Amiga to the UNIX host. (such as Zmodem uploads). A quick hack to get around this problem is to use a file transfer protocol which does not continuously send, such as Ymodem or Kermit. Note that downloads work fine with Zmodem, it's just uploads which are hosed. :-)
This is a bug in userverd which would take too much effort to fix. This effort is being directed at bigger better things. :-)

I am working on another package that will fix these problems and add many more features. See

Future

.

1.13 AmigaUW/The Future

Future

=====

I am implementing a successor to the AmigaUW package which will have some of the following features:

- completely based on BSD sockets
- Up to 64 simultaneous channels
- prioritized channels
- able to directly access many internet resources (telnet,ftp..etc)
- can accept connections from the internet (telnetd)
- AS225R2 compatible emulation
- AmiTCP compatible emulation

See the MultiLink.txt file for more information.

1.14 AmigaUW/Credits

Credits

=====

John D. Bruner for the original Macintosh UnixWindows protocol, which served as the base for this package.

Mark Jacyszyn for his comments and beta testing of AmigaUW version 0.99 (The first version which noone except me and him ever saw ;-)).

The whole IRC crew, for providing entertainment and some indispensable information during those long programming sessions.

The numerous people who emailed me with bug reports and suggestions. :-)

And.... Matthew Dillon, whose DNet and dnet.device started it all.

1.15 AmigaUW/TroubleShooting

TroubleShooting

=====

Here's a list of typical problems, and some solutions you can try. If you continue to have trouble, send me some mail with a complete description of your problem, including any information you can give about your configuration.

The UNIX side wont compile properly.

Did you check the makefile? Did you read the README_FIRST file? If you've done both of these things and you still can't get it to work, mail me with a description of the errors you're getting.

My terminal program keeps complaining about uw.device unit 0 being in use.

This is typically a result of the use of owndevunit.library, which blocks attempts to open the same device and unit twice. Try opening uw.device with a different unit number (unit 1, for example). uw.device will still work normally, and OwnDevUnit won't complain about it.

VLT won't start up a second session.

VLT will not start up more than one instance of itself unless you rename it's message port. Start VLT using the option "-n"
See the vlt documentation for more information.

VLT can't open uw.device, even though I started uwserverd.

For some insane reason, VLT opens then closes the device it's using before opening it again to use it. If you're using uwserverd with the -t option, this will cause uwserverd to quit itself before VLT has a chance to use it. So -- DO NOT USE the -t option on uwserverd

if you're using VLT as the initial comm program.

My terminal program can open uw.device just fine, but input or output from the modem behaves strangely, or auw does not start properly.

This is usually a sign that your serial preferences settings on workbench do not match your settings within your terminal program. Userverd uses the serial preference settings to open the serial.device, NOT your terminal program's settings. You need to restart userverd for the settings to take effect if you had to change them. If all else fails, specify the settings directly on the userverd command line using options such as -C or -b.

When I start auw, it prints "8" and nothing happens.

Auw MUST be started from your login shell. Make sure you're not inside of anything such as "screen" or "windows" when you start auw. Make sure you are using uw.device.

Terminus prints "CTS not enabled".

Keep CTS/RTS on in preferences and turn CTS/RTS off inside of terminus. This is not a critical error, but it keeps that annoying message from popping up. :-)

When I logout from a shell in an Amiga UW session the terminal program appears to lock up.

This is normal, since it is impossible for the terminal program to know that you've exited the session. Just make it a habit to end sessions by closing the terminal program instead of logging out of the shell. (This, of course, does not include the login shell).

I've accidentally started "auw" without userverd running. It printed "8" then locked up. How do I exit it without hanging up?

Just type ^A (control-a) following by DEL (the delete key), and auw will exit back to your shell prompt.

Zmodem downloads slow AmigaUW down a LOT. Characters seem to take over 3 seconds to appear on the screen while the download is occurring.

- or -

AmigaUW seems sluggish, even if I have nothing going on in the background, such as a Zmodem download.

Make SURE that you have the bps rate option (-b) for auw set correctly. It should be set to at least the highest base bps rate (2400, 9600, 14400, etc) your modem is capable of (noncompressed), and if your modem has inbuilt compression such as v.42bis or MNP5 you might want to tweak it even higher. For example, I use "auw -mb 16000" for my 14.4k v.42bis modem. You do NOT want the bps rate set too low OR too high, either way will adversely affect performance.

Note that some slowdown is to be expected, but if you have to wait more than 1 or 2 seconds for characters to appear when you type, something

is adjusted wrong.

Zmodem downloads under AmigaUW go at half the rate (cps) they usually do (outside of AmigaUW). Is this normal?

No, it is not. Make sure you have 8-bit mode on (-m) if your connection is capable of it on BOTH uwserverd and auw. Without the -m option, AmigaUW will work over 7bit lines, but your thruput will be cut in half.
