

## **AnnexWatch Version 2.0**

*Annex Watch - Copyright 1996-1999 G & R Data Group, Inc.  
<http://www.grdata.com/solutions/AnnexWatch/>*

Revision D, December 1999

*G&R Data Group, Inc.  
Software Distribution  
P.O. Box 428  
Woodstock, GA 30188*



*©Copyright 1997-1999 G & R Data Group, Inc.*

## **DISCLAIMER**

This software is designed to provide information about Nortel Networks, formerly Bay Networks, formerly Xylogics Annex(R) logfiles. This product is known by "Annex," "Remote Annex," and other names. Every effort has been made to make this program complete and as accurate as possible; however, no warranty or fitness is implied. Information provided to the end user is on an 'as-is' basis. The Authors, G&R Data Group, and its distributors shall have neither liability nor responsibility to any person or entity with respect to any loss or damages in connection with or arising from the information reported by this product. Nortel Networks, Bay Networks, Xylogics and their products are copyright their respective companies.

## CONTENTS

<b>1. INTRODUCTION</b>	<b>4</b>
<b>2. INSTALLATION</b>	<b>5</b>
<b>2.1. Licenses</b>	<b>5</b>
<b>2.2. Support</b>	<b>6</b>
<b>3. COMMAND LINE OPTIONS</b>	<b>7</b>
<b>4. THE ANATOMY OF A MAIN TABLE</b>	<b>9</b>
<b>4.1. DETAILS TO KEEP IN MIND</b>	<b>10</b>
<b>5. FREQUENTLY ASKED QUESTIONS</b>	<b>12</b>
<b>6. EXAMPLES</b>	<b>13</b>
<b>7. EXTENSIONS: USING THE WORLD WIDE WEB</b>	<b>17</b>
<b>8. EXTENSIONS: CRONJOBS</b>	<b>24</b>

# 1. INTRODUCTION

AnnexWatch is a system administration tool. The reader is expected to have a general knowledge of UNIX system administration, shell programming, and a conceptual understanding of Annex hardware. AnnexWatch includes methods of utilizing the World Wide Web (WWW), to use these features the system administrator should be familiar with web browsers or web servers.

AnnexWatch has many options designed to help the system administrator extract useful information from an Annex *acp\_logfile*. Some of the options influence how the statistics tables look, while others influence the scope over which the statistics are taken. AnnexWatch includes a few example shell programs to aid in automation of periodic tasks and event notification.

Options that influence the look of a table include table ordering and whether the table should be reported in terms of users or portnumbers. You may order a table by username, portnumber, total accumulation times, accumulation times for connection style (cli, ppp, slip), or total number of connections for a connection style (cli, ppp, slip). The importance of knowing port usage verses individual user usage's will dictate your basic tabulation style.

To aid in the development of post processing programs (data plotters, etc.) AnnexWatch offers a raw data output mode.

Options that influence the scope over which information is gathered allow you to expand or narrow the display of information. You may use these options to restrict the time period over which statistics are gathered, who they are gathered from, and which groups of ports are included.

Overall, AnnexWatch generates two types of tables: "quick tables" and "main tables". These can be thought of as general tables and specific tables, respectively. Quick tables yield very general information, for example, a quick table can answer questions such as: "Who has logged on?" What ports have been used? How many times has the Annex been rebooted? A main table may yield more specific information and can answer questions such as: How long has user X been on? Which ports has user X logged in on? Who has logged in over the last week? How much has port Y been used over the last month? How many different users have ever logged into port Y?

Through effective choice of tables, table ordering, and scope you may explore the information contained within your logfile. Frequently asked questions and the options that provide answers are included in the examples section.

Time line output is the second form of output type generated by AnnexWatch. Currently the user may generate time lines of average daily and weekly loads. Look for future developments in this area.

## 2. INSTALLATION

1) Decide where to set things up (/usr/local/AnnexWatch is a common choice). Log on as root and create an AnnexWatch directory.

```
% su -  
Password: *****  
# cd /usr/local  
# mkdir AnnexWatch  
# cd AnnexWatch
```

2) Copy the AnnexWatch shar file from the media that was shipped to you into this directory. If you received tape from us then you will need to use the tar command to extract the archive from tape. The tar command will look something like this:

```
# tar -xvf /dev/tape
```

If you have not received the software you may download it from our web server by using your favorite browser. Look under <http://www.grdata.com/products>. Alternately, you can download it from our anonymous ftp server <ftp.grdata.com>. You may also use this site to download the latest version up to the next major revision. Be sure to make all ftp transfers in binary.

3) Uncompress and untar the file in the /usr/local/bin/AnnexWatch directory:

```
# pwd  
/usr/local/bin/AnnexWatch  
# uncompress AnnexWatch.tar.Z  
# tar -xvf AnnexWatch.tar
```

4) It may be useful to put a link in /usr/local/AnnexWatch to your systems active logfile:

```
# ln -s /usr/annex/acp_logfile /usr/local/AnnexWatch/acp_logfile
```

5) Logout as root, AnnexWatch is almost setup, proceed to the next section to setup your license.

```
# exit  
% cd /usr/local/AnnexWatch
```

### 2.1.Licenses

1) Like all valuable software, AnnexWatch is bound to a single CPU by a license. If you filled in the pre-registration when ordering your copy of this product you should find your unique license token shipped with this manual. If you downloaded this product from our software server you may not yet have your license token. Visit our license server on the web to receive your token. Promotional license tokens are available on the web for free. Permanent tokens are available for purchase.

<http://www.grdata.com/solutions/>

2) Create a license file using any editor copy your token onto the first line of the file. Be sure to copy your token exactly. If possible, just cut and paste your token directly so as to avoid any typographical errors. Your license file may be called anything and placed anywhere, however we recommend calling it AnnexWatch.license and placing with your AnnexWatch distribution:

```
% cat > /usr/local/AnnexWatch/AnnexWatch.license
FAB4CAB5CA4XBCFA5BC6ABC8ABCAJ7B7CA
^D
%
```

3) To let AnnexWatch know where the license file is kept you will have to create an environment variable called AXW. The value of this variable points to the location of the license. If you use the c-shell then the syntax is:

```
% setenv AXW /usr/local/AnnexWatch/AnnexWatch.license
```

if you like to use the k-shell or borne shell then the syntax is:

```
% AXW=/usr/local/AnnexWatch/AnnexWatch.license
% export AXW
```

It is cool to place these commands in your .cshrc, .kshrc, or .profile files respectively so you never have to worry about licenses again.

5) AnnexWatch comes with a cgi-binary web driver, if this is of interest installation instructions are available under the section 'Extensions: Using the World Wide Web'.

## **2.2. Support**

We are interested in making G&R Data Group products better for you. Feel free to send suggestions, ideas, bugs, correspondence etc. regarding AnnexWatch to:

*support@grdata.com*

Or by regular mail to:

*G&R Data  
Software Distribution  
P.O. Box 428  
Woodstock, GA 30188*

Visit the AnnexWatch Home Page for demos, tips, tricks, or FAQ's when available:

*<http://www.grdata.com/solutions/AnnexWatch/>*

Please include your name, email, AnnexWatch Version Number, a screen dump (script file) illustrating your question, and a copy of your acp\_logfile, if possible, with any bug reports.

### 3. COMMAND LINE OPTIONS

*Synopsis:*

**AnnexWatch** [logfile] [-option [keyword] ... ]

*General Options:*

<b>-quiet</b>	Enter quiet mode, do not print warnings or info messages.
<b>-help</b>	Print a short help message on options.
<b>-rawoutput</b>	Produce unformatted output.
<b>-http</b>	Produce http formatted output.

*Options that effect the quick tables:*

<b>-list users</b>	Form a quick table of users encountered in the logfile
<b>-list ports</b>	Form a quick table of portnumbers encountered in the logfile
<b>-summary</b>	Print a quick summary table
<b>-typical day</b>	Usage over a typical day
<b>-typical week</b>	Usage over a typical week
<b>-binsize #</b>	Usage report binsize in minutes (default = 60 minutes)

*Options that effect the appearance of the main table:*

<b>-tableby username</b>	Form the main table by username (Default)
<b>-tableby port</b>	Form the main table by portnumber
<b>-average</b>	Show average login time per session rather than accumulated times.
<b>-units minutes hours days</b>	Show times using the specified unit (default is minutes).
<b>-sortby username</b>	Sort the main table by username
<b>-sortby port</b>	Sort the main table by port number
<b>-sortby totaltime</b>	Sort the main table by totaltime (Default)
<b>-sortby pptime</b>	Sort the main table by pptime
<b>-sortby sliptime</b>	Sort the main table by sliptime
<b>-sortby clitime</b>	Sort the main table by clitime
<b>-sortby pppconnects</b>	Sort the main table by pppconnects
<b>-sortby slipconnects</b>	Sort the main table by slipconnects
<b>-sortby clicconnects</b>	Sort the main table by clicconnects
<b>-nozeros</b>	Suppress reporting zero statistics

*Options that effect the scope of the main table:*

<b>-all</b>	report activity for all ports and all users
<b>-all users</b>	report activity for all users and no ports
<b>-xuser username(s)</b>	exclude reporting activity for user username
<b>-iuser username(s)</b>	include reporting activity for user username
<b>-all ports</b>	report activity for all ports and no users
<b>-xport # [# ...]</b>	exclude reporting activity for portnumber(s)
<b>-iport # [# ...]</b>	include reporting activity for portnumber(s)

The previous seven options control the 'space' scope to use. You may either specify each user, all users, each port, all ports, all ports and users to be in scope. One or more of these is required to produce a main table. There is no You may list multiple users and ports after the respective option.

<b>-from start datecode [timecode]</b>	report activity only after date [time]
<b>-to end datecode [timecode]</b>	report activity only up to date [time]

The *from* and *to* options control the 'time' scope to use. You may either specify the special keyword OR a date and optional time Date codes have the form *MM/DD/YY*, while optional time codes have the form *HH:MM:SS* (using a 24-hour clock). If no time code is given AnnexWatch will assume the time code to be 00:00:01 - that is to one second after midnight. The special keywords *start* and *end* specify that the time found at the start and end of the file should be used. The default is to include the entire logfile, -from start -to end.

*Debug Info:*

<b>-v</b>	toggle verbose debug output ( <i>NOT</i> recommended).
-----------	--

## 4. THE ANATOMY OF A MAIN TABLE

Here is a typical user table (i.e. a table based on usernames):

USER TABLE  
=====

```
(All times in minutes)
Username  Ports      PPP Time      SLIP Time      CLI Time      Total Time
~~~~~
   spot    3      3243.43   69      0.00    0      98.75   79      3342.18
   ernie    2         0.92    1     2348.48  54     121.30  56     2470.70
  littlej   2         0.00    0     853.65  49     1294.92 140     2148.57
(lines omitted to save space)
   charlie  1         0.00    0      98.08    4       0.27   4       98.35
   betty    2         0.00    0      92.80    9       0.47   9       93.27
   ronald   1         0.00    0      10.83    1       0.02   1       10.85
~~~~~
Totals          5335.23  154     10724.35  568     4806.13 1040     20865.72
```

Column 1 is labeled Username and contains the username as encountered in the logfile. Column 2 is labeled Ports and shows the number of different ports this user has been on. The next column shows PPP time and the number of PPP connections that this user accumulated. The totals show the total PPP time accumulated and the total number of connections. Similar statistics are gathered for SLIP and CLI. The last column shows the accumulated PPP, SLIP and CLI time. Accumulated connections has no meaning as a CLI connection will often spawn a PPP or SLIP connection. Reading this table one may see that user 'spot' logged in CLI 79 times and in all but 10 of those connections he spawned off a PPP connection. The user 'littlej' however uses CLI connections and only uses SLIP some of the time. He is probably telnetting out and using text connections.

Here is a typical port table (i.e. a table based on port number):

PORT TABLE  
=====

```
(All times in minutes)
Port      Users      PPP Time      SLIP Time      CLI Time      Total Time
~~~~~
   030     18      3153.52   84     5382.58  391     3610.12  703     12146.22
   033     8       2056.87   65     3975.25  128     121.98   207     6154.10
   031     17      124.85    5     1315.72  45     1066.75  124     2507.32
   034     3         0.00    0       50.02    3         0.15    4         50.17
   032     2         0.00    0         0.78    1          7.13    2          7.92
~~~~~
Totals          5335.23  154     10724.35  568     4806.13 1040     20865.72
```

Column 1 is labeled Port and contains the port number as encountered in the logfile. Column 2 is labeled Users and shows the number of different users that have been on this port. The next column shows PPP time and the number of PPP connections that this port has accumulated. The totals show the total PPP time accumulated and the total number of connections of this type to this port. Similar statistics are gathered for SLIP and CLI. The last column shows the accumulated PPP, SLIP and CLI time. Accumulated connections has no meaning as a CLI connection will often spawn a PPP or SLIP connection. Reading this table one may see that port 32 logged few connections while the modem on port 30 is being used heavily.

## 4.1.DETAILS TO KEEP IN MIND

Try it. If you want to see something added to the functionality of AnnexWatch left us know!

AnnexWatch assumes that there are 640 or less unique users, and there are 128 or less unique port numbers. Only the first 640 unique users encountered in a log file will be processed. The 641<sup>st</sup> user and beyond will be ignored. This restriction will be lifted in the next version.

Days start at 1 second past midnight.

Midday is 12 pm, Midnight is 0 am.

All typical day or week statistics are gathered internally at minute resolutions. Binsize only specifies the resolution for display. Tables and plots show statistics bin centered. For example a table or plot with 60 minute bins will plot statistics gathered from 2pm to 3pm at 2:30pm.

If not specified, on the command line, a file called `acp_logfile` will be used.

Some large logfiles may take a minute or so to draw statistics from depending on your platform.

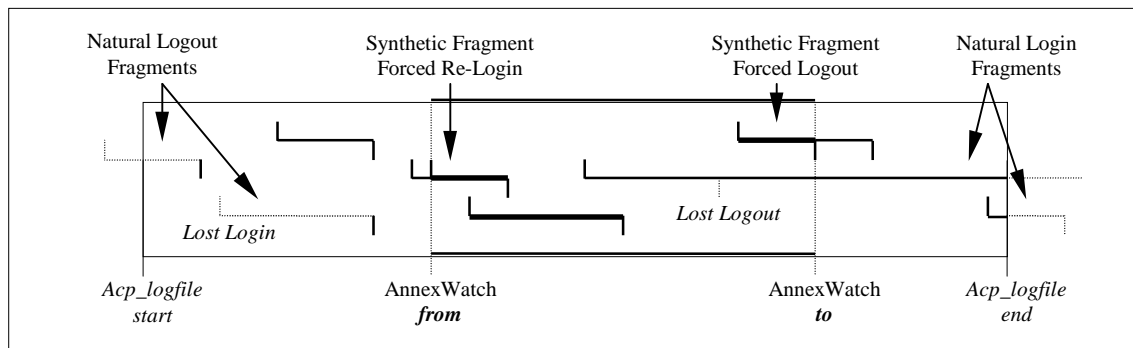


Figure 1: Fragment Definitions. Only those activities with bold horizontal lines are recorded for AnnexWatch statistics. Lost login or logout markers create lost time lines.

**Natural Fragments:** Natural Fragments are defined as a logout without a recorded login *or* a login without a located logout due to the natural behavior of an Annex box. Natural Fragments can and *will* occur under normal operating conditions. A typical natural fragment occurs when a logfile is truncated; that is, if you delete or move your `acp_logfile` while users are logged into your Annex. In this instance there will occur logout records without matching login records in the beginning of the new logfile. Similarly there will appear login records without matching logout records in the original logfile. At other times, perhaps due to very heavy network loading, or due to abnormally interrupted user sessions a login or logout record may not be recorded to the `acp_logfile`. AnnexWatch will ignore natural fragmentary records such as these but they will trigger warning messages when encountered. Figure 1 illustrates two natural fragments due to `acp_logfile` truncation, and two natural fragments due to lost records. Note that there really isn't anyway for AnnexWatch to distinguish between these two types of natural fragments. Any natural fragment will reflect some 'leaked' time. The magnitude of this leaked time cannot be estimated. For example, if user 'lucky' logs in before the `acp_logfile` is started and logs out after it ends there is no indication in the `acp_logfile` that the user was ever there (other than perhaps that one port never seems to have been used).

**Synthetic Fragments:** Synthetic Fragments are fragments similar to Natural Fragments but are created purposely by AnnexWatch. Synthetic fragments are created when the time period of interest is less than the time period of the entire logfile. In this case if AnnexWatch determines that usage crosses either the start or stop times it will create a synthetic fragment. Any logins that logout after the user specified time period of interest will be forced to “logout” at the end of the time period. This occurs *if and only if* a natural logout for that user can be found in the remainder of the logfile. If no natural logout is found then the record is considered a natural fragment and is ignored. Any user that is still logged in when the start of the time period is encountered will be forced to “relogin” at the start of the user specified period of interest. These are not considered natural fragmentary records. Informational messages are raised when synthetic fragments are encountered.

The following illustration shows where lost acp\_logfile records can start creating inaccuracies in the statistics that AnnexWatch gathers from the acp\_logfile. Because there is no way for you, me, or AnnexWatch to know about information not recorded to the acp\_logfile, the administrator must realize that coincidences may arise (very rarely) that may affect the final statistics.

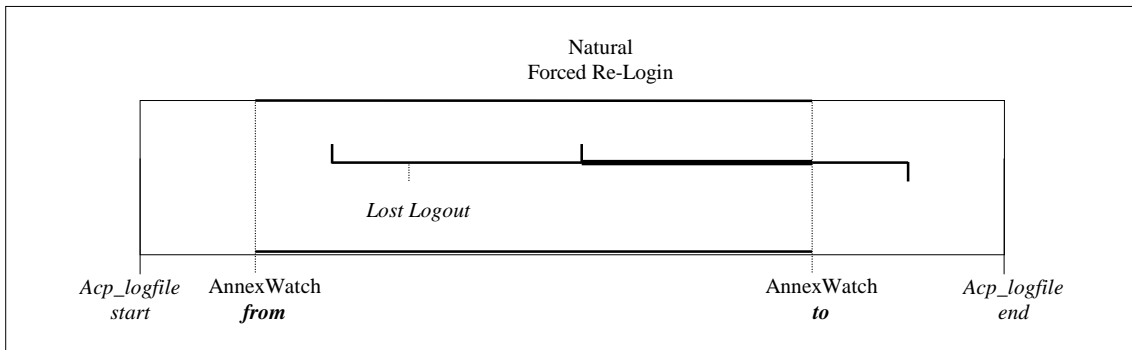


Figure 2: How rare coincidences with natural fragments may cause errors in statistics.

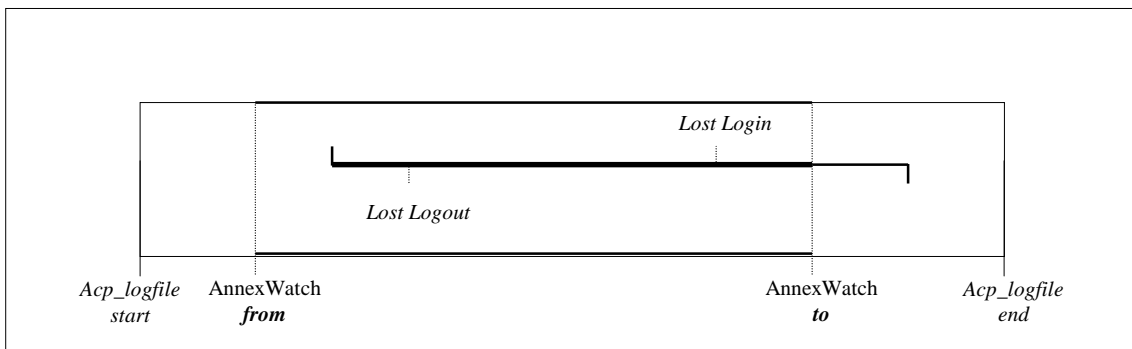


Figure 3: How rare coincidences with natural fragments may cause errors in statistics.

## 5. FREQUENTLY ASKED QUESTIONS

It is better to learn by doing; here are some starting points:

Q: Which users have entries in the logfile?

```
% AnnexWatch -list users
```

Q: Which ports have seen activity?

```
% AnnexWatch -list ports
```

Q: Which ports have seen activity, please exclude the WARNINGS?

```
% AnnexWatch -list ports -quiet
```

Q: How many users and ports have seen use?

```
% AnnexWatch -summary -quiet
```

Q: Can you show me all the details for all users for the entire logfile?

```
% AnnexWatch -all -quiet
```

Q: Can you do that again but order by username?

```
% AnnexWatch -all -sortby username -quiet
```

Q: How about by ppp time?

```
% AnnexWatch -all -sortby pptime -quiet
```

Q: What is the average connect time per person, ordered by average pptime?

```
% AnnexWatch -all -average -sortby pptime -quiet
```

Q: Can you show me all the details for all ports for the entire logfile?

```
% AnnexWatch -all -tableby port -quiet
```

Q: Can you show me all the details for all ports for June?

```
% AnnexWatch -all -tableby port -quiet -from 6/1/95 -to 7/1/95
```

Q: User 'joe' is annoying, how can I exclude him from the statistics?

```
% AnnexWatch -all -xuser joe
```

Q: I want to know which ports 'joe' has been on, and the amount of time?

```
% AnnexWatch -all ports -iuser joe -tableby port
```

Q: I want to know who has used port 33?

```
% AnnexWatch -all users -iport 33
```

Q: Can you show me who used port 30 on June 1st?

```
% AnnexWatch -all users -iport 30 -from 6/1/95 -to 6/2/95
```

Q: Can you show me the port activity for July?

```
% AnnexWatch -all -from 7/1/95 -to 8/1/95 -tableby port
```

Q: Can you show me if 'joe' used port 30 on June 1st?

```
% AnnexWatch -iuser joe -iport 30 -from 6/1/95 -to 6/2/95
```

Q: Can you do my dishes?

```
% echo no
```

## 6. EXAMPLES

*All examples use the example.acp\_logfile provided with the distribution.*

### **Q: Which ports have seen activity?**

```
% AnnexWatch -list ports
```

```
WARNING: Fragment record for 'mickey' - ignored
WARNING: Fragment record for 'mickey' - ignored
WARNING: Fragment record for 'mickey' - ignored
WARNING: Fragment record for 'mickey' - ignored
WARNING: Fragment record for 'barney' - ignored
WARNING: Fragment record for 'barney' - ignored
WARNING: Fragment record for 'olive' - ignored
File Spanning      03:03:57 pm, May 27 1995 to 11:51:32 pm, Aug 23 1995.
Statistics Spanning 03:03:57 pm, May 27 1995 to 11:51:32 pm, Aug 23 1995.
```

```
Ports for Entire File
```

```
=====
30
31
32
33
34
```

### **Q: Which ports have seen activity, please exclude the WARNINGS?**

```
% AnnexWatch -list ports -quiet
```

```
File Spanning      03:03:57 pm, May 27 1995 to 11:51:32 pm, Aug 23 1995.
Statistics Spanning 03:03:57 pm, May 27 1995 to 11:51:32 pm, Aug 23 1995.
```

```
Ports for Entire File
```

```
=====
30
31
32
33
34
```

### **Q: How many users and ports have seen use?**

```
% AnnexWatch -summary -quiet
```

```
File Spanning      03:03:57 pm, May 27 1995 to 11:51:32 pm, Aug 23 1995.
Statistics Spanning 03:03:57 pm, May 27 1995 to 11:51:32 pm, Aug 23 1995.
```

```
Summary for Entire File
```

```
=====
```

```

Period Length:          88 days, 527.6 minutes
Total Number of Users: 23
Total Number of Ports Used: 5

```

Summary for Period

=====

```

Period Length:          88 days, 527.6 minutes
Total Number of Rejects: 227
Total Number of Timeouts: 0
Total Number of Boots: 3
Total Number of Fragments: 7

```

**Q: Can you show me all the details for all users for the entire logfile?**

```
% AnnexWatch -all -quiet
```

```

File Spanning      03:03:57 pm, May 27 1995 to 11:51:32 pm, Aug 23 1995.
Statistics Spanning 03:03:57 pm, May 27 1995 to 11:51:32 pm, Aug 23 1995.

```

USER TABLE

=====

(All times in minutes)

Username	Ports	PPP Time		SLIP Time		CLI Time		Total Time
~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~
spot	3	3243.43	69	0.00	0	98.75	79	3342.18
ernie	2	0.92	1	2348.48	54	121.30	56	2470.70
littlej	2	0.00	0	853.65	49	1294.92	140	2148.57
barney	3	0.00	0	1656.33	36	7.97	40	1664.30
pluto	2	0.00	0	0.00	0	1485.63	53	1485.63
fred	2	0.00	0	1446.20	37	34.92	37	1481.12
leo	1	1369.85	48	0.00	0	8.68	48	1378.53
minnie	3	0.00	0	1363.30	159	6.20	163	1369.50
zeus	3	0.00	0	862.63	47	1.82	47	864.45
olive	2	0.28	1	770.13	57	6.63	61	777.05
mickey	3	0.00	0	372.83	43	262.10	67	634.93
jane	2	0.00	0	274.57	9	353.53	26	628.10
jupiter	2	0.00	0	0.00	0	478.20	42	478.20
bert	2	0.00	0	0.00	0	473.82	36	473.82
junior	2	376.10	12	0.00	0	42.50	20	418.60
goofy	4	132.93	8	90.65	7	104.92	35	328.50
larry	2	0.00	0	231.45	28	0.75	28	232.20
dizzy	2	211.72	15	0.00	0	6.28	18	218.00
norman	1	0.00	0	134.88	23	16.08	24	150.97
newton	1	0.00	0	117.52	5	0.38	6	117.90
charlie	1	0.00	0	98.08	4	0.27	4	98.35
betty	2	0.00	0	92.80	9	0.47	9	93.27
ronald	1	0.00	0	10.83	1	0.02	1	10.85
~~~~~								
Totals		5335.23	154	10724.35	568	4806.13	1040	20865.72

**Q: Can you show me all the details for all ports for the entire logfile?**

```
% AnnexWatch -all -tableby port -quiet
```

```

File Spanning      03:03:57 pm, May 27 1995 to 11:51:32 pm, Aug 23 1995.
Statistics Spanning 03:03:57 pm, May 27 1995 to 11:51:32 pm, Aug 23 1995.

```

PORT TABLE

=====

(All times in minutes)

Port	Users	PPP Time	SLIP Time	CLI Time	Total Time
030	18	3153.52 84	5382.58 391	3610.12 703	12146.22
033	8	2056.87 65	3975.25 128	121.98 207	6154.10
031	17	124.85 5	1315.72 45	1066.75 124	2507.32
034	3	0.00 0	50.02 3	0.15 4	50.17
032	2	0.00 0	0.78 1	7.13 2	7.92
Totals		5335.23 154	10724.35 568	4806.13 1040	20865.72

**Q: Can you show me all the details for all ports for June?**

*% AnnexWatch -all -tableby port -quiet -from 6/1/95 -to 7/1/95*

File Spawning 03:03:57 pm, May 27 1995 to 11:51:32 pm, Aug 23 1995.  
Statistics Spawning 00:00:01 am, Jun 01 1995 to 00:00:01 am, Jul 01 1995.

PORT TABLE  
=====

(All times in minutes)

Port	Users	PPP Time	SLIP Time	CLI Time	Total Time
030	9	1286.72 35	1503.55 89	5086.50 153	7876.77
033	5	115.62 6	1383.80 34	5.43 42	1504.85
031	8	30.33 1	22.30 3	216.95 26	269.58
034	1	0.00 0	3.70 1	0.02 1	3.72
032	0	0.00 0	0.00 0	0.00 0	0.00
Totals		1432.67 42	2913.35 127	5308.90 222	9654.92

**Q: I see from the previous table that 5 users have been on port 33 in June, can you identify who they are?**

*% AnnexWatch -all users -iport 33 -quiet -from 6/1/95 -to 7/1/95*

File Spawning 03:03:57 pm, May 27 1995 to 11:51:32 pm, Aug 23 1995.  
Statistics Spawning 03:03:57 pm, Jun 01 1995 to 11:51:32 pm, Jul 01 1995.

USER TABLE  
=====

(All times in minutes)

Username	Ports	PPP Time	SLIP Time	CLI Time	Total Time
barney	1	0.00 0	1285.47 20	0.45 20	1285.92
goofy	1	115.62 6	0.00 0	0.58 6	116.20
fred	1	0.00 0	58.62 5	0.13 5	58.75
mickey	1	0.00 0	35.87 8	4.23 10	40.10
zeus	1	0.00 0	3.85 1	0.03 1	3.88
olive	0	0.00 0	0.00 0	0.00 0	0.00
leo	0	0.00 0	0.00 0	0.00 0	0.00
ronald	0	0.00 0	0.00 0	0.00 0	0.00
junior	0	0.00 0	0.00 0	0.00 0	0.00
jane	0	0.00 0	0.00 0	0.00 0	0.00
pluto	0	0.00 0	0.00 0	0.00 0	0.00
newton	0	0.00 0	0.00 0	0.00 0	0.00
charlie	0	0.00 0	0.00 0	0.00 0	0.00

betty	0	0.00	0	0.00	0	0.00	0	0.00
jupiter	0	0.00	0	0.00	0	0.00	0	0.00
bert	0	0.00	0	0.00	0	0.00	0	0.00
minnie	0	0.00	0	0.00	0	0.00	0	0.00
spot	0	0.00	0	0.00	0	0.00	0	0.00
ernie	0	0.00	0	0.00	0	0.00	0	0.00
larry	0	0.00	0	0.00	0	0.00	0	0.00
norman	0	0.00	0	0.00	0	0.00	0	0.00
littlej	0	0.00	0	0.00	0	0.00	0	0.00
dizzy	0	0.00	0	0.00	0	0.00	0	0.00
~~~~~								
Totals		115.62	6	1383.80	34	5.43	42	1504.85

**Q: That wasn't quite what I meant, could you clean up the table so users with zero time for this period are not included in the table? Show me the 5 users that have been on port 33 in June.**

% AnnexWatch -all users -iport 33 -quiet -from 6/1/95 -to 7/1/95 -nozeros

File Spanning 03:03:57 pm, May 27 1995 to 11:51:32 pm, Aug 23 1995.  
 Statistics Spanning 03:03:57 pm, Jun 01 1995 to 11:51:32 pm, Jul 01 1995.

USER TABLE  
 =====

(All times in minutes)

Username	Ports	PPP Time	SLIP Time	CLI Time	Total Time
~~~~~	~~~~~	~~~~~	~~~~~	~~~~~	~~~~~
barney	1	0.00 0	1285.47 20	0.45 20	1285.92
goofy	1	115.62 6	0.00 0	0.58 6	116.20
fred	1	0.00 0	58.62 5	0.13 5	58.75
mickey	1	0.00 0	35.87 8	4.23 10	40.10
zeus	1	0.00 0	3.85 1	0.03 1	3.88
~~~~~					
Totals		115.62 6	1383.80 34	5.43 42	1504.85

## 7. EXTENSIONS: USING THE WORLD WIDE WEB

The web is a very effective tool for accessing and distributing information and the latest web browsers offer clean and efficient methods for presenting data. AnnexWatch will always run from the UNIX command line, however, html and cgi-binary extensions have been included. If you have a WWW server you may easily integrate AnnexWatch into a web accessible tool. If you do not wish to have your web server drive AnnexWatch you may still use any web browser to display AnnexWatch output. Since the second option requires no further setup, it is discussed first:

### Viewing AnnexWatch Output using a Web Browser

AnnexWatch provides a command line option to output html tags. The option is `-http`. All you need to do is pipe the output into a file and then open that local file using your favorite browser:

```
% AnnexWatch -all -quiet -sortby username -http > loadme.html
```

### Driving AnnexWatch using a Web Server

If you have a web server you can drop a few files into your cgi-bin directory and be on your way to some super slick effects. In the distribution you will find an executable called 'AnnexWatch.webdriver' and a small html file called 'AnnexWatch.html'. Assuming a common web server structure, you may use the following steps to set things up:

```
% su -  
Password: *****  
# cd /usr/local/AnnexWatch  
# mkdir ~www/cgi-bin/AnnexWatch  
# cp AnnexWatch.webdriver ~www/cgi-bin/AnnexWatch  
# chown -R www ~www/cgi-bin/AnnexWatch  
# cp AnnexWatch.html ~www/htdocs  
# chown www ~www/htdocs/AnnexWatch.html
```

The setup requires the driver to be in the cgi-bin directory. You may open up the AnnexWatch.html file and customize it if desired. You may also want to place the AnnexWatch.html into some other directory and we leave it up to you how you link into it.

Load the AnnexWatch.html into your favorite web browser and it will display the AnnexWatch web driver start button:



This will launch the AnnexWatch web driver and allow you to specify the command line options in an orderly form. Specify the location of the acp\_logfile you wish to examine and choose the output mode desired:

**1) Enter location and the name of Annex Logfile to examine:**

**2) Pick desired a run mode:**

- ◆ *Quick Table Output Mode*
- ◆ *Activity Table Output Mode*
- ◆ *Main Table Output Mode*
- ◆ *System Setup*

The default output mode is for a quick table, if you chose this option the next page will appear which asks for the type of quick table to be generated:

**3) Pick desired quick tables:**

- Create table of Usernames found.*
- Create table of Portnumbers found.*
- Create a Summary table.*

A Summary table is the default. At this point picking the 'Run AnnexWatch' button will execute AnnexWatch with the required options and return the following results (these results can be compared to the 'text' results in the examples section for the summary table.

# AnnexWatch Results

*Equivalent command:*

```
/tmp/AnnexWatch/usr/annex/acp_logfile -http -quiet -summary > AnnexWatch.out
```

## Statistics

	<b>From</b>	<b>To</b>
Entire File	03:03:57 pm, May 27 1995	11:51:32 pm, Aug 23 1995
Period Examined	03:03:57 pm, May 27 1995	11:51:32 pm, Aug 23 1995

## Summary for Entire File

Period Length	88 days, 527.6 minutes
Total Number of Users	23
Total Number of Ports Used	5

## Summary for Examined Period

Period Length	88 days, 527.6 minutes
Total Number of Rejects	227
Total Number of Timeouts	0
Total Number of Boots	3
Total Number of Login Fragments	10
Total Number of Logout Fragments	1

AnnexWatch run on August 22, 1996 18:14:43 ET (2314 GMT)

Note that the equivalent command line is shown at the top of the page. This option may be turned on or off through the 'System Setup' option back at step 2. Click on the 'back' button on your browser twice to get to step 2. Pick 'System Setup' and the OK button. The following page will appear:

### 3) Describe the system setup:

*Enter location and the name of Annex Logfile to examine:*

`/usr/annex/acp_logfile`

*Enter location of AnnexWatch on your local server:*

`/usr/local/AnnexWatch/`

*Enter location and the name of AnnexWatch license file:*

`/usr/local/AnnexWatch/AnnexWatch.license`

- Mute warning and error messages in the output.*
- Output results using http tables.*
- Show equivalent commandline usage.*
- Interpret dates as DD/MM/YY instead of MM/DD/YY.*

Make these the System Defaults

Clear

These are the current default values, changes to these settings will effect future sessions. If you have placed your `acp_logfile` and/or `AnnexWatch` files in different directories make adjustments here. The location of the `acp_logfile` here is the system default for step 1.

Returning to steps 1 and 2 by either committing your choices to defaults, or clicking your browsers 'back' button will allow you to explore the remaining two table options. Choose 'Activity Table Output Mode' for step 2 and click OK. The following page will appear allowing you to make typical usage profiles over a day or a week. This section will also allow you to change the bin size to alter the output resolution.

### 3) Create an Activity Table, using the following Parameters:

*Display usage over a typical*

day

*Statistics binsize (in minutes):*

60

Run AnnexWatch

Clear to Defaults

Running AnnexWatch with the default values produces a typical day usage profile. In this profile the accumulated number of connections for each time period is shown. This information shows that over the entire time period in the logfile 25 connections occurred between midnight and 1 am.

### Typical Day Total Activity

Time	Total Connections	
0:30 am	24.93	[=====]
1:30 am	11.17	[=====]
2:30 am	4.80	[=====]
3:30 am	3.02	[=====]
4:30 am	3.10	[=====]
5:30 am	2.93	[=====]
6:30 am	5.68	[=====]
7:30 am	12.60	[=====]
8:30 am	8.82	[=====]
9:30 am	14.23	[=====]
10:30 am	16.15	[=====]
11:30 am	15.38	[=====]
12:30 pm	13.18	[=====]
1:30 pm	17.65	[=====]
2:30 pm	19.33	[=====]
3:30 pm	17.13	[=====]
4:30 pm	15.42	[=====]
5:30 pm	18.88	[=====]
6:30 pm	16.10	[=====]
7:30 pm	9.03	[=====]
8:30 pm	17.15	[=====]
9:30 pm	21.13	[=====]
10:30 pm	22.60	[=====]
11:30 pm	24.23	[=====]

Hitting 'back' in your browser twice will allow you to explore the main tables. Pick 'Main Table Output Mode' in step 2 and click OK. This will bring up the page where you can describe the amount of detailed statistics you require. Step 3 asks you to choose how the table will be ordered and what unit to display the accumulated times in:

**3) Create an Main Table, using the following Parameters:**

*Tabulate Annex activity by*

*Sort Main Table by*

*Display results in units of*

**4) Limit the Scope of the Main Table, using the following Parameters:**

*Include all users in forming the results.*

*Include only these users in forming the results:*

*Include all ports in forming the results.*

*Include only these ports in forming the results:*

*Always exclude the following user names:*

*Always exclude the following port numbers:*

*From the beginning of the file.*

*From this point: Date*  *Time*

*To the end of the file.*

*To this point: Date*  *Time*

*Suppress reporting records with zero times.*

On the same page is Step 4 which allows you to limit the scope over which the data is collected. You may limit the users surveyed, the ports surveyed or the time period used. Running AnnexWatch with the default values (except units in hours) produces a main table statistic:

<b>USER TABLE (all times in hours)</b>								
<b>Username</b>	<b>Ports</b>	<b>PPP Time</b>		<b>SLIP Time</b>		<b>CLI Time</b>		<b>Total Time</b>
<i>spot</i>	3	54.06	69	0.00	0	1.65	79	55.70
<i>ernie</i>	2	0.02	1	37.58	53	0.46	55	38.06
<i>littlej</i>	2	0.00	0	14.23	49	21.58	140	35.81
<i>barney</i>	3	0.00	0	29.45	37	0.13	40	29.58
<i>pluto</i>	2	0.00	0	0.00	0	24.76	53	24.76
<i>fred</i>	2	0.00	0	23.54	36	0.01	36	23.55
<i>leo</i>	1	22.83	48	0.00	0	0.14	48	22.98
<i>minnie</i>	3	0.00	0	22.72	159	0.10	163	22.83
<i>zeus</i>	3	0.00	0	14.38	47	0.03	47	14.41
<i>olive</i>	2	0.00	1	12.84	57	0.11	61	12.95
<i>mickey</i>	3	0.00	0	6.21	43	4.44	69	10.65
<i>jane</i>	2	0.00	0	2.97	8	5.89	26	8.87
<i>jupiter</i>	2	0.00	0	0.00	0	7.97	42	7.97
<i>bert</i>	2	0.00	0	0.00	0	7.90	36	7.90
<i>junior</i>	2	6.27	12	0.00	0	0.71	20	6.98
<i>goofy</i>	4	2.22	8	1.51	7	1.75	35	5.47
<i>larry</i>	2	0.00	0	3.86	28	0.01	28	3.87
<i>dizzy</i>	2	3.53	15	0.00	0	0.10	18	3.63
<i>norman</i>	1	0.00	0	2.25	23	0.27	24	2.52
<i>newton</i>	1	0.00	0	1.96	5	0.01	5	1.96
<i>charlie</i>	1	0.00	0	1.63	4	0.00	4	1.64
<i>betty</i>	2	0.00	0	1.55	9	0.01	9	1.55
<i>ronald</i>	1	0.00	0	0.18	1	0.00	1	0.18
<b>Totals</b>		<b>88.92</b>	<b>154</b>	<b>176.85</b>	<b>566</b>	<b>78.04</b>	<b>1039</b>	<b>343.82</b>

## 8. EXTENSIONS: CRONJOBS

You are encouraged to extend AnnexWatch using shell scripts and cron jobs. If you come up with interesting examples feel free to let us know. The following are included as examples.

### Cron Jobs

AnnexWatch may be used for automated reporting via cron entries. For additional information on cron jobs, refer to your system administration manuals. This type of automated reporting is fairly straight forward, examples follow:

```
#
# On the first of every month, gather a general report
#
0 1 1 * * /usr/local/AnnexWatch/AnnexWatch -all > /usr/local/AnnexWatch/report
```

For specific event reporting, running the full version of AnnexWatch may be too slow, therefore a speedy event reporter is available. The program is *AnnexEvent* and the shell program designed for its cronjob use is *TestEvent*.

```
#
# Every 6 hours see if that pesky user foo is on my special reserved port 34
#
15 0 * * * /usr/local/AnnexWatch/TestEvent root acp_logfile foo login 34
15 6 * * * /usr/local/AnnexWatch/TestEvent root acp_logfile foo login 34
15 12 * * * /usr/local/AnnexWatch/TestEvent root acp_logfile foo login 34
15 18 * * * /usr/local/AnnexWatch/TestEvent root acp_logfile foo login 34
```

AnnexEvent takes care of triggering the requested event the first time it occurs. AnnexEvent keeps a record of the events that it has seen so you are only notified once each time the event occurs. AnnexEvent keeps track of notified events in a AnnexEvent.recordfile. TestEvent takes care of running AnnexEvent and emailing a specific user when the event occurs (root gets the email in the above example). If foo logs in 400 times between 1 am and 3 am on port 34, AnnexEvent only triggers one emailing at 6am to root. By Midday, if foo has had no further activity, AnnexEvent triggers no emailing.

*The TestEvent script follows:*

```
#!/bin/csh -f
#
# AnnexWatch event tester (AnnexEvent) for cron jobs
# Version 1.0
#
# AnnexWatch Suite
# (c) G&R Data Group, 1996
#
# USAGE: TestEvent user_to_notify logfile event [username [port]]
#
AnnexEvent $2 $3 $4 $5
if ($status == 0) then
    echo Dear $1, > AnnexEvent.letter
    echo >> AnnexEvent.letter
    echo It would appear that your event: $3 $4 $5 >> AnnexEvent.letter
    echo has occured in $2. Specifically: >> AnnexEvent.letter
    echo >> AnnexEvent.letter
    cat AnnexEvent.hitfile >> AnnexEvent.letter
    echo >> AnnexEvent.letter
```

```
echo This event will not be reported again until >> AnnexEvent.letter
echo this event is removed from the event record >> AnnexEvent.letter
echo file: AnnexEvent.recordfile. >> AnnexEvent.letter
echo >> AnnexEvent.letter
echo Have a great day, >> AnnexEvent.letter
echo AnnexWatch. >> AnnexEvent.letter
mail $1 < AnnexEvent.letter
endif
```