

AWC

NetPorter – Your Internet Gateway

© 2001 by AWC, All Rights Reserved

AWC

310 Ivy Glen Ct.

League City, TX 77573

stamp@al-williams.com

<http://www.al-williams.com/awc/awce.htm>

v1.0 1 May 2001

Table of Contents

Overview.....	1
How it Works.....	1
Using Stimulus.....	5
Error Logging.....	6
Example Rules.....	6
Using Telnet.....	6
Using the Web Server.....	7
Command Line Reference.....	8
Property File Reference.....	9
Rule Command Reference.....	11
Template Tag Reference.....	14
Stimulus File Reference.....	16
About Regular Expressions.....	17
Example.....	19
License.....	20

Overview

The Internet is everywhere, but simple microcontrollers and instruments often don't have the extra horsepower to send data over the Internet. Instead of redeveloping existing applications, it is often easier to use a computer as a bridge between the applications and the Internet.

That's what NetPorter does. Using a sophisticated pattern matching technique, NetPorter can extract information from any RS-232 device and make it available as a formatted file (like a spreadsheet or a Web page), send the data via e-mail, or publish it to a server using standard FTP. NetPorter can also Telnet-enable your device and act as a two-way Web server.

- Parses data input using regular expressions
- Transforms template files into data files
- Directly e-mails the results of template files
- Can send periodic queries by use of a simple stimulus file
- Supports flow control
- Provides optional Web server
- Allows Web pages to send data to the device
- Allows optional two-way Telnet access to the device
- Works behind most firewalls

How it Works

NetPorter is a Java program. Although AWC only certifies it for operation under Microsoft Windows, you may be able to run it on other Java platforms if they support the javax.comm library.

You configure NetPorter using two primary files: a properties file and a rules file. The properties file configures different options such as the COM port you want to use, the baud rate, and several other options. You can also set several defaults in this file. Normally, NetPorter looks for this file in NetPorter.properties. However, you can override this on the command line in order to use a different set of properties. This allows you to start multiple copies to service more than one serial port, for example.

The rules file (normally, rules.txt) specifies how your system presents data to NetPorter. There are at least three different ways you can handle data input:

1. If you are writing a program specifically for NetPorter, you can send your data in the form VARIABLE=VALUE. This will cause NetPorter to set the variable you name to equal the value provided.
2. If your existing program emits data in a particular format, you can use regular expression matching to transform the input into the VARIABLE=VALUE format. You can transform the same input multiple times if you need to parse multiple values.
3. If your existing device requires prompting, you can make NetPorter periodically send commands to the device.

The rules file uses several commands to specify how to deal with incoming input lines. The most important of these is MATCH. The MATCH command accepts a regular expression pattern that must match the input. If the input does not match, rule processing skips to the next MATCH command.

If a MATCH command succeeds, the rest of the commands following it (up to the next MATCH) will execute. Unless, the commands include a NEXT command, pattern matching will continue with the next MATCH command after the current commands execute.

Usually, you'll follow a MATCH command with a SET command this allows you to transform the matching input line into a variable assignment for NetPorter. The variable assignment is always in the form of VAR=VALUE.

Suppose your device periodically sends a weight as an ASCII string followed by a carriage return, that you want to capture using NetPorter. Your rule file would just match any data that arrives and then transform it into an assignment for your variable WEIGHT. To make this happen, you could use the following simple rule file (the italicized text is not part of the file):

```
MATCH (.* )           match the whole line
SET WEIGHT=$1         set variable
```

Suppose the scale sent two weight, separated by a colon. You could write:

```
MATCH (.*):(.* )
SET WEIGHT1=$1
SET WEIGHT2=$2
```

The \$1 refers to what matches the first parenthesis (counting from the left) and the \$2 refers to the second match.

Once you have these variables, what can you do with them? You apply them to a template to generate either an e-mail or a file on your local file system. You do this with the BUILD command. This command specifies a template file. This file is exactly like the file you want to create except it contains <VAR> tags that NetPorter will replace with the variable you specify. Here is a simple template file:

```
The current weight is <VAR WEIGHT1>.
```

You can also format numbers and show the date and time that a variable changed by using special syntax in the template file.

Once you have used the BUILD command you can use the SAVE command to save the resulting file to your file system, or APPEND to append the results to an existing file. In addition, a SEND command will send the result as e-mail. You can use multiple SAVE, APPEND, and SEND commands if you wish. The SEND command operates in the

background so sending mail doesn't halt processing any longer than necessary.

After saving a file, you can use the PUBLISH command to send it to an FTP server. Here is a rule file that would split two weights save a log file using APPEND, and mail each weight to different users. Finally, it publishes the log file to a Web server.

```
MATCH (.*):(.*?)
SET WEIGHT1=$1
SET WEIGHT2=$2
BUILD log.template
APPEND log.htm
BUILD mail1.template
SUBJECT Weight 1 Results
SEND alw@al-williams.com
BUILD mail2.template
SUBJECT Weight 2 Results
SEND super@al-williams.com
PUBLISH ftp.al-williams.com log.htm
```

You can place comments in a rule file by preceding them with the # character. As another example, consider the case where the device sends an A or B depending on which weight it is sending. It also sends the string ozs after the number. You could write:

```
MATCH A(.*?)ozs
SET WEIGHT1=$1
BUILD mail1.template
SEND alw@al-williams.com
MATCH B(.*?)ozs
SET WEIGHT2=$1
BUILD mail2.template
SEND super@al-williams.com
```

Variables can contain any string, not just numbers. The regular expressions recognized in the rule file are similar to Perl regular expressions, so you have many powerful constructs available to match whitespace, words, and special characters.

Using Stimulus

You can enable stimulus by specifying a stimulus file in the properties file (using the stimulus variable). This file is similar to the rules.txt file, but it uses different commands. If you do not specify a stimulus file, NetPorter will not send any data via RS-232.

BEGIN – Send the string when NetPorter starts and do not send it again.

SEND N text – Send the text every N seconds.

PREFIX – Set the prefix text for the next BEGIN or SEND command.

If you wish to include special characters for BEGIN or SEND you may use \r (carriage return), \n (new line), \b (backspace), \t (tab), \s(space), or \\ (backslash).

The PREFIX command is useful when your device doesn't emit anything but raw ASCII numbers. For example, suppose you have a scale that you poll for two different weights. If you send an A (followed by a carriage return) you get the first weight, and a B returns the second weight. Without the PREFIX command, NetPorter can't tell which number is which. Of course, you'd need to stagger the timing of the requests so they don't overlap. For example:

```
PREFIX WEIGHT1=  
SEND 10 A\r  
PREFIX WEIGHT2=  
SEND 15 B\r
```

Now the data will be in the exact format required for NetPorter.

Error Logging

By default, NetPorter reports informational and error messages to the system error stream (usually the console). However, you can provide a second argument to the program on start up to direct it to use a log file. If you put a plus sign in front of the log file's name, NetPorter will append messages to the existing log file.

Each message has a time and date stamp. It also has a specific code and detailed messages to help pinpoint problems. Some of these messages will be easy to understand (file not found, or host unreachable, for example).

Example Rules

Here are some regular expressions you might find useful:

Match two numbers separated by commas and white space.

```
MATCH \s*([0-9]+)\s*,\s*([0-9]+)\s*
```

Match a floating point number.

```
MATCH (([+-]?[0-9]*\.[0-9]*)|([0-9]*))
```

Match a 3 digit number in the 10th position.

```
MATCH ^.....([0-9][0-9][0-9]).*$
```

Match a 24-hour time.

```
MATCH ([0-2]?[0-9]:[0-5][0-9])
```

You can find more examples at <http://www.all-williams.com/awce/npexample.htm>.

Using Telnet

By default, NetPorter does not act as a Telnet server. However if you set a port number in the telnetport property, NetPorter will listen for Telnet connects on that port. The default port for Telnet is 23, however most programs will allow you to use any port.

You should also set `telnetpassword`. The server will challenge connected clients to provide this password (after the user presses Enter). Once the password is entered correctly, NetPorter will send each line that it sees on the serial input to the Telnet terminal. Anything you type on the Telnet terminal will also be sent to the serial output. If `telnetline` is true (the default) the serial device will see a carriage return/line feed at the end of each line. If the parameter is false, you can use characters such as `\r` and `\n` to insert carriage returns and line feeds in the string (see the SEND command in the Stimulus file for the details).

NetPorter doesn't echo characters, so you should turn your Telnet program's local echo on. It is permissible to have multiple Telnet users connected at one time. However, if multiple users (and a stimulus file) are sending data at once, there is no assurance of the order it will appear at the device and it may become confusing.

If you enter data from a Telnet terminal and it causes the serial device to respond, those responses are echoed to all Telnet terminals and also processed as usual by the rule file.

Using the Web Server

Since NetPorter can update files you can use any Web server to provide access to the generated files. However, you may wish to use NetPorter's built-in server either by itself or in conjunction with another Web server.

By default, NetPorter does not act as a Web server. However, if you set the `wwwport` parameter in the properties file, the program will listen on the indicated port. If you want NetPorter to be the default Web server, set the property to 80. If you want to coexist with another server, you can set the property to an unused port. If your machine is named `darkstar` and you have NetPorter listening on port 808, you could enter `http://darkstar:808/test.htm`, for example.

NetPorter serves files from the default directory, as set by the property file. If you are using the Web server, you must set this property. NetPorter won't accept URLs with `..` in them, so you do not have to

protect files above the default directory, but be aware that files below the default directory could be retrieved via a Web browser.

In the default directory, you can use the following special files:

- index.htm – The default main page, used if no file is specified by the browser.
- error404.htm – The page that appears if there is a request for a nonexistent file.
- error500.htm – The page that appears if there is a server error.

When you request a page via a browser, you can add extra data as a query string, by placing a question mark at the end of the URL followed by the query string. You can do this manually, or by creating an HTML form that uses the GET method. NetPorter recognizes any query string that starts with send=. If it finds this prefix, it takes the remainder of the query string and sends it to the serial device. So to send a reset command to the device, you might use:

```
http://darkstar:808/reset.htm?send=reset\r\n
```

Or you could send any command by using this form:

```
<FORM ACTION=http://darkstar:808/ok.htm>  
<INPUT name=send><INPUT TYPE=SUBMIT>  
</FORM>
```

Command Line Reference

The NetPorter program is a Java program. If you used the Windows installation program, you should have a shortcut ready to use. If you don't change the shortcut, the program will look for NetPorter.properties in the default directory. It will also use the system console to report errors.

If you wish to use a different property file, you can specify it as the first argument on the command line. You can also specify an error log as the second argument. If you want to override the default error log, but use the default property file, use a dash as the first argument:

```
java NetPorter - error.log
```

If the first character of the error log file name is a +, NetPorter will discard the + sign and open the error file in append mode so that any new errors will be at the end of the existing file.

Sending a ? or -? via the command line will print the version number and current CLASSPATH.

Property File Reference

Property file variables are case sensitive. You can prefix comments with a # character. Always use forward slashes (not backward slashes) to separate directory names. For example, use c:/NetPorter instead of c:\NetPorter. Where directory names are called for, do not end them with a slash.

Here are the property file variables:

Parameter	Default	Description
baudrate	9600	Sets the baud rate
binarybytes	1	In binary mode, the number of bytes that make up a "line".
binarymode	FALSE	Causes serial input to be converted into ASCII representation of the hex bytes
bits	8	Sets the number of bits for the COM port (5, 6, 7, or 8)
debug	FALSE	Enables debugging messages
defaultdirectory	none	Default directory for SAVE and APPEND
defaultftpdirectory	none	Default remote directory on FTP server

defaultthost	none	Default FTP server
defaultpassword	none	Default FTP password
defaultport	21	Default FTP port
defaulttemplatedirectory	none	Default directory for templates
defaultuser	none	Default FTP user
eolchar	\r	Default end of line indicator from COM port
flow	0	COM Port flow control – 0 = none, 1 = RTS/CTS, 2=XON/XOFF
fromaddress	none	E-mail address for From field
ftppassive	TRUE	Use passive mode FTP (works behind firewalls but OK for normal use too)
matchcase	FALSE	Match patterns with regard to case
mimetypes	mime.properties	MIME types for Web server
missingvalue	*UNKNOWN*	Used to replace a missing value
nonnumericerror	*ERROR*	Used to replace a value when you format something that isn't a number as a number
parity	0	Parity (0=none, 1=mark, 2=space, 3=odd, 4=even)
port	com1	Com port (case sensitive)
rules	rules.txt	Rule file
smtp	none	Name of SMTP (outgoing mail) server
stimulus	none	Name of stimulus file
stopbits	1	Number of stop bits (1, 2, or 15 for 1.5 stop bits)
tag	VAR	Name of replacement tag in templates

telnetline	true	If true, NetPorter adds a CRLF to the end of serial port lines received from Telnet
telnetport	none	Port for telnet server (default telnet port is 23)
telnetpw	password	Telnet access password
wwwport	none	Web server port (default Web server is on port 80)

Rule Command Reference

Rules are not case sensitive. Insert comments in a rule file by prefixing the entire line with a #. For file names, use / instead of \ to separate directories.

APPEND

Use this command after a BUILD to save the file at the end of an existing file. The single argument is a file name (which may include an absolute path starting with / or a path relative to the default directory).

BEGIN

The BEGIN command is similar to a MATCH command, but it only matches when NetPorter starts. You can use any normal commands (such as SET, BUILD, SEND, etc.) although there is no matching string so \$1, \$2, etc. in the SET statements will not work.

Here is a simple example that sends e-mail and initializes a variable:

```
BEGIN
SET V1=0
BUILD start-email.txt
SEND alw@al-williams.com
```

If you create a dummy variable in the BEGIN block you'll be able to examine its TIME attribute to determine when the system started.

BUILD

This rule specifies a template file that NetPorter will merge with the currently set variables to produce a file when using SAVE or APPEND. It also specifies a template for use with SEND. While the template is often a Web page, it can be any file format (such as a comma-delimited file for import into a spreadsheet).

MATCH

This is the first rule of a group. The argument specifies a regular expression to match (usually) the entire line. If the expression does not match the line, processing skips other rules to find the next MATCH command.

NEXT

If this command executes, processing of this line ends and the next line will continue processing at the start of the rules file. Without this command, it is possible to write more than one rule that will match a single line.

PREFIX

This command is similar to the prefix command for stimulus files and is for use before a STIM command.

PUBLISH

This command publishes the specified file to an FTP server. The first argument is the FTP server (use a dash if you want the default server). You can specify an optional FTP directory in quotes. The file name appears last. Here are some examples (the italicized text is for reference only):

PUBLISH ftp.myhost.com main.htm – *send main.htm to ftp.myhost.com*

PUBLISH – page.htm – *send page.htm to default server*

PUBLISH ftp.myhost.com "data" main.htm – *send main.htm to the data directory of the FTP server*

You can specify many default options in the properties file. However, you can override all of these by using the full syntax for the FTP server. For example:

```
ftpuser:mypassword@ftp.al-williams.com:21
```

This example specifies the user, the password, the host, and the port number. You can omit any part of the address except for the host name and the defaults will apply.

SAVE

The SAVE command follows a BUILD command and saves the expanded template to a file you specify. You can provide an absolute path (starting with a /) or a path relative to the default directory.

SEND

The send command follows a BUILD command and sends the expanded template to an e-mail address you specify. If you wish the e-mail to have a subject line, specify it with a SUBJECT command.

SENDHTML

This command is just like SEND, but it sends your e-mail as HTML instead of plain text.

SET

This command specifies a replacement pattern to transform the input string into a variable assignment. In the pattern, \$1 represents the match of the first left parenthesis of the MATCH command, \$2 represents the next match, and so on. You can include multiple SET commands for one MATCH command.

As an example, suppose that your input line looks like this:

```
333:222
```

You want to assign the first number to variable A and the second to variable B. You could write the following MATCH/SET lines:

```
MATCH ([0-9]+):([0-9]+)\s*$
```

```
SET A=$1
SET B=$2
```

Note that setting a variable sets it permanently (until you set it again). You don't have to use it only in the MATCH block that you set it in, but you can use it in any MATCH block.

STIM

This command sends a stimulus to the remote device as part of a MATCH block. You can use the same escapes that you can use as part of a STIM command in a stimulus file.

SUBJECT

This command precedes a SEND command and sets the subject for any subsequent SENDs.

Template Tag Reference

The default tag is VAR, but you can change that in the properties file using the tag parameter. The tag is not case sensitive.

`<VAR variable>`

This form of the tag simply substitutes the named variable's value. The variable can contain a number or a string.

`<VAR variable TIME>`

This form for the tag substitutes the time and date that the system acquired the named variable in a default format.

<VAR variable #format>

This tag displays variable as a number using the specified formatting. Within the format string the following characters are used (note the first # sign is not used):

# - non-zero digit	0 – digit (including 0)
. - decimal point (, in some locales)	, - comma or local grouping character
-- negative prefix	% - show value as a percentage
' - forces the next character to not have special meaning	

Any other character appears in the formatted value.

In addition, you can provide separate formats for negative and positive numbers by writing the positive format, including a semicolon, and then writing the negative format. For example:

##0.#;-##0.#

<VAR variable @format>

This takes the acquisition time and date of the named variable and formats it with the characters in the following table:

yy – two digit year	yyyy – four digit year
M – 1 or 2 digit month	MM – 2 digit month
MMM – Named month	zz, zzzz – Time zone
EE – Day of week (short)	EEEE – Day of week (long)
d – 1 or 2 digit day	dd – 2 digit day
h – 1 or 2 digit hour (1-12)	hh – 2 digit hour (1-12)
H – 1 or 2 digit hour (0-23)	HH – 2 digit hour (00-23)
mm – Minutes	ss – Seconds
SSS – milliseconds	a – AM or PM
D – Day of year (1, 2, or 3 digits)	DDD – Day of year (3 digits)

Stimulus File Reference

BEGIN

Use the **BEGIN** command to specify a string to send to the remote device when NetPorter starts. If you wish to include special characters for **BEGIN** or **SEND** you may use `\r` (carriage return), `\n` (new line), `\b` (backspace), `\t` (tab), `\s` (space), or `\\` (backslash).

SEND

This command takes two expressions, a number and a string. The number is the number of seconds that must elapse before sending the string to the remote device. For example:

```
SEND 5 ?\r\n
```

Note that you should plan to stagger your **SEND** commands so as not to confuse the remote device. For example:

```
SEND 5 A  
SEND 10 B
```

Will cause A and B to be sent to the device every 10 seconds (and just an A to be sent every other 5 second time period).

PREFIX

If you specify a **PREFIX** before a **SEND** or **BEGIN** command, NetPorter will put it ahead of the input from the device. This is useful when the device only outputs a number in response to a command.

About Regular Expressions

The patterns used by MATCH are regular expressions. This is a standard way to specify patterns used by languages like Perl, AWK, the Unix Shell, and in many other programs.

Each regular character in a regular expression must match a character in the input exactly. So if your pattern is DOG, then the input must contain DOG to match. By default, NetPorter does not respect case, so dog, Dog, or DoG would match.

Notice that the match does not have to be the only thing on the line. The line "my dog is here" matches the pattern "DOG" too. If you want to make sure it is the only thing on the line, you can use two special characters, ^ and \$. The pattern "^DOG\$" means, start at the beginning of the input, match DOG, and then only match if that is the end of the input.

Of course, you might want to allow spaces before or after the word. Then you could use the special character \s to represent any kind of space (blank, tab, etc.) and the * character to match 0 or more characters. So "\s*DOG\s*\$" would allow any number of spaces before or after DOG but no other characters. The period is a special character that matches any character.

What if you want to match a * (or other special character)? Then you need to put a backslash in front of it to tell NetPorter it isn't a special character. So "DOG\"*" matches the word DOG with an asterisk after it, while "DOG*" matches the word DO with 0 or more G's after it. The + sign requires one or more (instead of 0 or more) characters, so "DOG+" matches DOG or DOGG or even DOGGGGGGG. A ? makes the previous character optional. So "DOGS? AND CATS?" matches "DOGS AND CAT" or "DOG AND CAT" or several other permutations.

Parenthesis have two meanings in the regular expression. First, it groups characters together. So you could write "(ab)?c" to match "abc" or just

Example

In the example directory, you'll find a simple example using a Basic Stamp II as the target instrument. The Stamp responds to a ? command to send two data items, an increasing count, and a random value. The separator is a colon and a space. The Stamp also accepts an @ command to reset the count to 1.

NetPorter reads the data and appends the data to a log file (log.htm). It also updates history.htm which includes log.htm and some other data.

When NetPorter starts, it resets the device and the log file using statements in the BEGIN rule. It also publishes a Web page with the current start time. When the rules detect the 200th entry, NetPorter clears the log file, resets the count, and sends you an e-mail. You can examine the history and log files by browsing <http://localhost:808/history.htm> (substitute the correct computer name if you are on a network).

Files:

test.bs2 – The example Basic Stamp II program.

files – Directory containing saved data files.

templates – Directory of templates.

templates/emptylog.template – Erases the log.

templates/log.template – Template for a line of the log file.

templates/history.template – Template for history.htm.

templates/main.template – Template for snapshot page (main.htm).

templates/uptime.template – Template for "uptime" page.

files/reset.htm – Form to reset the device.

files/error404.htm – Custom error page.

You must change the NetPorter.properties file to reflect your local file locations, serial port number, etc. before the example will work.

License

This is a legal agreement between the end user ("You") and AI Williams Consulting., its affiliates and subsidiaries (collectively "AWC"). This Agreement is part of a package (the "Package") that also includes, as applicable, executable files that you may download, a disc, or a CD-ROM (collectively referred to herein as the "Software") and certain written materials (the "Documentation"). Any patch, update, upgrade, modification or other enhancement provided by AWC with respect to the Software or the Documentation shall be included within the meanings of those terms, for the purposes of this Agreement, except to the extent expressly provided below.

BY DOWNLOADING OR INSTALLING THE SOFTWARE, YOU ACKNOWLEDGE THAT YOU HAVE READ ALL OF THE TERMS AND CONDITIONS OF THIS AGREEMENT, UNDERSTAND THEM, AND AGREE TO BE BOUND BY THEM. YOU UNDERSTAND THAT, IF YOU PURCHASED THE PACKAGE FROM AN AUTHORIZED RESELLER OF AWC, THAT RESELLER IS NOT AWC'S AGENT AND IS NOT AUTHORIZED TO MAKE ANY REPRESENTATIONS, CONDITIONS OR WARRANTIES, STATUTORY OR OTHERWISE, ON AWC'S BEHALF NOR TO VARY ANY OF THE TERMS OR CONDITIONS OF THIS AGREEMENT.

If You do not agree to the terms of this Agreement, do not download or install the Software and promptly return the entire Package to the place You obtained it for a full refund. If you should have any difficulty in obtaining such refund, please contact AWC at 281-334-4341.

LIMITED LICENSE: You are entitled to download or install, and operate this Software solely for your own use, but may not sell or transfer reproductions of the Software or Documentation to other parties in any way. You may download or install, and operate one copy of the Software on a single terminal connected to a single computer. You may not network the Software or otherwise use it on more than one computer or computer terminal at the same time.

OWNERSHIP; COPYRIGHT: Title to the Software and the Documentation, and patents, copyrights and all other property rights applicable thereto, shall at all times remain solely and exclusively with AWC and its licensors, and You shall not take any action inconsistent with such title. The Software and the Documentation are protected by United States, Canadian and other applicable laws and by international treaty provisions. Any rights not expressly granted herein are reserved to AWC and its licensors.

OTHER RESTRICTIONS: You may not cause or permit the disclosure, copying, renting, licensing, sublicensing, leasing, dissemination or other distribution of the Software or the Documentation by any means or in any form, without the prior written consent of AWC. You may not modify, enhance, supplement, create derivative work from, adapt, translate, reverse engineer, decompile, disassemble or otherwise reduce the Software to human readable form.

LIMITED WARRANTY: AWC warrants for a period of ninety (90) days following original retail purchase of this copy of the Software that the Software is free from substantial errors or defects that will materially interfere with the operation of the Software as described in the Documentation. This limited warranty: (i) applies to the initial purchaser only and may be acted upon only by the initial purchaser; and (ii) does not apply to any patch, update, upgrade, modification, or other enhancement provided by AWC with respect to the Software or the Documentation or to any bonus game provided by AWC at no extra charge as part of the Package, which are provided on an AS IS BASIS ONLY. EXCEPT AS STATED ABOVE, AWC AND ITS LICENSORS MAKE NO OTHER WARRANTY OR CONDITION, EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, REGARDING THIS SOFTWARE. THE IMPLIED WARRANTY THAT THE SOFTWARE IS FIT FOR A PARTICULAR PURPOSE AND THE IMPLIED WARRANTY OF MERCHANTABILITY SHALL BOTH BE LIMITED TO THE NINETY (90) DAY DURATION OF THIS LIMITED EXPRESS WARRANTY. THESE AND ANY OTHER IMPLIED WARRANTIES OR CONDITIONS, STATUTORY OR OTHERWISE, ARE OTHERWISE EXPRESSLY AND SPECIFICALLY DISCLAIMED. Some jurisdictions do not allow limitations on how long an implied warranty or condition lasts, so the above limitation may not apply to You. This limited warranty gives You specific legal rights, and you may also have other rights which vary from jurisdiction to jurisdiction.

If you believe you have found any such error or defect in the Software during the warranty period, contact AWC as described in the Documentation. If you have a problem resulting from a manufacturing defect in the Software, AWC's and its licensors' entire liability and Your exclusive remedy for breach of this limited warranty shall be the replacement of the Software, within a reasonable period of time and without charge, with a corrected version of the Software. Some jurisdictions do not allow the exclusion or limitation of relief, incidental or consequential damages, so the above limitation or exclusion may not apply to You.

LIMITATION OF LIABILITY: AWC AND ITS LICENSORS SHALL NOT BE LIABLE FOR SPECIAL, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY OR OTHER INDIRECT DAMAGES, EVEN IF AWC OR ITS LICENSORS ARE ADVISED OF OR ARE AWARE OF THE POSSIBILITY OF SUCH DAMAGES. IN NO EVENT SHALL AWC'S AND ITS LICENSORS' AGGREGATE LIABILITY EXCEED THE PURCHASE PRICE OF THIS PACKAGE. Some jurisdictions do not allow the exclusion or limitation of special, incidental, consequential, indirect or exemplary damages, or the limitation of liability to specified amounts, so the above limitation or exclusion may not apply to You.

GENERAL: This Agreement constitutes the entire understanding between AWC and You with respect to subject matter hereof. Any change to this Agreement must be in writing, signed by AWC and You. Terms and conditions as set forth in any purchase order which differ from, conflict with, or are not included in this Agreement, shall not become part of this Agreement unless specifically accepted by AWC in writing. You shall be responsible for and shall pay, and shall reimburse AWC on request if AWC is required to pay, any sales, use, value added (VAT), consumption or other tax (excluding any tax that is based on AWC's net income), assessment, duty, tariff, or other fee or charge of any kind or nature that is levied or imposed by any governmental authority on the Package.

EXPORT AND IMPORT COMPLIANCE: In the event You export the Software or the Documentation from the country in which You first received it, You assume the responsibility for compliance with all applicable export and re-export regulations, as the case may be.

GOVERNING LAW: This Agreement shall be governed by, and any arbitration hereunder shall apply, the laws of the Texas, U.S.A. This software uses Jakarta-Oro developed by the Apache Software Foundation (www.apache.org). Jakarta-Oro is Copyright © 2000 The Apache Software Foundation. All rights reserved. NetPorter is Copyright © 2000, 2001 by AWC. All rights reserved.