PingPlotter Pro

(c) 1998, 2014 Nessoft, LLC

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: May 2014

Special thanks to:

All the people who contributed to this documentation and to the PingPlotter software.

Special recognition to:

Pierre Schwab
Tom Larrow
Jenna
Clara
Andrea, D.D., Nina
Mele
Cam M.
Larry M.
# Table of Contents

## Part I Introduction
- 1 Getting Started Guide ........................................................................................................... 48
- 2 Interpreting Results - A Quick Example ................................................................................ 48
- 3 Interpreting Results - Longterm Monitoring ....................................................................... 48
- 4 Interpreting Results - Gamers ............................................................................................... 49

## Part II How PingPlotter Works

## Part III Operation

### 1 Basic Settings
- Event - Tray Icon Change ........................................................................................................ 38
- Event - Log to file ....................................................................................................................... 37
- Event - Tray Icon Change ........................................................................................................ 38
- Variable Substitution ................................................................................................................ 38
- Event - Tray Icon Change ........................................................................................................ 38
- Email Setup (for alerts) ............................................................................................................. 34
- Event - Play a Sound .................................................................................................................. 35
- Event - Send Mail ....................................................................................................................... 35
- Event - Launch an executable ..................................................................................................... 36
- Event - Log to file ....................................................................................................................... 37
- Event - Tray Icon Change ........................................................................................................ 38
- Creating / Configuring Alerts ..................................................................................................... 28
- Troubleshooting Alerts ............................................................................................................. 33
- Email Setup (for alerts) ............................................................................................................. 34
- Alert Events ............................................................................................................................... 34
- Event Notification ...................................................................................................................... 34

## Part IV Common Tasks

### 1 Getting notified of a network problem (alerts)
- 1 Getting notified of a network problem (alerts) .................................................................... 28
- Creating / Configuring Alerts ..................................................................................................... 28
- Troubleshooting Alerts ............................................................................................................. 33
- Email Setup (for alerts) ............................................................................................................. 34
- Alert Events ............................................................................................................................... 34
- Event Notification ...................................................................................................................... 34
- Event - Tray Icon Change ........................................................................................................ 38
- Event - Log to file ....................................................................................................................... 37
- Event - Tray Icon Change ........................................................................................................ 38
- Variable Substitution ................................................................................................................ 38
- 2 Running as a service ................................................................................................................. 39
- 3 Exporting data for further analysis ......................................................................................... 41
- 4 Long term monitoring and auto-saving data .......................................................................... 43
- 5 Automatically start tracing when starting PingPlotter ........................................................... 44

## Part V Understanding Output
- 1 Getting Started Guide ............................................................................................................. 48
- 2 Interpreting Results - A Quick Example ................................................................................ 48
- 3 Interpreting Results - Longterm Monitoring ....................................................................... 48
- 4 Interpreting Results - Gamers ............................................................................................... 49

(c) 1998, 2014 Nessoft, LLC
Part I

Introduction
1 Introduction

PingPlotter Pro is a network troubleshooting and monitoring tool. Its goal is to monitor network connections, and then to help identify the source of the problem (when problems occur).

PingPlotter was first conceived in early 1998 to troubleshoot problems with a cable modem – with an "end user", consumer focus. Over the years, it has grown to include a lot of capabilities, covering everyone from a "weekend troubleshooter" to the full-time network administrator. The Pro version of PingPlotter takes these capabilities to new heights, targeted specifically towards the network professional.

Where to find additional help

If you can't find an answer to your question (or see your area of interest) in this documentation, we have more information available online at the PingPlotter Pro web site (http://www.pingplotterpro.com).

PingPlotter Pro is a superset of the features available in PingPlotter Standard 3.0 (or the PingPlotter 2.x line). The same concepts apply, and some documentation is shared between the versions.

For more information on how to use PingPlotter, or if you have questions that are not covered here, please check our support topic in this help file. Make sure you read the Getting Started Guide – it has a lot of great introductory material.

In addition, our knowledge base offers a lot of answers to questions that might be of low interest to some people (but a lot of interest to a few!). We also have a search engine online that indexes our support forums, knowledge base and the rest of the site. This is highly recommended if you have questions you can't find the answer to in this documentation.
Part II

How PingPlotter Works
2 How PingPlotter Works

At its heart, PingPlotter is a trace route utility. It's souped up and on steroids, but the basic data it collects is based on the theory of trace route.

A ping packet is an IP packet requesting that a copy of its contents be echoed back to the sender. When you "ping" a site, you send over an echo request and that site responds back that it received it. The amount of time it takes for the packet to get to that site, and then return to you, is the ping time, or latency. In general, the lower this is, the better your connection to the site. This time is usually specified in milliseconds (1/1000 of a second).

One of the parameters on a ping packet (and any packet, but we're only talking about ping here) is something called "Time to live" (TTL). TTL is an IP header field designed to keep packets from running in loops, essentially forever, throughout a network (this can happen when there is a route change, and the routers involved don't all know the same information as new information is being replicated out). Initially it's usually set to somewhere between 64 and 255, and is reduced by 1 every time it passes through a server.

If the TTL should ever reach zero, the packet has expired, and the router that it's passing through will send it back to the source. Again, this happens so that packets don't get caught in an endless loop.

Trace route plays with this TTL number on outgoing packets. It first sends out a packet with a TTL of 1. The first router that sees this decrements it to 0, and then sends it back. It also sends back its own IP address with the packet, and DNS is used to do a lookup for an actual domain name.

Ok, so next, traceroute sends out a packet with a TTL of 2 so it can find out what the next computer in the route is. Then it sends out a packet with a TTL of 3. This process is repeated until the final destination is reached. At that point, you know the entire path the packet has traversed to reach the destination computer/router. Each server/router in this chain is called a hop.

This method can help us determine the route a packet takes, but if we time each of these packets, we also know how long it takes for a packet to make it from our source PC, to that router, and then back again. This is called latency.

The last hop in a (successful) trace route is actually the round-trip time to the destination server. This is an important concept to understand. You don't add up all the times between you and the destination host - as that time has already been added. The time to the last hop in the chain is exactly the same as is if you'd used a ping utility to that host. So a trace route utility is actually two utilities - ping AND trace route.

PingPlotter speeds up this process by sending out packets to the first 35 servers in the route all at the same time. This makes a HUGE difference in overall speed. It also means that the network conditions for each hop are very similar - so the numbers are better compared.
Part III

Operation
3 Operation

3.1 Basic Settings

1. **IP Address or DNS name** of the destination you want to trace. If you enter an IP Address here PingPlotter will start tracing immediately before the IP is resolved to a name. The name will show as “resolving” until the request is complete.

2. If the site you wish to trace is already listed, you can select it instead of typing it in. To delete a host, right-click it in the list and select “Delete”. Double-clicking on a site in this list starts a trace to that site.

3. The **Route Change** pane is used to show the history of route changes. Anytime any hop in the route changes, PingPlotter stores the old and new route data and adds the time of the change to this list box. Double-clicking on any time will show the route as of that time. This is the starting time for the change.

4. The **# of times to trace** allows you to stop tracing after a certain number of times. If you’re only interested in a set trace count, you can save some bandwidth by not allowing PingPlotter to trace forever.

5. The **Trace Interval** is the amount of time PingPlotter will wait between each sample set. If you’re doing a long term monitoring project, you may want to set it to be 1 minute (or more). If you’re doing a quick test, you might want to set this to something lower (5 seconds or 10 seconds). If the up/down arrow doesn’t have the amount of time you want, just type the time interval you want (e.g. 3.5 seconds).

6. When running a trace, PingPlotter can look at just the most current **samples**. This is great to watch “trending” (where the response changes over time). If you include ALL samples (type 0 in this field for ALL), then after a lot of samples, new samples don’t affect the graph very much. Setting this to something like 10 allows you to see how the response times are right now. All numbers in the trace (upper) graph are affected by this. When zooming in on the timeline graph, it’s important to not have this set to “ALL”. 

---

Did you know?

To restart a trace from scratch to the same host without restarting PingPlotter: right-click on the Trace/Resume button and select “Reset & Restart”
### 3.2 The Interface - Graphs

The graphs are where PingPlotter really shines. At a glance, you're able to visually see where a problem lies. There are actually two graphs available, the **Trace Data Graph**, and the **Timeline Graph**. We'll explore both in this section, as well as some other items related to them. Please refer to the below image (annotated with the blue numbers) that we've saved from PingPlotter (File>Save Image), and the explanations (referencing the blue numbers) below the graphs.

The upper graph is called the **Trace Graph**. All columns on the trace graph are resizable. The lower horizontal graph is called **Timeline Graph**.

All numbers on the trace graph use the "Samples to include" setting to control how many samples are
used in the calculations. Also, by default you're looking at the most recently collected samples, but you can also focus on samples that are not the most recent samples by double-clicking on the time graph. This will focus the upper graph on that period you double-clicked.

1. Shows you the thresholds you've set for the colors on the trace graph background. In this case we've set 50ms as the warning color, and 90ms as what we thought was a critical speed. PingPlotter defaults to 200ms and 500ms for the warning and critical values, however you can change them as we did here in the Display section of the Options dialog available under the Edit menu.

2. The red line on the graph represents the average response time for each host for the currently selected samples. The blue X represents the response time for the current packet (can be turned off if you're sending a copy of your graph to somebody - some folks find it confusing when they're not actually there watching the live trace). The black horizontal lines represent the minimum and maximum response times. The red horizontal bar shows the packet loss for that hop (same as the PL% column, but there for readability). PingPlotter uses a dynamic scale for its graph. The bottom number is usually 0, and the top number represents the maximum response time in milliseconds. If you wish, you can change this to a fixed scale in the Display section of the Options dialog available under the Edit menu.

3. The Avg column shows the average response time of the last N samples (where N is the samples to include). Any timeouts/lost packets are not included in this value. The Cur column shows the individual sample time of the most recent sample included in the set. If a number is displayed as ERR, that means the packet was lost, i.e. a packet was sent out, but never made it back.

4. This column shows the DNS name of the device for that hop. A -------- in this column indicates that PingPlotter was unable to resolve a name for that device's IP address. This is not a flaw in PingPlotter. It just means that your DNS server doesn't have a name for that IP address, or that address just doesn't have a DNS name period.

5. The Round Trip line is basically there for ease of reading. It's the same value as the last server in the route. This is the time it takes for a ping to get from your computer to the target device and back.

6. The Timeline Graph (TG) is one of the most powerful features in PingPlotter, and great for long-term monitoring projects you may be doing.
   - Double clicking anywhere on the TG will focus the trace data graph (top graph) to that particular point in time. This is particularly useful for investigating spikes or time-outs (see next item).
   - A red line on this graph denotes a time-out for that period. Double clicking anywhere on the graph shows you the trace data detail for that period in the upper graph (denoted by the two vertical "focus" bars you see in the example above - the "Samples to Include" setting controls the number of samples encompassed by the "focus" bars). This allows you to see what was happening along the route when that particular time-out occurred.
   - Right clicking on the TG allows you to pick the time period you want to view - from 60 seconds up to 48 hours. This value affects all timeline graphs. If you'd like to add additional time periods you can do so by modifying your pingplotter.exe.ini file. This is covered in the Advanced Topics and Tips - Timeline Intervals section of this tutorial.
   - You can slide the graph (by doing a "Click-Drag"), allowing you to look at the past data for this trace. You can right-click and select "Reset focus to current" to move all graphs back to the current time. If you've been running the trace for a long time, it's helpful to adjust the time period you have set before you start moving back in the history, i.e. if you're going back a couple of days you might want to set the time period to 24 hours first.
   - You can see a TG for any hop by right clicking on that hop in the trace graph and selecting "Show this timeline graph". You can also get rid of that TG by right clicking and deselecting it. An underline appears under the hop # to show that a TG is being shown for that hop. A shortcut for
this is to double-click on a hop # to show / hide the time graph.

- We discuss time graphs a bit more in the "Timeline Graphing" and "Timeline Graph Annotation" sections.

7. Comments are denoted by a red up arrow, and are a very handy feature enabling you to add a comment (that is saved when you save the sample set) for things like planned outages, configuration changes or whatever else you want to make a note of.

8. The **IP address** of the router that reported back for that hop.

9. The **PL%** indicates the number of packet(s) that have been lost in the current sample set. If you're only including the last 10 samples, then only the number of lost packets in the last 10 samples are shown here. If you want to find out how many time-outs have happened over the entire session, change the "Samples to Include" to 0. This is important. There is no ALL setting for this. **0 = ALL**.

10. The **number of hops** that device in the route is from your computer. If a hop has brackets around it (like [2]), this means that hop is being monitored for an alert (alerts are covered in the advanced settings section of this tutorial). Multiple alerts can be configured for the same IP, and alerts don't work unless some IP in your current route is being monitored (i.e. has brackets around it). If a hop has an underline under it (like [[8]]), that hop is being traced on a time-line graph.

11. The brackets ([[]]) on a hop shows that you have an alert tied to that hop. Alerts are covered in depth in another section of this document.

12. **Beginning and ending times/dates** for the trace. Very useful if you're saving off graphs. It's nice to know the time window the trace was done in. On "Copy as image" or "Save as image"-created images, you'll see the "XXXX samples timed:" value used to create this graph (where XXXX = Samples to include). In the live application, this line isn't shown because it is available elsewhere.

13. The **DNS name and IP Address** for the host you're doing the trace to. If you've got multiple instances of PingPlotter running, this is in nice big letters so you quickly know which trace you're looking at.

**Tools and other options available for the Trace Data Graph**

You can display the **Minimum and Maximum columns** by right clicking on the upper trace graph and selecting them (via the "Customize View..." option).

You can copy the IP address or DNS name for a hop to the clipboard by right-clicking on that hop, selecting the **Clipboard** option and then clicking on what you want to save.

You can do a **WhoIs** on a particular hop by right clicking on it and selecting WhoIs Information. Note that by default this queries whois.crsnic.net. You can add additional WhoIs servers by editing your pingplotter.exe.ini file. Instructions for doing this are in the Advanced Topics and Tips - WhoIs section.

You can lookup who owns the particular IP range a hop is in by right clicking and then selecting "IP Block Lookup (ARIN)".

### 3.3 Summary Graphs

The summary graph screen is a listing of any final destinations or intermediate hops from any of your destinations. This summary gives you a way to compare multiple targets, and also gives you an easy one-glance summary of all the important parts of your network.

**Adding targets to the summary screen**

By default, when you trace to a new target, that final destination will automatically be added to the
summary graph screen, and a time-graph for that summary will be added as well.

In addition, if you want to manually add a target (or any intermediate router) to the summary graph, just right-click on that hop in the trace graph and select "Show on Summary Screen".

At any time, you can turn on or off a time graph by double-clicking on that router / target in the upper graph.

From the summary screen, you can switch to the trace graph by right-clicking on a host and selecting the menu option "Show trace graph".

You can sort the list of targets by clicking on a graph heading. Clicking again reverses the sort order.

### Resizing the time graphs

Once you get a few targets on the summary screen (and a few time graphs), you may wish you could resize (make them taller or shorter) the time graphs but are unable to because of the existence of the scrollbar on the time graph area.

To resize all of the time graphs, scroll to the last summary graph (the scroll bar should be at the bottom of its range). Now, float your mouse cursor between the lowest graph and the one above it - you should get a resize pointer and you can then size the graphs at your pleasure. All time graph will resize to match.

![Resize pointer example](image)

**Time periods on the summary screen** (ie: "samples to include")

The summary graph screen works slightly different than some of the other graphs in that the upper summary graph may have any number of targets, all using different collection methods and possibly different trace intervals. Because you might have different trace intervals (and even different time scales), the "Samples to Include" concept is replaced by a "Graph focus time". This Graph focus time is a time-based setting, rather than a sample count based setting like the trace graph uses.

As a related note, you can also enter a time period on a trace graph - and PingPlotter will convert that to a sample count - in that case, though, it's a one-time deal. On the summary screen, the time period is constant and the sample count (internally) is dynamically updated as samples are collected.

### Other notes

Many features on the summary screen are similar to those on the primary target screen, so see that documentation for more details.

### 3.4 Timeline Graphing

The timeline graph feature of PingPlotter is specifically built for long-term monitoring projects.

Often, your ISP problems occur when you're not watching it. The timeline graph feature of PingPlotter allows you to trace over a long period of time - and then look back over this history to find when and where the problem occurs.
By default, PingPlotter will automatically trace the last hop (the host you're tracing to) on a time-line graph. You also trace any of the other hops by right clicking on that hop in the trace graph and then selecting "Show this Timeline Graph". You can also turn off any graph by this same mechanism.

Once tracing, PingPlotter will record each sample on the time-line graph. The amount of data displayed on the graph can be changed by right-clicking on the timeline graph - then selecting the amount of time you want displayed. This scale affects ALL timeline graphs and is saved when you shut down PingPlotter.

If there is more data collected than you can show on the timeline graph with your selected scale, click on the graph and while holding down your mouse button, drag the graph. This will allow you to move back in history and examine the samples. You can also use the keyboard to scroll. To scroll with the keyboard, select the timeline graph and use one of the following keys:

- **ALT-HOME**: scroll to the beginning of the collected data
- **ALT-END**: scroll to the end of the collected data
- **ALT-LEFT**: scroll back in time (about 5% of the graph width)
- **ALT-RIGHT**: scroll forward in time
- **ALT-PGUP**: scroll back in time (about ½ of the graph width)
- **ALT-PGDN**: scroll forward in time

If you find a time period that looks “interesting”, you can double click on the timeline graph at that point. The upper graph (the trace graph) will move to that time period. This allows you to troubleshoot where the connection problem may have occurred. Keep in mind that the upper graph always focuses in on the number of samples in "Samples to include". If you have that set to ALL, then the upper graph can't zoom in on any piece of the graph.

It may be that you see a problem period in the 48 hour graph. Double-click on that period, and the upper graph will focus on that period. You can then change your timeline graph scale and the lower graphs will try to stay focused on the time period you selected. This makes it easy to see details about a specific period.

Another handy way to navigate through the time graph is with the mouse wheel. If you select a graph, scrolling your mouse wheel will move the graph forward and backwards. If you hold down the mouse wheel while you scroll, you'll zoom in on the period your mouse is pointing at.

When you're done zooming the graph, right-click on it and select "Reset focus to current". This will return both graphs (the timeline and the trace graph) to the current time.

Because PingPlotter takes so little CPU time when running, you can set it up to trace while you're doing something else. One particularly interesting way to use it is to set it tracing to a site that is hosting a game you're playing, or server you're using (streaming stock quotes, for example). If you notice your response slowing down, you can check out PingPlotter and use it to quickly troubleshoot where the problem is coming from.
### 3.5 Timeline Graph Annotation

When using PingPlotter to monitor long-term connectivity/latency, you sometimes run across a situation where your network was affected by something you know about (power outage, reboot of router, big file download, etc.). These incidents show up in PingPlotter. Also, you may run into a situation that you don’t control - but you know what’s causing it. You want to take notes about this kind of thing and tie it to your data.

PingPlotter allows you to make notes on your time graph. These notes show up on the graph as triangles, and when you float over them, the notes appear. When you save data as an image from PingPlotter, these notes get attached to the image, along with the times they happened. This is especially valuable when you want to send an image to a network provider and there are events on the data you're sending them that need to be explained.

You can mark these on the time graph and when you create the image, the notes will be included.

To create a note, just right-click the lower time-graph at the point you want to create a note and select “Create Comment” from the popup menu. PingPlotter will prompt you for the note, and then draw a red triangle on the lower edge of the time graph. You can float your mouse over this triangle to see the note, or you can right-click on the triangle to edit or delete it. You can edit or delete an existing comment by right-clicking on the comment.

### 3.6 Tracing to Multiple Targets

You can trace to more than one target at a time within one PingPlotter instance, and each target can be managed independently of the others (unlike in MultiPing, where each target’s schedule, packet size and other settings are all shared).

**Adding targets to PingPlotter Pro**
To use this capability, enter your first target's name or IP address in the "Address to Trace" field and hit either the "Trace" button or the enter key. Now, to add an additional target, either use the "File" -> "New Target" menu option, or hit the little icon next to the "Address to trace" line:

```
Address to Trace: [Input Field]
```

If these entries are disabled, that means PingPlotter's already waiting for you to enter a new target address in "Address to trace:" and get started.

Multiple targets will show up as tabs. In addition, each target will be listed as a menu entry in the "Workspace" menu to make it easy to select.

All menus and controls are connected to the currently selected target. Switching between targets will switch the controls to that target.

To close a target, just hit the "X" (i.e.: Close) button above the graph on the right next to the thumbtack. This will stop data collection and get rid of all collected data from memory.

PingPlotter Pro works great up to about 40 targets. Somewhere around this number, it starts to get a bit unwieldy because of the number of tabs - and entries in the Workspace menu. We've had it up to 250 targets, but this isn't an entirely smooth experience - just from a user interface standpoint.

Remember that for each target, we're doing an entire trace.

One primary reason to use PingPlotter Pro to monitor multiple targets, rather than just using multiple instances of PingPlotter standard, is that PingPlotter Pro can make sure multiple targets don't all get queried at the same time. We always want a bit of time between each packet so we don't overwhelm the local connection and impact our measurements by sending too much data. PingPlotter Pro makes sure time spacing is honored between each packet - no matter how many targets are being monitored.

**Docking, floating etc.**

By default, each target shows as its own tab. That works, but might not be exactly what you're after. Maybe you want 10 targets all displayed filling your high resolution "war room" screen. You can certainly do this.

Each target is living in its own "dock" control. Just grab a tab and drag it away from the PingPlotter main application (or double-click a tab to undock and redock it). If you drop it someplace else on your screen, you'll now have a "stand-alone" display for that target. You can also dock a target on the side of the PingPlotter window so you can have side-by-side (or above / below) display in the PingPlotter main program instance. There are a number of combinations here.

One of our favorite setups with 5 to 10 targets is to dock the trace graphs to the side, and then set them to auto-hide (this is done by hitting the push-pin button at the right side of the caption). If you leave the summary screen visible in the center and auto-hide the targets on one (or more!) sides, you can easily access them by floating your mouse over, by selecting that target from the "Workspace" menu or right-clicking on that summary graph and selecting “Show Trace Graph”
3.7 Named Configurations

PingPlotter Pro allows you to run multiple targets in a single instance. You may not want to use the same settings for all of these targets, though. Maybe you have different expected latencies and need your good / warning / bad colors displayed at a different point. Maybe you want to use ICMP packets on one target, and TCP packets on another. Maybe you want to auto-save data from some targets, but not others.

PingPlotter Pro allows you to do this with the concept of "Named Configurations". Each named configuration can be associated with any number of targets, and changing the settings for this configuration changes that setting for all targets that are using it.

Named configurations are created and manipulated in the "Options" Menu. When you first install PingPlotter Pro, you will have one named configuration out of the box - called "Default Settings". Under that name in the tree view, you'll see a number of additional "configuration specific" areas - for display, engine, auto-save and route changes. The values you configure are all part of this named configuration. At any time, you can edit the name (and it's recommended that you do edit this name when you start to use this feature), or make changes to any of the values.

To create a new named configuration, just right-click on any existing configuration and select either "New Configuration" or "Copy Configuration" from the popup menu. Selecting "New Configuration" will use whatever PingPlotter Pro default values are, while copying the configuration just makes a duplicate of the configuration you right-clicked.

Once you've created a new configuration, you should rename it. In fact, once you create a new one, you should also change the "Default Settings" to a more appropriate name (like ICMP Packets, or something that identifies it well when comparing with the new configuration you just created).

Just change the label and enter a description (if you want to describe what you're doing with this, so you can remember later).

Some example configuration names are:

TCP Port 80
Auto-Save every 30 minutes
VoIP
Remote via our website
Now, when you close the options dialog, you'll notice a drop-down below your "Address to trace" area labeled "settings", which lists all the named configurations you have set up.

Changing this drop down will change the currently running target to use that named configuration set.

Any changes you make to a specific configuration are not carried over to any other named configurations.

The "Summary" screen's display settings come from whatever configuration is first in the list. To make a configuration work on the summary screen, move it to the top of the list using the right click menu. Using the "Set as default" option from the menu moves the selected configuration to the top of the list. The configuration at the top of the list is the default configuration.

3.8 Web Interface

When enabled, the PingPlotter Pro web interface allows access to many of the options of the PingPlotter client, but without having to be local on the machine that is collecting information.

Enabling the web interface using the built-in web server

The easiest way to try out the web interface is to go to Edit -> Options, Web Server section, then turn on the web server. Leave the port at 7464. For local testing, turn off the checkmark for "Use Security". Hit "OK" on the options dialog, applying the changes and closing it.

Just to make things easy, let's trace to a target from the main user interface. Pick one of your favorite targets from the history in "Address to Trace" and start tracing.

Accessing the web interface from your local machine

To access the web interface from your local machine, enter this URL in your browser (or, if you're accessing this from online help, just click on this link):

http://localhost:7464

This should open your browser to the PingPlotter web interface, which lists all the current targets and allows you to pick one (from the list on the left) to see for details.

You can trace to a new target, stop and start an existing trace session, close an existing session, change trace intervals and do basic examination of the data.

Doing basic analysis

There are some basic operations you can do from the web interface, but some things require the local client. Here are some of the operations you can do.

- Change the length of time you're examining on the time graph. There is a combo box below the graph that lets you pick a different time.
- Change "Samples to include" to look at a bigger or smaller time window in the trace graph.
- Turn on or off a time graph for any hop. To do this, right-click on the hop (Internet Explorer and Mozilla Firefox only), or double-click on a hop number (IE, Firefox and Opera).
- Download sample data for more analysis with a local instance of PingPlotter
- Do whois lookups through www.whois.sc.
• Add / Remove columns (IE/Firefox only)
• Change column widths

**Browser Requirements**

Most of these operations require Javascript be enabled. With lots of Javascript running, newer browser versions are going to be faster and have a richer interface. Although things work on Internet Explorer 6, Firefox 1.03, Opera 8.5, these browsers are pretty old - we recommend IE8, Firefox 10+, browsers that are generally less than a couple of years old. If you don't have javascript enabled, you'll still be able to do a little bit, but you'll be working against an image of the graph rather than a locally drawn version (so you won't be able to do a lot of things like turning on time graphs or resizing columns).

**Using Microsoft IIS instead of the built-in web server**

Using the built in web server is really easy, and it works great for basic use, but it doesn't have some of the options that using a more industrial strength web server does. Here are some of the reasons you might want to use IIS:

• You need more than one login (you can use IIS to set up additional logins)
• You want to customize the web interface and use server variables (the list of server variables in the built-in server is pretty close to non-existent, including, currently the ability to get the IP Address of the person connected).
• You want to customize the web interface with cookies and/or session variables.
• You want to use a web server with known issues
• You want to share a web server with other applications
• Others...

The PingPlotter web server *emulates* an ASP-based VBScript IIS server, so the exact same source files can be used for both. These source files are installed in the following location (if you use the default path):

c:\program files\PingPlotter Pro\www

If you create a new IIS application and point it to this directory, it should work. The default page is called "Default.asp", which should work without special configurations when you point IIS at this directory.

For best results, PingPlotter Pro should be running as a service. There are security implications (COM object security) that bite us if we don't use a service.

We have had some reports of people needing to open up DCOMConfig and change some of the security options to make things work. If you have problems with this, let us know.

You shouldn't be required to make any changes to any .asp files.

We will certainly find more information about what works and doesn't work here on some machines, so feel free to contact us if you have any questions or problems - support@pingplotter.com.

**Troubleshooting IIS problems**

• If you're getting a 404.3 error in your browser after adding the IIS application, then you probably don't have ASP functionality enabled on the IIS server. PingPlotter uses ASP and requires the ASP options to be installed. This is an option in the IIS configuration. Enable ASP; the .NET version is not necessary for PingPlotter.
• If you get permission errors (a red screen with a permission-based error about the PingPlotter object), run `dcomcnfg`, fold open "Component Services", "Computers", "My Computer", "DCOM Config" and look for "PingPlotter". Right-click and go to properties. In "Security", change Launch and Activation Permissions to Customize and add the IUSR user. Turn on local launch and local activation for the IUSR user.

We will certainly find more information about what works and doesn’t work here on some machines, so feel free to contact us if you have any questions or problems - support@pingplotter.com.

### 3.9 Reporting

PingPlotter doesn't have any built-in **printing** options. What it WILL do for you though is copy its data to the clipboard or text file and let you manipulate it in your favorite software package from there.

There are several ways to output your data.

1. **The graph.** There are a few ways to get a picture as shown in the graph. A quick way to do this is to select the Edit/Copy as Image option from the main menu to copy the graph and the legend to the clipboard. The column and graph sizing will match what’s on the screen, so make sure everything you want printed is displayed. You can paste this into MS Paint or any other graphics application to print or save. You can also go to the file menu and "Save as Image" in PNG, BMP or GIF format. You can have PingPlotter automatically create these images for you as well, by scheduling it in the Options menu.

2. **Raw data - in TraceRT format.** Select the Edit/Copy as Text option to copy the raw data to the clipboard. This is copied in a format that's similar to most text-based Trace route programs. The difference here is that the "Samples to Include" option on the main screen is used to decide how many samples to include in this. If you're sending a graph to someone trying to show them there's a problem in the network someplace, it's probably a good idea to include this data too. Just paste this into your e-mail program or whatever. For best results, use a fixed pitch font (Courier New) like you see here.

Note: If you have "Samples to Include" set to 10 or higher, this menu option will show "Copy as Summary Text" instead of "Copy as Text". Holding down the shift key when you click on the "Edit" menu option will switch between the modes.
3. Comma delimited text file. This is especially built to import into a program like Microsoft Excel - so you can manipulate the numbers around and create output in different formats.

To export to a comma delimited text file, go to the File menu and select "Export to Text File". From there, you can specify the file you want to export to and a couple of other options. You can either export all samples in memory - or the range as specified on the main screen. The option on this screen "Include sample times in export file" will specify whether or not to include the time each sample was taken at. If you don't have this turned on, all the samples will be output, but you won't get corresponding times. Turn this option on to include the times.

We discuss more reporting options in our knowledge base.

3.10 Tracing from a remote server

PingPlotter Pro is able to initiate trace requests from a remote "agent" on both Windows and UNIX based machines. The UNIX agent requires perl, a web server and the traceroute command, whereas the Windows agent is fully self-contained (it has its own TCP server and GUI component) and does not require any additional software or server components.

The basic concept of the remote trace agent is that it resides on a remote machine listening for a request for data. When it receives a request, it does a single traceroute and returns the results back to PingPlotter Pro. PingPlotter Pro then collects this data and displays it locally like any other trace data. The interval at which PingPlotter Pro contacts the remote agent is controlled by the sampling parameters (# of times to trace, Trace interval, etc.) you've defined locally via the PingPlotter Pro interface.

When using a remote agent, PingPlotter works and looks the same as it does when you're using the local engine. Essentially the only difference is that the trace isn't originating from the workstation running PingPlotter Pro, but instead the agent is allowing you to source and see the traceroute from the vantage point of the machine running the remote agent. You can even see the results from multiple remote sources/agents by defining and using Named Configurations.

Example:

Lets say that we have three offices: a main office in Boise, another in Amsterdam, and one in Tokyo. With a single instance of PingPlotter Pro running at our Boise office, we already know that we can monitor the connection between us, and the other two offices. We can also monitor our connection from the Boise office to our business server, and VoIP server.

When using remote agents deployed to both the Amsterdam and Tokyo office - we can now also measure the connection from both of those offices to both the business server, and VoIP server as well - all from the single instance of PingPlotter Pro that's running back at our headquarters in Boise!
Our summary screen in PingPlotter Pro would look something like this:

<table>
<thead>
<tr>
<th>PL%</th>
<th>IP</th>
<th>DNSName</th>
<th>Settings</th>
<th>Avg</th>
<th>Cur</th>
<th>Jtt</th>
<th>Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>80.81.82.83</td>
<td>Amsterdam Office</td>
<td>From Boise</td>
<td>173</td>
<td>179</td>
<td>6.37</td>
<td>0</td>
<td>283</td>
</tr>
<tr>
<td>90.91.92.93</td>
<td>Tokyo Office</td>
<td>From Boise</td>
<td>203</td>
<td>187</td>
<td>12.00</td>
<td>7.4%</td>
<td></td>
</tr>
<tr>
<td>100.50.50.2</td>
<td>My.VoIP.Server.Com</td>
<td>From Boise</td>
<td>120</td>
<td>121</td>
<td>2.39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200.60.60.3</td>
<td>My.Business.Server.com</td>
<td>From Boise</td>
<td>45</td>
<td>43</td>
<td>3.88</td>
<td>15.4%</td>
<td></td>
</tr>
<tr>
<td>200.60.60.3</td>
<td>My.Business.Server.com</td>
<td>From Amsterdam</td>
<td>233</td>
<td>229</td>
<td>6.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200.60.60.3</td>
<td>My.Business.Server.com</td>
<td>From Tokyo</td>
<td>260</td>
<td>273</td>
<td>11.88</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Image generated by PingPlotter Pro 3.41.0p (http://www.pingplotter.com)

Requirements:

- PingPlotter Pro version 3.10.0p or higher, installed.
- A Windows XP or higher workstation (Windows agent), or a UNIX machine with Apache, perl, and appropriate permissions (UNIX agent).
Installation instructions

Installing the Windows Agent

The PingPlotter Pro remote agent for Windows is a self-contained executable that includes a small server and some portions of the PingPlotter core trace engine. This agent can run as an application or as a service under Windows XP or newer.

Note: Presently the Windows agent does not come packaged with an installer or uninstaller.

- Download the Windows remote agent from the agent download page.
- Extract the agent (PingPlotter_Remote_Agent.exe) to a working directory (e.g., c:\Program Files \PingPlotter Remote Agent).
- Browse to the directory where you extracted the agent and launch it. The agent will show up in the system tray.
- Ensure your firewall(s) allow access to the agent on TCP port 7465, or change the server port for the agent. To change the port, as well as other settings for the agent, double-click on the tray icon.
- From your PingPlotter Pro machine, launch your web browser and access the agent at http://(your agent machine name):7465.
- Verify the output. It should look something like this (in your browser):

  PingPlotter Pro Remote Windows Agent V0.8.2.18
  Tracing to:
  Error: Must specify an IP Address

Continue on to the Configuring PingPlotter Pro to use a remote agent section.

The windows agent has a few options, to see about its use, see the configuring the Windows agent section.

Installing the UNIX Agent

The PingPlotter Pro remote agent for UNIX is a perl or php script that uses the UNIX traceroute command to gather sample set(s) and pass the results to PingPlotter Pro via a web server. Because of all the various flavors of UNIX that exist, and the possible differences when it comes to output of the traceroute command, you may need to modify the script for your particular source machine. We've primarily tested under FreeBSD and Debian. Contact us if you have problems here.

- Download the unix agent script from the agent download page.
- Extract the perl script (trace.pl) or php script (trace.php) into a web-accessible directory on a UNIX web server that is capable of serving the appropriate script.
- If necessary, set the script to be executable by your web server, (chmod 755, or similar, and don't forget to check the permissions for the directory).
- Test it from a web browser to make sure it's accessible. Example: http://www.yourwebserver.com/pp_pro_agent/trace.pl (or trace.php if you're using the php version).
- Test a full trace from the web browser. This requires the ?IP= parameter (?ip=parameter will fail, the 'IP' needs to be capitalized). This would look like this to trace to 192.168.1.1: http://
www.yourserver.com/pp_pro_agent/trace.pl?IP=192.168.1.1 Ideally you'll want to use a valid IP address here, of course - one that works on the remote agent’s network.

- Verify the output. If it’s working, you should get output that looks something like this:

PingPlotter remote trace agent, V0.4

1 192.168.1.1 0.684 ms

You’ll probably have more hops than this. This is just an example if you were using an IP local to your server.

- If you have an error, look at the source code for the script for some different alternatives and help.

**Configuring PingPlotter Pro to use a remote agent**

Once the remote server(s) are configured, we need to set up PingPlotter Pro so it knows about those servers.

**Turning on the remote trace capability.**

By default, the remote engine capabilities of PingPlotter Pro are disabled. To enable them:

- Open PingPlotter Pro, go to Edit->Options, Plugins & Scripting.
- Turn "On" the "Trace via Remote Server" script. Hit "OK" to save the settings.

![Options and Plugins](image)

- Restart PingPlotter Pro.

This will enable PingPlotter Pro to do remote agent testing, but you'll need to create a named configuration for each remote agent you want to use (not each target, but just each remote agent machine).

**Creating a named setting (configuration) to use a remote agent.**
Named configurations are covered in more depth in the named configuration section.

- Go back to the options dialog (Edit -> Options).
- Create a new configuration (Right-click on the “Default Settings” in the tree view and select “New Configuration”).

In the newly created configuration, change the name in the main section (e.g., “via <your server name>”).

- Go to the "Engine" section of this new configuration and select "Trace via Remote Server" from the dropdown. (If this engine type is not listed, look at the turning on the remote trace capability section to enable the remote trace engine.)

- In the new section "Remote Server Source Settings", Enter the URL to the trace agent there. For a Windows based agent, the URL will look like this: \texttt{http://yourservername:7465/}. For a UNIX-based server, the URL will look something like this: \texttt{http://yourservername/script_path/trace.pl} (or trace.php, if you're using the php version).
• Hit “OK” to close the options dialog and save your settings.

Usage Instructions

Now, on the main PingPlotter Pro screen you'll see a "Settings" dropdown below the "Address to trace" field. Pick "via (servername) ..." there (or whatever you named your configuration). Enter your target, and then click on the Trace button.

That should access your remote server and populate data into PingPlotter Pro from that.

Warning: Your tendency will be to pick a server very familiar to you, which in a lot of cases is going to be the same server you just set up the script on. If you do this, your trace will only be one hop long!

Error handling and troubleshooting

A remote agent adds additional complexity to any deployment, and PingPlotter's is no exception to this. In an attempt to keep things as simple as possible, and still satisfy our needs, we use existing technologies - web servers, http, existing command-line tools (for UNIX, where good ones exist), etc.

This means that in most cases we're dealing with familiar territory when troubleshooting. All the remote
agent capabilities can be accessed via a web browser. If PingPlotter Pro is having problems reaching a remote agent, then you can use your web browser to check and verify. You can also use PingPlotter Pro (not the remote agent) to constantly monitor the availability of the server and route to the remote agent.

If the remote agent is inaccessible for some reason, there is a network failure between the PingPlotter Pro machine and the agent for example, any alerts you have configured will not fire. If you want an alert to fire, you need to set up PingPlotter Pro to monitor the agent machine itself.

There are a number of configuration challenges that might hit you here, but these should be familiar to you if you've done much network and/or application troubleshooting. ZoneAlarm, for example, might block access between PingPlotter Pro and the agent machine. Other firewall issues could also come up. If you view the remote agent as a web server, the troubleshooting techniques you've used elsewhere should help you here too.

If PingPlotter Pro is unable to reach the agent, because of an authentication problem or network failure, the error will appear above the trace window on the targets it applies to. This message might be an HTML error (e.g., 401 authentication failed), or a network failure message. These events will show up in PingPlotter Pro's time graph the same way a route change does, with the background color on the time graph matching background colors elsewhere rather than having the graph scale colors painted on.

Notes, thoughts, shortcomings...

The Windows agent uses a fully threaded engine to make working with it just as responsive as working with the local engine (if there is a timeout, other hops will still report back). For the UNIX agent, though, if hop 1 times out hop 2 won't report back until the timeout period has elapsed. This means there may be times, when the trace interval is < 5 seconds, where you see empty spots in the data that later fill in with a lost packet or a good sample. Everything works, just the results get delayed a bit. The Windows agent shouldn't have this problem.

Configuring the Windows agent

Because the Windows agent is a full-fledged server unto itself, you may want to have it behave differently depending on where you're installing it.

Server Port

The default server port of the Windows agent is 7465. If this doesn't work for some reason in your network, you can change it by right-clicking on the tray icon and selecting the "Settings..." option. Clicking on the "Apply" button makes that change immediately, and if the server is unable to start on that port an error will occur.

Security

You might not want to expose the agent to anyone or limit access to it. This can be done via the username / password settings in both the agent and the remote trace setup in PingPlotter Pro. The authentication system uses standard HTTP basic authentication. If the agent requires a username / password, and it doesn't match what's specified in PingPlotter Pro, an error will show up in PingPlotter Pro. If no username or password is required by the agent, then the username / password settings in PingPlotter Pro will be ignored.

Running the agent as a service

If you want to be able to trace from a machine at any time, you can install the agent as a Windows Service.

To install as a service, browse to the agent's install directory using a command prompt and then enter the following command:

```
PingPlotter_Remote_Agent /install
```
After doing this the agent will show up in the control panel services applet, or you can start the service manually with this command: `net start PP_RemoteAgent`

To uninstall the service (remember that because there is no installer / uninstaller for now, you'll need to manually remove it), stop the service, then run this command:

```
PingPlotter_Remote_Agent /uninstall
```

You can also use the "/reinstall" parameter to register the agent in a new location.

If any of these commands have an error, then you'll get a popup message telling you about that error. If you have no errors, then the command will complete with no popup messages. If you do get an error, you can check in the "PP_Remote_Agent_error.log" file to see what the error was.

Note that under Windows Vista and newer, you will not have access to the user interface when running as a service. For best results, stop the agent, run it as an application to change its settings and do testing. When you have everything working the way you expect, close the application and restart the service.

### Configuring the UNIX agent

The UNIX agent is much simpler (more primitive?). It's a perl or php script that runs the traceroute command and shows the output on a web client. Because it uses an existing web server, all the settings need to be configured on that server including the server port and security.

**Security**

Because PingPlotter Pro uses standard basic authentication to secure access to the agent, you just need to password protect the agent application using an .htaccess security setting (if you're using apache, which most people are). PingPlotter Pro will send the username and password to the web server using this protocol, and it should all just work.

As always, please [contact us with questions](#).

### 3.11 Workspaces

PingPlotter Pro workspaces allow you to save targets, data and settings and later reload this workspace's targets, data and settings.

A workspace file has a .pws extension. If collected data is saved with the workspace, then a corresponding directory is also created. If your workspace is called "MyWorkspace.pws", then a directory will also be created called "MyWorkspace". This will hold all the collected data.

A workspace (in PingPlotter Pro V3.30p and higher) captures the following information:

- A list of all targets being traced
- The configuration name that each target is using
- Trace Interval; Samples to Include; Maximum Trace Count
- Column Widths; Time Graph Heights & Time Period
- Whether a target is tracing or paused
- Window positions, sizes; tab docking status
- Instance-specific alert associations
- Summary Graph visible targets
- Collected Data (if that option is selected).
When PingPlotter is first launched, the workspace it uses is unnamed until you save the workspace. Once the workspace has been saved, it can then be configured to autosave / load when PingPlotter Pro opens / closes, auto-save based on a time interval, in addition to some of the target-specific alert configurations.

An open workspace is **locked** so that another instance of PingPlotter Pro won't open it, and get confused about who "owns" the workspace. Workspaces can be opened read-only if opened elsewhere.

If saving data with a workspace, collected data is saved in a directory named the same as the workspace. For example, if you have a workspace called "Company Network Status.pws", then all collected data for that workspace is saved in a directory called "Company Network Status", in the same directory where the "Company Network Status.pws" file resides.
Part IV

Common Tasks
4 Common Tasks

4.1 Getting notified of a network problem (alerts)

4.1.1 Creating / Configuring Alerts

What is an alert?

Alerts basically monitor the conditions of a specific IP address, and then do something when those conditions exceed a specific range. The things you can do with an alert are:

- Send an email
- Play a sound or .WAV file
- Log to a text file
- Change the tray icon / show a message
- Launch an executable
- Your own alert event type (Pro only)

For example, let's say you need to know when a destination you're monitoring stops responding. You can attach an alert within PingPlotter to that IP address so that you receive an email alert if the last 10 of 10 sample requests are lost.

Another possible alert condition to check for is if the average for the last 10 samples is > 500 (or any other number). You can send an email alert, maybe play a .WAV file (if you're usually within hearing distance) or both. Also, if you're trying to show your ISP there's a problem, you might log the data to a file so you have records of every time it happened over a time period.

One user had hardware problems with his cable modem. He set up an alert that launched an executable that then communicated with a device attached to his computer to reboot the cable modem when alert conditions were met. There are a whole variety of things you can do with alerts and events.

Alert setup can at first seem confusing, but it involves these four steps:

1. Setting the Alert Name
2. Setting up the Condition that will trigger the alert.
3. Specifying the Event(s), or what you want PingPlotter to do when your alert(s) are triggered.
4. Tying the alert to a target that you're monitoring.

What is an event?

Events go hand in hand with alerts. Any number of events can be created with an alert — so that when some conditions occur, something happens. The "Alert" part specifies the conditions and serves as a container for the events. The "Events" specify what action(s) will happen. You can have any number of events tied to an alert, and any number of alerts tied to a host. Note: You must have an alert tied to an IP address before it will work. Keep reading for more details.
Alert Dialog / Settings

Alerts are defined in the Alert Setup screen under the Edit/Alert Setup menu. Once there, you'll see an image similar to the one above. (Note: if you haven't setup an alert before, you won't see any alerts listed in the list area on the far left of the screen shot). This screen is pretty busy, but also pretty self-explanatory. From the screen capture above, you can see that we have (labeled on the image with corresponding numbers in this list):

1. The Alert Name. In this example we're using "Destination is Over 100ms".
2. The list of alerts you currently have defined. You can see from the image that we have two defined.
3. The Conditions for the alert. In other words, what has to happen for this alert to fire.
4. The Show Targets button that allows us to see how many targets/hops this alert is assigned to, as well as the IP address or DNS name for that hop or hops. If this alert is set to watch 0 targets, then the alert will not fire! We discuss this in more detail later in this topic.
5. The Notification/Trigger Events for this alert. An unlimited number of events can be tied to an alert, but most likely you won't use more than four or five. Please see the Event Notifications/Trigger Notifications section for all of the trigger notifications available.
6. Each Event/Trigger Notification will allow you to "do something", based on the overall type of alert you're defining. For example, with a Tray Icon Alert you define the message the pop-up will contain, a Send an Email Alert needs to know where to send the email and with what frequency to send, a Play a Sound Alert needs to know what to play, etc. You enter that information here under the trigger event.
7. (These options are only for the Send an Email Event type). The Edit Body button will take you into the email Alert Email Body Editor, where you can fine tune the alert message body for optimal viewing on a cellphone, pager or other type of handheld device. By clicking this button, you can verify that your email settings are correct, and what your alert email will look like.

8. To define a new alert you click on the New button. To delete an alert, highlight that alert in the list above these buttons and click on the Delete button. When you’re done defining the alert click the OK button. Cancel exits this screen without saving anything.

So referring to the image above, you can see we have an an alert with a Condition defined where when the last 10 samples are greater than 100ms, we want PingPlotter to email us. We've called this alert "Destination is Over 100ms ". If you were watching for a timeout, you'd enter 9999 instead of 100. Do note that 9999 is not a magic number. A lost packet is always greater than any number entered, so you can use 1000, or 20000 here and a dropped packet will exceed either of those numbers.

**Setting up an alert.**

It's usually best to start out with an alert and event that's easy to verify - like playing a sound, or changing the tray icon. So, let's set up the alert to play a sound when packet loss goes about 40%. Packet loss is a big bad thing, and in many cases is more important than latency. We run a similar alert at Nessoft HQ pretty much continuously.

- Run PingPlotter and go to the "Edit/Alert Setup" menu. If you've never set up an alert, you won't have any listed here.
- Hit the "New" button (near the bottom left).
- Enter a name for the alert (We're using "Server is Down") in the "Alert Name" section.
- In traces to examine, use "10". Set it to alert when 4 or more samples are over 9999. (Note: 9999 is not a magic number. A lost packet is always greater than any number entered, so you can use 1000, or 20000 here and a dropped packet will exceed either of those numbers). That setting will alert when 40% packet loss is achieved. The 40% comes from saying to alert if 4 of the last 10 samples were dropped. You might want to do 40 of 100, or 2 of 5 – it depends on the period of time you want it to look at. You probably *don't* want to notify if only one sample is lost, but maybe 2 in a row (which would be examining 2, and alerting if 2 or more traces over 9999). Lots of things you can do here.

**Alerting on packet loss.**

A common condition to want to alert on is packet loss. The fields you need to manipulate are in the Alert setup screen is the "Alert conditions" portion.

Example: Let's say you want to notify when packet loss equals or exceeds 40%.

To do this, set "Samples to Examine" to 10, and Alert when "4" or more samples are over 9999ms. A lost packet always exceeds any number you enter in the threshold area, so if you want to consider only explicitly lost packets, set this to 9999. If you want to consider any really high latency packets as well, set this to something lower (maybe 1000 or 1500).

This only examines the last 10 packets, but let's say you want to examine a higher period - and notify on a lower packet loss percentage.

Set Samples to examine to 10000 (or some other high number). Alert when "500" or more samples are over 2500ms. This will alert when you hit 5% packet loss over a period of a few hours (depending on what trace interval you use).
Picking an event.

Next, select what you want to happen when the alert fires. We're going to "Play a sound" (This is in the Event 1 area). Notice that as soon as you change the "Event type" to "Play a sound", that "Event 2" will appear, with "(.. no additional notification ..)". You can have as many events as you like – and to delete an event, just change its type to "(.. no additional notification ..)".

As soon as you pick an event type, a set of options relating to that event type will come up. The "Notify" dropdown is described in more detail elsewhere. For more details on how to set up a sound event, click here.

For this example, use "Notify: each time alert conditions are met (repeating)" and then hit the folder icon beside the "Play Sound:" edit field. Browse to an appropriate sound file (.wav or otherwise) on your computer to fill this box.

Using this event, each time a sample is sent out and the alert conditions are met, the selected sound will play.

Tying an alert to an IP.

Once you've got your alert set up, you need to tell it which IPs to "watch" for those conditions. To do this, trace to a destination just like you normally would. Once the path has listed, pick the router/destination from the list you want to monitor - and right-click on the hop number. From the popup, select "Watch this host (Alerts)". If you want to monitor a destination that isn't responding for some reason, just right-click on the lower time-graph for that host, and you'll have a similar popup menu.

You'll get a dialog that shows the DNS name (if there is one) and the IP address you selected - and then a list of available and selected alerts. Move the alerts you want applied to this IP address into the "Selected" list - and then close the dialog. Monitoring will start immediately.

When a hop is being monitored for an alert, the hop number will have brackets around it (i.e.: [10]). You can stop monitoring by right clicking on the hop number and selecting "Watch this host (Alerts)" and then removing any alerts from the selected list.

Any time an alert has fired, PingPlotter will let you know by putting a red exclamation point (!) next to a hop that has had an alert fire. This is particularly handy when you have an alert that doesn't give you visual or audio cues normally when it fires, like a Send an Email Alert. Now you can see it on the main PingPlotter screen before the alert email gets to your inbox.

Global vs Local alert associations

Version 3.30 of PingPlotter Pro introduces the concept of target-local alert associations. Previously, only global associations existed.

Global associations

Global associations mean that any time that router participates in a route, an alert will be started. When tracing to multiple targets and associating alerts with final destinations, this works great - you normally don't trace to the same destination twice. Having a global association with a target means that once you've made a decision to alert on a target, those alerts will be associated with that target any time you trace to it.
This doesn't work so well, though, when you want to alert on an intermediate hop and that intermediate hop/router appears in multiple routes. In that case, the alert works double (or triple, or whatever) duty, and you get too many notifications. Here's where Local associations make sense.

**Local associations**

Local associations only work within the context of a specific trace session. If you trace to a target and set up a local association between an alert and a target, you'll get alerted just fine. If, however, you trace to that same target in another tab of PingPlotter, the alert won't be created again.

This relies on a specific workspace to remember your setup. If you close and restart PingPlotter and manually trace to that target again, you won't be alerted. To have this alert association be remembered, you need to save the workspace, and then reload it later. The alert association will be remembered inside that tab of the workspace. The alert association is remembered within the tab, and if you stop/start/reset it will continue to work. If you trace in a new tab, you'll need to associate the alert with the target again.

Any alert can be associated globally or locally, or a combination. The link between the target and the alert controls the scope.

**Editing the body of an email**

The Alert Email Body editor allows you to fine tune the body of the alert email to whatever makes sense for your particular receiving device.

For example, if you have alert emails going to your cellphone's inbox, you may only want to have the Target Name, First instance of Failure and Last Instance of Failure in the email. If you have any more text than that, it will be too long, your cellphone will truncate it and the text is pretty much useless to you. We recognized the fact that a lot of people are using RIM devices, pagers or cellphones to receive
alert email, and so now we have an editor to help you out.

You can customize the email body text to your liking in the left pane's editor, highlighting attributes on the left and clicking the **insert Selected Item** button to enter that attribute into the body text to the left. This ability to highlight and insert the alert specific attributes is a great way to avoid typographical errors, and is highly recommended. When you're done with your changes, just click the **Save** button to save your new alert email body. If you get to the point where you want to start over again with the original email, just click the **Revert to default email** button and then start your changes anew. If you change your mind about editing the email body at all, just click the **Cancel** button to go back to the Alert Editor.

A note about "average" response times.

**Average** response times are a problem. The real problem with **mean** averages is when a server stops responding - what is the average of the last 10 samples if the last 10 were timeouts? Because of the problem with this we always do "when X or more samples is > Y" (this is a **median** average). You can still get your alert to work like an average - by saying "when 5 or 10 samples exceeds 300 ms" (this would be like a mean average over 300ms, but would also fire when there were lost packets).

4.1.2 Troubleshooting Alerts

If alerts aren't working, there are a number of things you can do to troubleshoot. Here are some suggestions - and feel free to check out our [support pages](#) for more information.

**Make sure you're "Watching" a host with the alert.**

By far, the most common reason that an alert isn't working is because it isn't tied to an IP address. An alert won't just start working automatically – you need to tell it which host(s) you want that alert to watch.

To attach an alert to an IP, trace to the host you're interested in. Then, right-click on the hop you want to monitor and select "Watch this host (Alerts)..." from the menu.

Note that if you're tracing to a destination that doesn't respond, and you want to watch that destination (even though it's not on the upper graph), just right-click on the time-graph on the bottom and this same menu item should be on that menu.

From here, you can move an alert from the "Available" to the "Selected" list. Any alerts on the "Selected" list will watch this host whenever this host is involved in any route (be it an intermediate host, or the final destination).

When a host is being watched by the alert system, there will be brackets around the hop number in the upper graph. If those brackets aren't showing, then that host isn't being watched. This should put a [...] around the hop that's being monitored.

**Set up an alert that will fire instantly, with an event that is very evident.**

If you have an alert set up – and tied to a host (see above), but it seems like the alert isn't working, then changing your alert parameters (or create a new "test" alert).

Set up "Traces to Examine:" to 10. Alert when "1" or more traces are over 1ms.

Unless your network is responding in 1ms or less, this alert will fire on the first collected sample with the alert enabled.

For an event type, use "Play a sound", or "Tray icon change/notification" as both of these events happen immediately with no wait. In addition, for the "Play a sound", use "each time alert conditions are met (repeating)“, as this will continuously make sound, rather than just when conditions start / stop.

Using this sequence, you should be able to tie an alert to just about any host and have the alert conditions fire immediately. Now, add on another event type (ie: email).

You can leave multiple events tied to a single alert – that way you can continue to hear the sound while you're
troubleshooting another event type.

4.1.3 Email Setup (for alerts)

The email setup dialog is used to set up emailing for alerts. If you're not using alerts, or you're not interested in having the alert system email you, then setting this up is not required.

Return Address

All outgoing emails will have a return address specified, and this is the address that is used. Please make sure you specify a valid address here since this is where all the bounce messages will come from. Some ISP SMTP servers only allow emails sent out with a "from" address of their domain as well, so if you're having problems getting the SMTP server to work, make sure you're using a valid return address.

Attach PingPlotter Savefile (.pp2) on Email Alert

When an alert goes out, data will be attached. The data that is attached encompasses all the alert period since the "last" similar alert was sent out. This is a global setting for all email alerts. Note that you'll need PingPlotter installed on a machine to read the attached data.

Include "alert" samples in text message

If this option is checked, any outgoing alert emails will include text showing the samples that failed the alert. As each alert email goes out, the past history is marked as being sent already so that you won't get duplicate data in reoccurring emails.

SMTP Server

The SMTP server is the server that your outgoing mails will go through. This may have been given to you by your ISP or your mail administrator.

Server Port

The default port for most SMTP servers is 25. If you connect to your SMTP server via a different port, then enter that port here. Leaving this blank will use port 25.

SMTP Authentication

Some SMTP servers require a username and password to be able to deliver mail. If this is the case with your server, turn on the "Use SMTP Authentication" checkbox, and then enter your username and password. The password is saved in your MultiPing.ini file using a basic XOR encryption scheme – this will keep your password hidden, but this encryption method is relatively simple to "crack" if someone really wants to figure it out.

Use STARTTLS encryption if server supports it.

Most current SMTP servers prefer or demand the use of a secure channel to protect your username and password. If you use one of these servers (GMail is a great example of one), then you'll need to turn on STARTTLS. For more details on this topic, see our knowledge base article.

4.1.4 Alert Events

4.1.4.1 Event Notification

Many of the events share a notification mechanism. Here is a list of the types. Note that any alert can have multiple events of the same type, so you can set up a single alert to do something at any one or all of these times.

Each time alert conditions are met (repeating)

The event will happen every time conditions are met. This means the event will happen over and over
again – on each sample that causes the alert to fire. In previous versions, this was the only supported notification type.

**When alert conditions start (enters alert state)**

The first time alert conditions occur, the event will happen. As long as the conditions continue, though, the event won't be repeated. This is a popular use – as you find out about new conditions when they happen, but don't have to be bothered again. As soon as the alert stops happening, then as soon as it starts again, this event will happen again.

**When alert conditions end (leaves alert state)**

This happens when network conditions improve so that the alert is no longer firing. As soon as the conditions move from bad to good (based on your settings), then this event will happen. A use of this is to have PingPlotter email you each time conditions go bad (see above), and then when they improve again – but not to tell you anything in between.

**Each time alert conditions are *not* met**

This is the exact opposite of the first notification type above. As long as things are good on the network, this event will fire each time a sample is collected.

4.1.4.2 Event - Play a Sound

One of the most simple event types is to play a sound (i.e.: .wav file) of some kind.

This event can happen based on the standard notification rules and can play anything that Windows multimedia sound function wants to play. If you want to launch a sound file that this event type doesn't support, use the "Launch an executable" option instead, as it will launch any file, including sound files.

Click the folder on the right side of the file name to browse for a file. When browsing for files, you can right-click on any sound file to play it (this is an operating system feature, and may not be supported on all operating systems).

Enter "BEEP" (no quotes) here if you just want to beep the computer speaker instead of playing through the sound card.

4.1.4.3 Event - Send Email

A very popular event type is the "Send Email" event.

Before you can create an event to send an email, you must configure your SMTP server and return email address. Note that your SMTP server must be accessible on the network to be able to send emails, so it's possible a network failure may not be able to email you. PingPlotter will continue to try to send emails once a minute until it is able to get an email out.

Emails are a bit more complicated to set up than most Event types – as it is dependant on your SMTP server, and you don't want to be overwhelmed with emails when conditions are bad, but you "do" want to know what's going on.

First off, you can fire emails based on the standard PingPlotter notification types. See the associated documentation for more details.

Send e-mail to:

This can be an individual email address, or a list of addresses separated by either a , or a ; (both work equally well). Please do not set this up to be someone at your ISP unless they have agreed that they
want to see this information. A huge portion of getting problems solved is playing the game right, and overwhelming people with automated emails is almost certainly going to work against you.

Email Subject:

This defaults to “PingPlotter Auto-Alert!”, but can be customized with a variety of variables / text. [Click here to see this list][33].

For emails, the $host makes a huge amount of sense (i.e.: $host is down!), while the time/date aren’t as useful because the email already contains data about this in most cases.

Maximum email frequency and minutes to wait

The next two settings control the frequency at which you’ll get e-mails during alert conditions.

We recommend settings both of these values to 0 (no delay) when using a “When alert conditions start” and “When alert conditions end” event types.

When using a “Every sample” event type, though, this will result in massive overload of emails, so you’ll want to limit the number of emails sent.

The first (Maximum e-mail frequency) means that you’ll only get emails that often for this alert/target/event combination. Once you get an alert email, you won’t get *another* one until this amount of time has passed. If conditions call for an email to be sent, it’s delayed until this amount of time has passed. Beware that conditions may have changed between the time the alert is “queued” (and delayed) and the point where the alert email is sent out (because it’s been delayed by this setting). This can be confusing (which is why we recommend setting this to 0 on alert conditions start and alert conditions end events).

The second (How long to wait for worse conditions) specifies how long PingPlotter will wait after its first alert condition to send an e-mail. This option allows you to wait a few minutes to find out if it was a temporary or more permanent alert condition. You may not want one immediately - because you’ll want to wait a bit more info to be included - so you may want to wait 5 minutes or so before that first e-mail gets sent off.

Testing and error messages

Once you have your e-mail set up, use the “test” button to see what the message will look like (and also to make sure all the settings are working). Any errors should be displayed here.

Many of the errors that occur during testing can be attributed to incorrect email setup – so go there and validate your settings[34].

Here are some specific knowledge base articles on possible error numbers:

Socket error 10053

Socket error 10060

Socket error 11004

Any Socket error is being generated by the SMTP server itself, not PingPlotter, so if you’re getting an error number not listed here, or in our knowledge base, try doing a search on your favorite search engine to see if you can find more information about the error you’re getting.

4.1.4.4 Event - Launch an executable

This event type gives you all kinds of capabilities to do things when network conditions go south.

While this option is called "Launch an executable", it can actually launch any file. It can launch documents, links, .mp3s, batch files, whatever – really anything with a file association that Windows will
know what to do with.

Of course, you can control when you want to launch the executable (when the problem starts? when it ends?). See the associated documentation for more details.

The filename to launch (and/or parameters for that filename) can use variables to pass an IP Address, name and date/time to the called program. Note that these are always parsed – and there is no way to cause these to be passed as literals. If you need to have one of these strings as literals passed to an executable, then you'll need to set up an intermediate link, batch file, or similar.

Note: The launching program isn't closed – it's just launched. You'll need to configure your setup to do any appropriate follow-on actions.

### 4.1.4.5 Event - Log to File

The "Log to file" alert writes data to a text file whenever alert conditions are met.

#### Log times for entire route?

This option specifies if you want to write data to the text file just for the monitored host, or the entire route. Leaving this off means that for each time alert conditions are met, one data item will be written to the file – for the monitored host. If the switch is turned on, then data for the entire route will be written.

#### Filename:

The filename is *required* to have the $host variable in it (or some variable that changes depending on which host / target is being logged). If it is missing, then the file will be nonsensical if you attach this alert to more than one host. The following variables can be used as part of the filename. Note there is no way to "escape" the following sequences, so these are always parsed and can't be specified as literals in the filename.

- **$dest**
  - The target destination's name (or IP Address if the IP didn't resolve to a name). This is different than $host because an alert might be attached to an intermediate hop, whose information is accessed via the $host variable(s). The final destination, though, is accessed via the $dest variable(s).

- **$destip**
  - The destination IP address

- **$destname**
  - The resolved DNS name for the destination

- **$config**
  - The name of the configuration used when tracing the target in question. This is useful when you're using multiple configurations and want to be able to distinguish between them and the target name / IP isn't enough. The $settings name is also equivalent to $config - either can be used interchangeably.

- **$host**
  - Host name (or IP address, if no name exists). If used in an alert on a non-target hop, the alerting hop's information will be used.

- **$hostip**
  - The host IP address

- **$hostdnsname**
  - The resolved DNS name for the host

- **$year**
  - Current Year

- **$month**
  - Current Month

- **$day**
  - Current day of month

- **$hour**
  - Current hour
Starting with PingPlotter 3.40, directories will be created to any depth needed to satisfy your file name (prior to this version, only one level of depth was created). If you specify c:\ppdata\$year\$month\$day\$host\$host data for $day.log, then each directory level will be created.

4.1.4.6 Event - Tray Icon Change

This is a great event to add to most of your alerts. It's helpful to be able to see if there's an alert condition under way, and a quick glance at the tray can let you know by using the event.

The Tray Icon Change notification will do one or both of the following:

**Change default icon to red during alert conditions**

If you already have PingPlotter showing in the tray, this will change the existing green icon and add red to indicate that an alert has fired. If you don't have PingPlotter in the tray already, then a red icon will be added to the tray. When the alert condition(s) are over, the icon will change back to green.

**Popup message in tray**

This shows a “balloon” message coming out of the tray. Not all versions of Windows support this message (ie: some versions of Windows 95), in which case no balloon will show. Only one balloon can be shown at a time, so the newest balloon always wins (a new balloon message will replace an older one).

Variable substitution will be done on your message text.

One of our favorite messages is: Alert fired on $host, $date $time.

This is a great message because it stays up until you acknowledge it, so when you come back to a PC, you can see what alert fired, what host it happened on, and what time it "last" happened. This may be over a weekend, but the message will still be there telling you that an alert happened.

4.1.5 Variable Substitution

A number of the alert parameters allow you to insert a variable which will be substituted for a value when the alert happens. Here is a list.

- **$dest** The target destination's name (or IP Address if the IP didn't resolve to a name). This is different than $host because an alert might be attached to an intermediate hop, whose information is accessed via the $host variable(s). The final destination, though, is accessed via the $dest variable(s).

- **$destip** The destination IP address

- **$destname** The resolved DNS name for the destination

- **$config** The name of the configuration used when tracing the target in question. This is
useful when you're using multiple configurations and want to be able to distinguish between them and the target name / IP isn't enough. The $settings name is also equivalent to $config - either can be used interchangeably.

$host Host name (or IP address, if no name exists). If used in an alert on a non-target hop, the alerting hop's information will be used.

$hostip The host IP address

$hostdnsname The resolved DNS name for the host

$year Current Year

$month Current Month

$day Current day of month

$hour Current hour

$minute Current minute

$second Current second

$date Same as $year-$month-$day.

$time The same as $hour$minute - note the absence of any punctuation - that's to make sure the file name is valid, if this mask is being used in a file name.

4.2 Running as a service

This documentation covers PingPlotter Pro 3.40p and higher only. If you're using an older version, see the documentation that came with that version or upgrade to the latest version.

If you normally monitor the same set of targets and you *always* want to monitor them (and possibly alert on them), you might want to set up PingPlotter Pro to run as a Windows service. Once you've done this, PingPlotter will start running when you boot your machine and will keep running - even if you log in as a user and then log out.

Important concept about running as a service

When running as a Windows Service on Windows Vista, Windows 2008, Windows 7 and newer, the user interface of PingPlotter is running on Session 0. To edit settings on the service instance, you should interact with the Session 0 instance - you should not start a new instance of PingPlotter Pro to make changes.

If you are running Windows XP or 2003, PingPlotter Pro will appear in the tray, but that for other more recent operating systems, you should use the "Show Windows Service" shortcut, rather than starting a new instance. If logging in via Remote Desktop to 2003, you'll need to tweak a setting, covered here.

Installing PingPlotter Pro as a service.

To set up PingPlotter to run as a service, enable that option when installing. If you want to disable the service or re-enable the service, do that by modifying the options - either by re-launching the installer, or by using the "Modify" option in the Add/Remove Programs control panel area of Windows.

Interacting with PingPlotter Pro as a service
Once you've configured PingPlotter Pro to run as a service, it should always be running as long as your computer is turned on. It will automatically start as soon as the installer completes.

To make changes to the configuration, add targets, view status or any other user interface options of PingPlotter Pro, you can access the user interface by this method:

**If you're running Windows XP or 2003**

PingPlotter Pro should show up as an icon in your tool tray. Double-click that to bring up the user interface. If you're using Windows 2003 and logged in via remote desktop, you'll need to tweak a setting, covered here.

**If you're running Windows Vista, 2008, 7, 8, 2012 or higher**

You should have a shortcut in your PingPlotter Pro program group, called "Show PingPlotter Service". This will switch you to Session 0, where you'll be able to interact with the service instance of PingPlotter Pro. If this shortcut doesn't seem to do anything, make sure the "Interactive Services Detection" service is running (Windows control panel, Services - start it and make sure Startup Type is set to "Automatic").

**Creating a workspace**

Start out by creating a "workspace" of targets you'd like to normally trace. To do this, use PingPlotter Pro to trace to the targets you want to monitor, setting up your Trace Interval, engine settings, etc. Then, use the **Workspace** -> **Save Workspace** menu option to save these settings into a workspace file.

This workspace records all the destinations, trace interval, window positions time-graph interval and a variety of other settings. What it does **not** save is the actual engine parameters. Instead, it saves the name of the configuration you're using, and then uses the current setting for that named configuration.

**Configuring the service options**

There are a few options that should be set to make running as a service as easy as possible. For more details on these options, see the help topic that covers this.

In "Workspace to load", browse to the .pws file you created earlier (or type in the path here). Note that this should be an absolute path - with a drive letter. It should not point to a network share or a mapped drive letter, as these might not be valid when the computer first boots. Mapped drive letters won't work at all.

If you turn on the option to "Automatically update workspace on shutdown / reboot", then any changes made to the list of targets while running will automatically be written back to the workspace file so the next reboot will match that.

Another important setting is "Save File Location". This is where PingPlotter Pro writes collected data. This must have read-write access to the System user, and should be an absolute path as well. You can leave it blank to use the PingPlotter install directory, but this is not as reliable as specifying a fixed location (example c:\ppdata\service files).

**When is data saved?**

Most people run PingPlotter Pro as a service when they are collecting data full-time.

At a minimum, you want PingPlotter Pro to save your collected data when you shut down, and reload it when you restart. This is how PingPlotter Pro works, and the "Save File Location" is where these data files are stored.

But what if you lose power to your computer, or your computer crashes for some reason. Do you lose all your data? In an attempt to make this less risky, the PingPlotter service configuration automatically
makes a duplicate copy every time you auto-save your data. So if you set up a 30 minute auto-save interval, then the most data you'll lose is 30 minutes, if the computer crashes.

You'll probably want to set a maximum memory size in your auto-save configuration as well - so you don't exhaust your computers memory after a few weeks of collecting data. We discuss some best practices for this in our knowledge base.

General Caveats

There are some challenges and shortcomings that come when PingPlotter Pro is running as a service. Most of these problems come from the fact that you're using a piece of software that is running as the System user, but you're interacting with a desktop that is *not* the system user (but, instead, your own user account).

- When opening a save or load dialog, you might run into odd visibility or user rights issues. When using any standard PingPlotter Pro dialogs, PingPlotter Pro tries to "impersonate" the logged in user, but this sometimes doesn't work right.

- Auto-save, startup and shutdown actions need to have the proper user rights. Since PingPlotter Pro starts before anyone is logged in when your reboot, any action that PingPlotter Pro might do needs to be accessible by the "System" user account. This includes Auto-saving of images and data, Service workspace, and anything else that might write to a file.

- If PingPlotter Pro is running on a Windows Terminal Server, logging in as a standard user session will not show the tray icon for PingPlotter Pro. See here for details on how to solve that problem.

- The service instance won't have access to your user's license information, so you may need to re-enter your license key in the service instance.

- If you want to keep the Session 0 instance from timing out (on some versions of Windows, the timeout is hard-coded to 60 seconds), you can install an extension to keep the session from timing out.

- When accessing Session 0, you won't be able to copy and paste from your normal user session. This means you may have to manually key in your license key.

4.3 Exporting data for further analysis

Exporting data is done via the File -> Export to text file... menu option. This allows you to take all of the data currently in memory in PingPlotter Pro and save it to a text file which can then be loaded into Excel or a similar tool.

We're going to briefly discuss the options here. If you want more information and an example Microsoft Excel spreadsheet, check our knowledge base (in particular, this topic and this topic).
Include Sample Times in export file
If left unchecked, the times for each trace don't get saved - only the data.

Samples to Export
Select All Samples if you want to export your whole trace to text. Select Current Sample Set if all you want to save is the currently displayed sample set shown on the Time Line graph. The current sample set is the setting you'd use if you're wanting to email trace data to an ISP, etc. - though just saving a graph would be a better option.

Export Format
PingPlotter gives you two different Export Formats to save your data in. Both are shown below. If you're using Excel, the second "1 column per hop" usually works best.

Examples:

One row per hop, one column per sample

```
1,,N/A,N/A,N/A,N/A,N/A,N/A
2,loop1.bois-dsl-gw1.bois.uswest.net,216.161.136.254,10,10,10,10,10,10,10
3,100.fa2-0.bois-agw1.bois.uswest.net,207.108.229.29,30,20,10,10,10,10,10
4,--------------,207.108.224.247,20,10,20,10,20,10,10
(....middle data snipped for brevity....)
16,192.ATM7-0.GWS.SJC1.ALTER.NET,152.63.54.21,60,50,50,50,50,50,50
17,digexoc12-gw.customer.alter.net,157.130.214.154,50,50,50,50,50,50,50
18,gsr-01-p2-0-a00a02.af.sjc5.digex.com,164.109.130.26,50,50,50,50,50,50,50
19,lcl.com,164.109.154.154,50,50,50,50,50,50,50
```

One column per hop, one row per sample

```
Host Information
1, ,
2,loop1.bois-dal-gw1.bois.uswest.net,216.161.136.254
3,100.fa2-0.bois-agw1.bois.uswest.net,207.108.229.29
4,--------------,207.108.224.247
(....middle hosts snipped for brevity....)
```
4.4 Long term monitoring and auto-saving data

If you run PingPlotter 24 hours a day, 7 days a week, you're going to want to save your data (in case of power failure or other event), and you're also going to want to limit the memory footprint for PingPlotter. We talk about some best practices and recommendations here, although your needs may be different. For example, if you're using PingPlotter Pro and monitoring a lot of targets, you'll probably want to keep fewer samples in memory for each target, otherwise the footprint of PingPlotter will be pretty big.

Setting up your memory footprint

PingPlotter defaults to keep 250,000 samples in memory. If you regularly do long term monitoring, though, you may want to understand this number so you can change it to fit your needs. In particular, if you keep too many samples in memory, you may run out of system memory at some point.

- Determine how often you want to sample. 2.5 seconds gives a good amount of accuracy without too much data. Some people use 10 seconds. Anything much longer and you might miss problems.

- We find that having 4 to 5 days of data in memory at a time works well. There are 86,400 seconds in a day, 432,000 seconds in 5 days. Divide this 432,000 by your trace interval. For 2.5 seconds, this gives us 172800 samples in 5 days.

- In PingPlotter, go to Edit -> Options, Auto-Save section. Enter your calculated number in "Maximum samples to hold in memory". 172800 samples takes up roughly 10 to 15 megs of RAM in memory, which puts the PingPlotter memory footprint around 40 megs total (it keeps multiple copies of the data in RAM at some points, and general overhead). This is workable for just about any workstation, unless you're tracing to more than a handful of targets.

- If you're using multiple configurations, make sure you review the proper settings for each named configuration.

Setting up PingPlotter to save data

With 4 to 5 days of data in memory, each save of data will have all of this, which puts each save file around 1 to 3 megabytes. Having one file per day gives you easy access to a day's data, along with the previous 4 days for good analysis. We suggest saving every 30 to 60 minutes, with a filename like this: c:\ppdata\$dest\$dest $date.pp2

- In PingPlotter, go to Edit -> Options, Auto-save section.
- Turn on "Auto-save data" 
- Set "Save Interval" for "30 minutes"
• Set filename to "c:\ppdata\$dest\$dest $date.pp2". If you're currently tracing to a target, floating over the filename field will show you a hint of the file that would be actually saved. **Note: It's very important that you specify an absolute path for your save file name!** If you're running as a Windows service and you get save files in your Win32\System directory (or some other unexpected directory), that's because you didn't set an absolute path here.

We set up "30 minutes" for a save interval. The filename controls how often we create a new save file. If the file already exists, it will be overwritten. You can include $hour in the save file name to get a new file every hour, but we don't recommend this for save data, since you'll get a new save file every hour and you may run out of hard drive space.

This will give you a new file each day with 5 days of data in it. Each day's file will be missing the last few minutes of the day as the 30 minute save interval may hit at 23:35 or 23:59, but that data will always be stored in the "next" day's file.

Feel free to tweak these settings however you want. This is just a discussion of some possible starting points.

**Autosaving with a lot of targets**

If you're using PingPlotter Pro and monitoring a lot of targets, you'll need to be cognizant of the impact of auto-saving a lot of targets. Each time the auto-save timer fires, PingPlotter completely overwrites each of your auto-save files. With a lot of data in memory and a lot of targets, this can take a few seconds. As you approach the memory limit of a machine, this can often mean memory swaps occur as well, to pull in the old samples and write them out to a file. On some machines, this can take a minute - during which time tracing can stop and the GUI can become unresponsive.

The best way to manage this is to limit the number of samples in memory - if you can keep just a couple of days in memory for each target you'll still have the history and you'll get better performance (and have more available memory on that machine).

4.5 **Automatically start tracing when starting PingPlotter**

When you start PingPlotter Pro, in it's default configuration, it does not automatically start tracing to the targets you were using last time. By default, PingPlotter Pro starts in "adhoc" mode, and it's ready to trace to whatever target you want to enter.

It may be, though, that you want to start tracing one or more targets automatically when PingPlotter Pro launches. If this is the case, then it's possible to do this, you just need to set things up. Follow these steps:

• Start tracing to the targets you want to use each time.

• Save a workspace to a suitable directory (Workspace -> Save Workspace As...)

• Go to Edit -> Options, and then go to the "General" section. Turn on the "Auto-save/load active workspace" option. This will cause your current workspace to save every time you close PingPlotter Pro, and it will automatically re-load each time you start it.
You may also want to schedule auto-saving the active workspace, just in case your computer loses power or crashes. By default, PingPlotter Pro will only save the workspace when you explicitly tell it to, or when you close PingPlotter Pro. Enter a free-form time (e.g., 1 minute, or 2.5 hours) in the time field.

Now, close PingPlotter Pro. Reopen it. It should have all your previous settings / targets already tracing, and will resume where you left off.

Notes:

- If you start tracing to any new targets during a session, those targets will be resumed the next session as well, unless you close those targets.

- If you load any old collected data sets, and have that open in memory when you save a workspace, those targets will be saved in the workspace as well. When you close and restart, targets that were in memory but stopped will still be stopped when you reload.
Part V

Understanding Output
5 Understanding Output

5.1 Getting Started Guide

Our PingPlotter Getting Started guide is a great resource, with:

- Examples
- Best practices
- Interpretation of results
- Hands-on walk through on how to use PingPlotter Pro

The Getting Started Guide can be found at http://www.pingplotter.com/gsg. Please visit there for a great introduction on how to use PingPlotter Pro.

5.2 Interpreting Results - A Quick Example

So let's get into some specific examples of how to interpret the results from PingPlotter.

For the first example, you're getting intermittent packet loss to nessoft.com. What can we determine from the graph below?

First off, the final destination shows 9% packet loss. There's a problem someplace in the route, but we need to determine where....

Hop 4 shows 6% packet loss. Hop 5 doesn't show packet loss, though, so you know that the problem in hop 16 isn't because of hop 4. Hop 4 is likely just a router using a different CPU path for TTL=0 packets than it does for routing data through.

Hop 9, however, shows 10% packet loss, and this packet loss is carried on through to the final destination. This is a huge, huge indication of where the problem lies.

Now, all we know from this is that the problem happens after hop 8. We don't know if it actually happens because of CPU overloading in hop 9, a router problem in hop 9 (or even on the exit side of hop 8), or if it's the connection between hop 8 and 9. A little bit more troubleshooting is needed for this.

Digging deeper, we can see (from the domain names) that hop 8 is in the rr.com domain, while hop 9 is in the alter.net domain. Also, the IP addresses show decidedly different ranges. This is a strong clue that it's actually the connection between hop 8 and 9 that's causing the problem. It's likely that there's not enough bandwidth between those two locations.

5.3 Interpreting Results - Longterm Monitoring

PingPlotter allows you to use the timeline graphs to zoom in on any particular time, so even if you weren't there (or didn't save an image) when something was happening, you can still recover that exact image later. You shouldn't ever have to be sitting in front of your computer when an outage happens, or you experience other problems on your network, to get the data you need from PingPlotter.
The Scenario:

You're having outages (or situations you want to communicate to your ISP) randomly throughout the day. Let's say twice a day. The problem is that you can't be there everytime an outage happens so you can save a graph image.

For this example, you're keeping 24 hours of data in memory or even more. We suggest that you normally use 2.5 second trace intervals and keep 200,000 samples in memory. This is almost a week's worth of data. You can change the number of samples to keep in memory in the Auto-Save section under the Edit/Options menu selection.

Using PingPlotter's timeline graph, you can see over the past X time period (see below) to identify a time period where there was problems. Problems will demonstrate themselves as packet loss (red), or high latency.

Now you want to show the route, and the packet loss/latency in the upper graph for that time period. Since that time has already passed, you need to change the focus of the upper graph to that time in the past.

First off, you need to make sure your "samples to include" focuses in on just the period in question, so let's change that to 100 (it's important to not have it set to 0/all, but to have it be a number smaller than the number of samples in memory to be able to focus the upper graph). Right click on the timeline graph and pick a reasonable period of time to set the viewable time period. For instance, you may want to set it to six hours so you're not scrolling forever. You can then "click and drag" the graph to the left to go back to the time period you want to focus on and drag it to the right to go forward in time. Note: You can have custom timeline intervals show up in your right-click menu by adding a setting in the pingplotter.exe.ini file.

Double-click on the "problem period" in the lower graph. You'll see a focus rectangle appear on the lower time graph, and the upper graph will change to represent the data you have "focused" in the lower graph. Once you've done this, you might want to change the scale of the lower graph to show more detail. Right-click on the lower time graph again and change the scale to an hour (or maybe even 30 or 10 minutes depending on how long the outage was). The focus rectangle should still be visible. You can fine tune the data being displayed in the upper graph by double-clicking on the lower graph again.

Using these techniques, you should be able to zoom in on exactly the right data to best illustrate the problems you're seeing. You can look at the data after the problem occurs and get the perfect picture and not have to sit there watching PingPlotter all day and night.

You can auto-save that data by for instance having the auto-save function in PingPlotter create new files every day, and then load up a prior day to do the same thing you did above for a particular time. This gives you the capability to have pretty close to 100% coverage of your network performance and be able to zoom in on any particular outage, period of slow response, etc.

The options in the alert setup do allow you to have the .PP2 file (trace data) emailed, and then you can use these same capabilities to zoom in on that data.

5.4 Interpreting Results - Gamers

For this example, let's assume you're an online gamer, specifically a Quake III player (though the following is representative of any online game really - flight sims, online RPGs, etc.). You've got two servers that are running the same maps you like to play, so the only issue you have is which one out of the two is going to give you a better connection. I realize some folks aren't going to be so patient as to use the method below to decide which server they're going to play on...bear with me here. We're learning! The same topics we go over in this section, as far as graph interpretation, are also applicable if you were trying to figure out why a connection to a specific server you were just playing on is so cruddy.
The first thing you need is the IP addresses or DNS names for the two servers. You'll launch Ping Plotter, enter the IP address of the first server into the Address to Trace box and click the Trace button. Five seconds is a good value for the trace interval, and the # of times to trace should be unlimited (we're gonna watch it for awhile). You'll then launch another session of PingPlotter and enter the second server's IP address using the same settings you used for the first server.

You then go get yourself a Diet Pepsi, do some stretches or whatever in preparation for a night of gaming. When you get back to your computer you'll have two graphs that look similar to the ones below. Let's analyze both and see which server you want to play on.

Hmm. This doesn't look too bad until you get to hop 10 and start looking at the history graph. Let's take the history graph first.

Red is bad. Every time you see red on the history graph you had a timeout, or in other words there's dropped packets. Packet loss is the bane for most online games. When you're running across a big open area and then all the sudden *blip* you're on the other side (most likely dead), that was more than likely caused by timeouts, or packets you didn't get from the server or to the server. Besides the red lines on the history graph, you can also see your packet loss in the PL% column and, if you look at hop 11, the horizontal red line contains your packet loss value.

Digging a bit deeper, you can see that you're running under 111ms (the Max at hop 2) all the way down until you hit hop 10. Notice that you move off of Touchamerica's backbone into USWest/Qwest between hops 7 and 8. No problem there, there's plenty of bandwidth between those two providers since the time doesn't really go up. From the DNS on hop 9, we can see that hop is a gateway (thus the mpls-dsl-gw7 part of the name) to some DSL customers. Where we start running into problems is when we get off of Qwest Minneapolis and hit the Mediagods domain that's hanging off the DSL link. All the sudden your latency goes up to 330-501ms at hop 10. That DSL connection is busy. Once you make it to the server at hop 11, not only is your latency still way up there, but you're getting 10% packet loss. That server is a busy bee also it seems. So busy that he's not keeping up. Combined with the bandwidth saturation we're getting on the DSL line itself, it's best to try later. We don't want to play here.

Now let's look at our second server.
Now this is more like it. Really, anything under 150 ms is a great connection. Most online games are designed to not crater the 56k gamer. We don't even make it over 40ms until hop 12. Sweet.

Let's look at that connection between hops 11 and 12. Notice from the DNS names that you actually go from Seattle across Global Crossing's backbone to Cleveland. When you factor in speed of light latency, you can account for about 40-50ms of your latency to hop 17 with that hop across the backbone between hops 11 and 12. So you've got a 120-130ms ping to hop 17. That's pretty good. If you didn't have that fat pipe installed, and were instead running a modem, you'd probably be running at about 220-230ms for your latency.

"What about that 10% packet loss at hop 16?", you ask. Judging from the numbers for that hop and hop 17, what we're most likely dealing with here is a router that probably has a low priority for ICMP packets. A lot of network admins will set a router up to drop ping/ICMP packets first if it starts getting busy. Another hypothesis is, judging from the private 10.x.x.x address at hop 16, most likely hop 17 is probably sitting behind a cable or DSL connection like the first server we looked at. With the myriad of DSL/Cable modems out there, some of them don't have the most solid BIOS running on them. The problem could be that the router is just a little flaky. If you look up Adelphia.net, you can also see that it's a cable provider. Isn't traceroute fun? *grin*

So which server are you going to play? Obviously it's the second server above.

Other considerations

Your graph results can be affected by a number of things that are out of Ping Plotter's control.

1. In the analysis of the second graph above, we mentioned that hop 16 is more than likely just a problem with that router not giving us back good information. Many routers put a low priority on ICMP traffic. Still others don't even echo back requests (this will show up as a blank entry for that hop). Obviously Ping Plotter has no control over these situations.

2. Ping Plotter can't track the route that your traffic takes on the return trip from the server back to you. If your inbound traceroute traffic is taking a different route back to you than the outbound traffic to the server, this is called an asymmetrical route. By definition traceroute doesn't take these types of
routes into account and, unfortunately, Ping Plotter isn't going to be able to tell you about problems with the return route in these cases. One clue that this is happening is that you'll have a great trace up until the last one or two hops on your trace. In other words, you don't have an easily identifiable problem at hop $X$ further up the route that is mucking up the rest of the route downstream.

One thing you can do is save your trace data to a text file and post it up on a support message board for the particular game that you're playing. Even better, save off a graph and post it instead. Many savvy game server admins will actually do a traceroute (or even better a Ping Plotter trace... *smile*) back to you and be able to tell you if there's problems with a route back to you when asymmetrical routes are involved.

There are some sites that can do traceroutes back to you if you want to investigate on your own. They can be found here.

In closing, we can't emphasize this enough - latency is the bane of online gaming. Much more so than bandwidth limitations. The good thing is that Ping Plotter can tell you this latency, and provide you with ammo in the way of trace and graph data when you're beating up on your ISP.

### 5.5 Interpreting Results - ISP Problems

For this example, we're assuming the role of a user that's having problems with a broadband connection. What we'll be taking a look at is snapshots of six continuous days worth of trace data. One thing to keep in mind is that if you're doing long term monitoring and want to look at more than the largest default time span on the time-interval graph (48 hours), you can add custom time intervals in the pingplotter.ini file located in PingPlotter's installation directory.

Before continuing, if you're not familiar with how the graphs work in PingPlotter please make sure you've read the introduction to graphs earlier in this tutorial.

One common mistake we see folks make is that they'll trace to their ISP's border router. This is a bad thing. If you're tracing to the border router and your route changes (i.e. they take that router down for maintenance or you get load balanced onto another router) you really have no idea what happened. If you want to keep your traces local to your ISP, trace to an address that isn't going to change on you like you're ISP's mail server. This is actually a good thing to do if you're having mail problems and it's your ISP's mail server going down. Otherwise just pick a destination that you know has a reasonably good chance of always being up. This is a better choice since routes within your ISP can change, and PingPlotter keeps track of those route changes. The cool thing is that you're doing a traceroute here, not a ping, so even if that destination host goes down you can drill down on the timeline graph and see if it's your connection, or if it's just the destination being down (as in all hops but the destination don't show timeouts).

Another problem with tracing to your ISP's border router is that your ISP will not respect the data that you collect this way. No application targets a border, so they have no reason to trust this data. For best results, you want to pick a target that is one you use and are having problems with.

Note: For clarity, all the graphs below show us ignoring Hop 1 which you can do from the View Menu. All the graphs were saved with the File/Save Image command within PingPlotter then converted to .gif for this tutorial.
This first graph shows what the traceroute should look like with no load on the connection, i.e. no downloads, streaming audio, on-line game playing, etc. "What about all that red on the history graph? I thought red was bad?", you ask. Actually red is bad, however before I saved out this graph I double-clicked on the timeline graph to drill down, or zoom-in, and am looking at the data for 9/25/01 at 11:56 a.m. If you look at the top of the graph you see "10 Samples Timed: 9/25/01 11:56:09 AM - 9/25/2001 11:56:32 AM". So basically the above graph's trace for that particular time looks good. However when you look at the timeline graph, you can start to see the tale of woe. What we have here is a really flaky broadband connection. So how do we prove it? Read on.

Just sending your ISP a graph with red lines isn't very convincing. However, when you start zooming in on those sections with timeouts, and send graphs of them as well like this second saved graph, it's pretty obvious the connection's hosed when you can't see out to Hop 2. This is the same time interval as the first graph, just showing a different period in time for trace data.

For our third graph we've got data for early on the second day of our trace. Lots of red, and when we focus on the 3:06 a.m. time period the connection's still poor. You're unable to see out. It's hard to argue with the graph. Also keep in mind that what we're showing here is that the whole timeline graph isn't solid...
This isn't an issue where you accidentally kicked the plug on your router.

For our fourth graph, you can see from the timeline graph's times that we've adjusted the time-interval so we're only looking at 60 seconds worth of data. The trace graph is showing the section of the time-interval graph that we double-clicked on which is 4:07:21 p.m. through 4:07:49 p.m. So what's up with the 60-70% packet loss showing up on the trace graph? Notice that we didn't have timeouts for that whole 28 second period (the right vertical bar is kind of hard to see in this example, but it's at the end of the last time-out there at 4:07:49). Out of the ten samples we're looking at (depending on what hop we're looking at - remember PingPlotter is tracing each hop at the same time so it's logical that Hops 5 and 13 could at this point only be showing 60% loss), roughly 70% of them were timeouts. This is important. When we're looking at the trace data we're looking at those 10 samples we selected and the numbers for those samples, not the whole range of data shown on the time-interval graph! This is not a graph you want to send to technical support. All it's going to do is confuse them.

Ok, so what's up with this last graph? Well, like parents we try to be fair, and what you see is a flaky
connection (as you can see from the timeline graph though you'd want to zoom in to be sure where), and trace data (notice the vertical focus bars at the left of the timeline graph that denote the section of data we're looking at) that shows a problem out of your ISP's control. The ten samples we're looking at actually show a problem with the connection between Touch America (tamerica.net) and UUNET (alter.net), or Hops 6 and 7. It's hard to say exactly what's going on there, but more than likely the link between those two routers is saturated. We could try and blame a flaky router at Hop 7, but there isn't any packet loss. It's a good guess that it's the link, not the router, or we'd see the router at Hop 7 dropping packets.

So in summary, PingPlotter allows you to show your ISP where the problems are. In the these examples, we were essentially showing the whole link going down. However, we could've just as easily seen if the ISP's connection to the Internet was down at Hop 4, because we were tracing to a destination not on our ISP's local network. If there was indeed a problem at Hop 4, we would've had good trace data at Hops 2 and 3, timeouts at Hop 4 and possibly no trace data past Hop 4. If the router at Hop 3 was being flaky, and for instance you saw a lot of packet loss, it's easy to save an image showing just that so you can email it. When sending graphs to your ISP, we've found it's best to send one graph showing data for an extended time period, and then drilling down on the timeouts and sending graphs that truly show them what's going on. PingPlotter allows you to save in .png or .bmp format. We recommend .png because they're smaller.

5.6 Interpreting Results - Bad Hardware

Scenario:

External customer has problems using your network resources.

A Customer (not inside your network) has problems losing connecting to services inside your network.

In this scenario, you are acting as a service provider for some network service. This might be provided via HTTP, or possibly through something like Citrix or Windows Terminal services.

Your customer (possibly employees of your company, or maybe subscribers of your service if you're an ASP-based service) is complaining of frequent disconnects, and possibly slow performance sometimes.

How do you troubleshoot this kind of a problem? Where is the problem – is it at the entry point to your network, in the customer network, or possibly in one of the providers in between?

One way to pinpoint the problem is to have your customer run PingPlotter against your service. They can easily download PingPlotter, capture data, and then either analyze that data themselves, or email the data back for you to analyze.

This information can be used to pinpoint which hop (or router) in the chain is adding latency or losing packets that might be causing problems back to your ASP.

Analysis:

Here's an example extracted from a real-world customer situation. We'll walk through some of the symptoms, collected data, then analysis and how we came to the proper conclusion.

Our ASP (we'll call the ASP CitServeCo, a totally fictitious company) is accessed via Citrix, which is relatively sensitive to high latency and packet loss. The customer (who uses financial applications served by CitServeCo) was frustrated by disconnects during the day, which lasted anywhere from a few seconds to a minute or two, interrupting their ability to do business.

The customer will almost certainly blame CitServeCo initially for a problem like this – but we need a way to determine where the problem is being caused, and help the customer solve the problem. Any reliable ASP will get numerous complaints like this – that they "know" are the fault of something beyond their
control (like the customer’s Cable connection, or similar). Most ASPs will attest to the fact that customer connections are a prime source of network problems, but we can’t just tell the customer “It’s your problem.”

When the customer contacted CitServeCo, CitServeCo suggested that they download PingPlotter, install it onto the workstation they normally work with, and configure PingPlotter to monitor the Citrix server inside CitServeCo’s network. A one second trace interval is suggested in situations where there is ample bandwidth and a known problem.

If a disconnect or slowdown occurs in Citrix, we suggest to the customer that the right-click on the lower “time-graph” in PingPlotter and record the symptoms at the point they problem occurred. This correlates real world symptoms with the collected PingPlotter data, an important part of troubleshooting.

A day of collecting data resulted in only a single disconnect, and the customer dutifully recorded that for us. They then sent us the saved data for analysis. Because PingPlotter data is already compressed, there is no need to .zip up the file, making it easier to send data.

Here is the result of that day.

<table>
<thead>
<tr>
<th>Hop</th>
<th>FL%</th>
<th>IP</th>
<th>DNSName</th>
<th>Avg</th>
<th>Cur</th>
<th>Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>10.0.3.254</td>
<td>------------------------------</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>10.117.227.1</td>
<td>------------------------------</td>
<td>15</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>24.117.227.227</td>
<td>24-117-227.227.cpe.</td>
<td>16</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>12.118.161.57</td>
<td>------------------------------</td>
<td>57</td>
<td>213</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>12.123.196.134</td>
<td>gbr6-p51.dltx.ip.alt.r</td>
<td>52</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>12.122.12.69</td>
<td>tbr1-p012701.dlstxip</td>
<td>52</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>12.122.12.74</td>
<td>ggr1-p340.dlstxip.alt</td>
<td>50</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>204.255.168.9</td>
<td>so-0-0-2.BR6.DFW9.</td>
<td>58</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>152.63.97.57</td>
<td>0.so-1-0-0.L1.DFW</td>
<td>55</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>152.63.101.254</td>
<td>0.so-4-0-0.XR1.DFW</td>
<td>53</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>152.63.100.193</td>
<td>185.ATM7-0.GW7D</td>
<td>52</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>gateway n-2</td>
<td></td>
<td>61</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>gateway n-1</td>
<td></td>
<td>62</td>
<td>53</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>destination</td>
<td></td>
<td>66</td>
<td>55</td>
<td></td>
</tr>
</tbody>
</table>

**Target Name:** N/A  **IP:** -- masked target --  **Samples Timed:** 35  **Time:** 4/21/2004 4:35:08 PM - 4/21/2004 4:36:08 PM  **Graph:** 213

**Round Trip:** 66 55

Notice the packet loss in the timeline graph at 11:20 am. This corresponds with a note from the customer in the data file that shows the disconnect happens. We decide we want to take a closer look. To do this, we set “Samples to include” to 35 (an estimated of the number of samples included in the loss period), double-clicked on the red point in the time-graph, and then reset the time-graph period to 5 minutes instead of 6 hours.
Wow. A big outage. If no packets were getting through, we’re definitely going to see problems. Where did that loss occur? If we look at the upper graph, we see that we have consistent packet loss across “all” the routers (double-clicking on the lower graph focused the upper graph on the period we were interested in). The packet loss was similar, but let’s have a look at the actual lost packets. To do this, double-click on a router in the upper graph to show a time graph for that router.

It looks like hop 1 lost just as many packets as hop 14. Since every piece of data needs to go through hop 1 to get to any of the other hops, a blockage there will look just like this. It looks like we’ve found a likely culprit – the router at hop 1, or possibly anything between the computer collecting data and hop 1. This might be as simple as a network cable, or it might be a significant amount of network equipment. We don’t know until we check with the customer to find out what’s here.

We asked the customer what kind of network hardware they have in place. While they were collecting this information for us, we had them continue to monitor their connection.

It turns out that all the computers at this location are hooked up to a “SOHO” router. This SOHO router is, in turn, connected to a cable modem provided by Cable one. From Cable one, the customer is uncertain as to the network configuration. We see some of this in the PingPlotter graph – a list of routers that are participating in sending data.

The customer was uncertain as to what the individual pieces of hardware reported in as, so we had them continue to monitor, but asked that they cycle the power for the devices if they had a disconnection problem. Several days later, they had an opportunity to do this, and captured the experience in
Based on the comments, we can see that they had a disconnect, then rebooted their SOHO and their cable modem. Notice how hop 1 stays working through the reboot of their cable modem, but not through the reboot of the SOHO. So this must be the SOHO. Also, notice during the disconnect, hop 1 was non-responsive. This indicates the problem was similar to having the SOHO powered off.

Maybe there was a bad power supply on the SOHO, or maybe a bad network cable someplace? The customer replaced the network cables, and the problem persisted. They then replaced the SOHO router and the problem was solved.

It’s not every case where the problem is so “in our control” as it was in this one. Sometimes, the problem requires replacing an ISP, or talking an ISP into replacing some piece of their hardware. PingPlotter can
be just as effective locating the outages, packet loss, or latency on ISP equipment, though – it’s just harder to fix those problems.

5.7 Interpreting Results - Bandwidth Saturation

Every network connection has a limit, and in this example we’ll talk about how to recognize bandwidth limits on a local DSL / Cable modem.

The scenario here is a 640kbs downstream, 256kbs upstream DSL modem running a 2.5 second trace interval to www.pingplotterpro.com. The computer running PingPlotter was connected to the DSL modem via a wireless network card, which loses packets occasionally.

We're downloading a 811 meg file, which should take somewhere between 3 and 4 hours. During this 3 to 4 hour period, the bandwidth of the DSL modem should be completely saturated, which means that anyone else using this network connection is going to notice significant latency and possible packet loss. Any user might decide they want to troubleshoot this situation, and might run PingPlotter to do so.

A quick 10 minute trace might look something like this:

First, notice the big latency jump between hop 1 and hop 2 - the DSL modem. This same latency jump is translated into all downstream hops, which indicates that hop 2 (or the link between hop 1 and hop 2) is significantly impacting network latency. Since we know what's going on here, it's pretty easy to
recognize. In a lot of cases though, you might not know what’s happening and you’ll see latency like this.

Now, the “Sawtooth” pattern in the timeline graph is a classic bandwidth saturation pattern. Anytime you see a pattern like this, think “bandwidth saturation”. Keep in mind that bandwidth saturation is totally normal, and happens on just about every network. Consumer-grade DSL and cable modems have the highest chance of seeing a pattern like this, but it can happen on any connection where the bandwidth at a network point is being totally saturated.

Here is the same collected sample set, but with more time shown on the time graph:

<table>
<thead>
<tr>
<th>Hop</th>
<th>PL%</th>
<th>IP</th>
<th>DNSName</th>
<th>Avg</th>
<th>Cur</th>
<th>Graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>152.168.1.1</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>139.60.134.13</td>
<td>DSLModem13-3341.spr.net</td>
<td>329</td>
<td>245</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>139.60.134.17</td>
<td>fa12-0-0-core1.boi.spr.net</td>
<td>331</td>
<td>216</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>138.60.134.10</td>
<td>fa5i-1-0.big2.boi.spr.net</td>
<td>334</td>
<td>280</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>65.121.130.33</td>
<td>boi-edge-02.net.qwest.net</td>
<td>332</td>
<td>310</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>205.171.155.21</td>
<td>bse-core-02.net.qwest.net</td>
<td>334</td>
<td>315</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>205.171.205.150</td>
<td>tkw-core-01.net.qwest.net</td>
<td>348</td>
<td>366</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>205.171.190.6</td>
<td>tkw-core-03.net.qwest.net</td>
<td>352</td>
<td>352</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>205.171.205.110</td>
<td>svl-core-02.net.qwest.net</td>
<td>373</td>
<td>386</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>205.171.14.117</td>
<td>svl-core-01.net.qwest.net</td>
<td>371</td>
<td>388</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>205.171.214.38</td>
<td>pax-brdr-02.net.qwest.net</td>
<td>373</td>
<td>381</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>208.50.13.185</td>
<td></td>
<td>374</td>
<td>402</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>67.17.67.225</td>
<td>so0-0-0-2488m.cr1.pao2.gblk.net</td>
<td>375</td>
<td>395</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>67.17.68.122</td>
<td>so0-0-0-622m.ar2.cle1.gblk.net</td>
<td>431</td>
<td>462</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>64.214.174.178</td>
<td>pair-networks.so-2-1-0.ar2.cle1.gblk.net</td>
<td>436</td>
<td>456</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>100</td>
<td>0</td>
<td>100% packet loss</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>216.92.151.75</td>
<td>pingplotter.com</td>
<td>439</td>
<td>450</td>
<td></td>
</tr>
</tbody>
</table>

**Round Trip:** 439 450

Here you see the entire 3.5 hour period where the download was happening. Outside the download, the latency is pretty much rock solid. The packet loss stays relatively constant across the entire period, indicating that the packet loss isn’t being caused by bandwidth saturation. We know that it’s actually caused by a wireless network that has about 1% normal packet loss, and the download rate doesn’t affect it at all. At 6:10pm we switch to a wired network card and the packet loss goes away.

There are several solutions to the latency problem:

- Procure (buy) more bandwidth
- Don’t transfer as much data

This same situation happens in any network. Your ISP runs into these same exact options with the connection between their network and the outside Internet. They can limit what you download, or they can buy more bandwidth. On a bigger pipe the jump in latency isn’t going to be as pronounced, and the
bandwidth will be saturated by a number of different loads. Often on a primarily consumer-use network (DSL / Cable), the increase in latency will be gradual as more people use the bandwidth, and then it will gradually drop off as people stop using it. The rise in latency often happening at 6-9pm, and the drop off happening at 11pm - 1am. It's often accompanied by packet loss as well.

On your own network, the bandwidth saturation could be happening for a number of reasons: another user downloading something; temporarily restricted pipe size (problem with your Internet connection), p2p applications in use, or possibly even a virus/worm type application that is using bandwidth sending out new instances of itself.

In summary: Saturated bandwidth is normal, and isn't always an easy problem to solve. It's often easy to recognize the symptoms though, especially if you look at the trend over time. PingPlotter can really help pinpoint the problem, giving you the ability to see trending over time, latency patterns short and long-term, and latency / packet loss correlations.

5.8 Voice over IP (VoIP) troubleshooting

Introduction

Using an IP Network (like the internet) to conduct a voice conversation (VoIP) is a growing trend and is becoming easier and easier for people to do. It can be inexpensive and relatively reliable.

It can also be challenging - with poor voice quality, the inability to hear and communicate, delays and other problems.

The underlying technology for VoIP is extremely network dependant. If you're having voice quality problems, the problem is often related to the network - maybe your internet provider or maybe some other component between you and the called party.

This article will talk about some basic troubleshooting techniques that can be used to locate where the problem is occurring so you can make good decisions about how to solve the problem.

Network-related VoIP Symptoms

Many symptoms of VoIP problems are network related (although certainly not all of them). Here are some examples of symptoms that are often network related:

- Garbled words ("blips" and clicks mixed in with the words)
- Parts of words missing
- Gaps where the other side is talking, but you don't hear any of it
- High "distortion"
- Delays between the time you talk and the other side hears you (and vice versa)
- You start talking not realizing that the other party has started talking already too, and you talk over each other for a few seconds

Other symptoms might not be network related. In particular, if the symptom *always* happens, any time of the day, any day of the week, then there's a decent chance it's not a network problem.

- Echo when you talk. This can be exacerbated by network problems, but constant echo is usually not caused by network problems.
- Inability to connect a call to some users while you can call others

Using PingPlotter to identify the source of network problems

PingPlotter has some unique capabilities in its ability to help you track down the source of network
problems.
What you really want to know is:

- Can you fix the problem or do you need to call someone else to fix it?
- If you need to call someone else, who do you call? Your ISP? Your VoIP provider?
- When you call them, how do you convince them it's their problem to fix?

PingPlotter can offer a lot of insight into all of these questions.

**Collecting data with PingPlotter**

Before we can do much analysis, we need some data to analyze. We cover some of these topics in our [Getting Started Guide](#), so we will not cover *details* of how to set things here.

First, we need a target server to monitor. Ideally, this would be the actual VoIP server of your VoIP provider, or something on the same area of the network. If you called your VoIP provider and they asked you to collect PingPlotter data, they may have given you a server to use. In many cases, the use of any server can work, but this will only identify problems with your ISP - not with your VoIP provider. The good news here, though, is that the vast majority of VoIP problems are because of front-line service providers (like your ISP). If you don't know what address to use and you have no way of finding out what address your VoIP hardware or software is using, try using the web site of your provider.

For this discussion, we will be using our web server - www.nessoft.com - as the server we're monitoring and using for troubleshooting.

Use the following settings in the main PingPlotter screen:

- **Address to trace**: (the server just discussed - www.nessoft.com in our case)
- **Trace Interval**: 2.5 seconds
- **# of times to trace**: Unlimited (enter 0 to set this)
- **Samples to Include**: 250 (this not important right now, so you can leave it at default)

Now, hit the “Trace” button. You should see a picture appear that looks something like this:
The upper graph should show a full route, including the "Round Trip". If you don't get a Round Trip, check in the troubleshooting section of this document for some ideas.

Now, let this run for at least 30 minutes - preferably, during a period where you're making a voice call. Ideally, you'll have a period where you have a voice call that's good and one that's bad, but that might be possible. If nothing else, just let it run for long enough to get a good sample of your network conditions.

A great thing to do while you're collecting data is to make notes in the PingPlotter data about what you're experiencing. You can see instructions on how to do this in our Getting Started Guide's chapter on collecting data in the "Creating a comment or note" section.

The data we collected covers several days. PingPlotter works great to just run over a long period of time so you get a good idea of what network conditions look like - during good times and bad.

Examining data with PingPlotter

Once you've collected some data, it's time to have a look at what might be the problem. We cover some of the PingPlotter commands on zooming, focusing and digging in the Finding the source of the problem section of our getting started guide.

One of the key things to know here is that we're looking for problems at the last hop only - and then using the other hops to determine where the problem starts. Packet loss or latency that shows up only at an intermediate hop is not a problem!

Let's look at the graph above. Notice how hop 15 has a full 100% packet loss, and hop 11 has 27% packet loss? The final destination looks rock-solid, though - no packet loss and the latency is really nice and smooth. This is, in general, what you want to see - a solid, flat line at the final destination, no packet loss.
An analogy - network traffic is a bit like freeway traffic

A network is a bit like a freeway - it works great when everyone is going the speed limit and we have 50% of the maximum traffic that's designed to go on that freeway. As we start to add more traffic, at some point we don't have capacity for more. We start to have problems as people merge on the freeway. People already on the freeway sometimes slow down and cause traffic jams. Sometimes, when it gets too bad, people give up and decide not to continue the journey. On a freeway, this might be called "congestion".

This "congestion" happens on network too - and it works pretty much the same way. Packet loss, latency and jitter (the 3 enemies of call quality) are all symptoms of congestion - when there's too much traffic for the network to handle.

PingPlotter sends out packets that go all the way "there" and back again. We measure the time it takes, and also measure how often a packet (or a router) gives up on the packet we sent.

Let's continue the freeway analogy a bit. Let's say between here and our destination there are 15 offramps with turnaround points off the freeway. We'll send out 15 cars and assign each car to one of these intersections, with instructions that when they get to that intersection, they should turn around and come back. Then we'll measure the time it takes to get from us, to each intersection and then back again.

The most important time is the one that goes all the way to our target - that 15th car. If it makes it there and back again in expected time, then we know the freeway traffic is running pretty well - everything made it through just fine.

If, however, the 15th car takes longer than expected (or it never returns!) we can look at our records for the other intersections to find a likely place where problems are occurring. Maybe all the cars out through intersection 9 had no problems - and returned quickly, but the cars going to intersection 10 and beyond started getting delays. From this, we can see that there is some kind of traffic problem past intersection 9.

Just for the sake of taking this analogy too far, let's look at one other aspect. Let's say that intersection 5's turnaround spot is in a small town where the police are of a disposition to pull people over for no reason at all. Each of our cars who are going to intersection 5 have to use that turnaround, and 20% of the time they get pulled over there. Another 15% of the time, there's someone else pulled over and they have to wait while that car moves off the road. Meanwhile, traffic is whizzing by on the freeway, unrestricted.

This situation can happen on a network with PingPlotter as well - where the packets going to hop 5 might get waylaid by some local rules and show packet loss, latency and jitter that are not being experienced by packets destined for other places.

So, what are we looking for, when it comes to problems?

The first place to look is the final destination. If the time graph looks like the graph above (straight line, no red), then PingPlotter is not finding network problems. Look for problems at the final destination. If you find a problem at the final destination, then look back until you find the first hop showing similar symptoms - that's who we probably need to contact to get the problem corrected.

Examples and Analysis

Note: Most of this data is fictional - based on truth, but not real. You should certainly not make any decisions about the respective companies indicated in this data - as that would be very, very wrong. Some of this data is 8 years old!
Example: Distributed packet loss

Let's look at an example, this time an example with problems:

Here, we see 8% packet loss at hop 16 (the final destination). This would result in poor voice quality, dropped "bits" from words and hard to understand conversation. Notice that the latency is pretty good still - it's just the packet loss that's a problem (packet loss is all of the red in the time graphs and the red bars in the trace graph). With a pattern like this, voice quality would be consistently "iffy" - not unusable all the time, but not very good either.

Notice how the packet loss is happening at all hops from hop 6 onward, while hop 5 looks relatively good. The packet loss percentage is similar all the way down - around 8% (statistically, it would be just about impossible for all hops to have identical packet loss percentages with this kind of loss). To turn on and off time graphs like this, just double-click on the hop number in the trace graph.

Hop 11 has high latency and higher packet loss - but see how hop 12 goes back to results similar to hop 10? So hop 11 probably isn't introducing any new badness into the traffic. So we should ignore hop 11 here. The same thing goes for hop 15 - it's not sending any data back, but it's passing through data just
fine to hop 16.

So, in this situation, the problem looks to be between hop 5 and 6. It's pretty likely that Qwest knows about this problem - it's in the "middle" of their network - and it's all owned by Qwest (we can see that from the DNS Name column).

In this case, since we're subscribed to Qwest, it's a pretty easy decision to call Qwest and complain. The picture above is pretty compelling and would be a good communication tool to them.

**Example: Local bandwidth saturation**

Note: We have another example of this in our tutorial / manual.

Here, notice the big latency jumps - you have a nice flat line, then a jump in latency - including some packet loss. This pattern is one that is almost always a bandwidth saturation issue (which is the same as congestion). In the case we have here, hop 1 is inside our network (our DSL modem, actually) and hop 2 is inside Qwest.

This is a case where we were transferring too much during this period - and we were using all of our
available bandwidth. A VoIP call would suffer significantly during these periods - there is a lot of jitter (the "ragged" line is an easy way to see jitter - where packets take different amounts of time to arrive), higher latency and some packet loss. The voice quality would be bad, there would be additional lag, and it would probably have audio drops.

There are a few options for solving this one, but none of them involve complaining to anyone else:

- You can install a traffic shaping modem - that gives higher priority to VoIP data (actually, unless you've configured PingPlotter packets to look like voice data, you might already have one of these in place - this article does not cover that topic, though).
- You can get more bandwidth (although that doesn't solve all problems - as you'll still be able to use all your bandwidth).
- You can use less bandwidth.
- You can get an additional broadband connection and dedicate it just to VoIP (this is an especially good idea for heavy VoIP users or businesses). The low cost of an additional broadband connection makes this viable in a lot of situations.

**Example: Border congestion**

Congestion often happens at network borders - where one network hands off to another. This is relatively common for small, growing ISPs - where they just do not subscribe to enough bandwidth to handle everything. Let's have a look at what this condition might look like. We're going to use a different network for this picture.
This one isn't *quite* as simple, as there are a few factors. The symptoms of conditions like this would be:

- During high load times (evenings), this connection would be completely unusable for VoIP. With 15% packet loss and really high jitter, this would be absolutely horrible for any kind of voice call.
- During the morning till afternoon, it would be *bearable*, but the jitter would cause the words to be garbled sometimes, and not too fun.
If we look at the network conditions with PingPlotter, we see a couple of problems. First off, there's some serious packet loss starting at hop 9. This packet loss is carried down through the rest of the route to the final destination. This is a border - between rr.com and alter.net. Having problems at borders like this is pretty common - that's where one company pays another to handle traffic - and if a company is growing, it might be "oversubscribed" - using more bandwidth than is available.

An interesting part of this is how during heavy load times for home users (ie: evening hours), the packet loss and latency are worse. During early morning hours, it goes back to being OK again. This is a big sign that the problem is load related - and that this link is having congestion problems at "rush hour". Time to add some lanes!

Another problem, though, is inside the rr.com network there is significant latency and jitter. There are some slight symptoms at hop 2 (which is the border between our internal network and rr.com - so that's the cable modem), but starting at hop 6 there's some real bumps in latency and the jitter (latency variation) is also a big cause for concern.

In this case, both problems are inside the rr.com network, and since we are an rr.com customer, we would call them for help on this.

**Example: 802.11b network near its range limit**
Here’s an example where we’re connected using a computer-based VoIP service (like Skype). Our computer is hooked up to our DSL modem via a wireless 802.11b network. Hop 1 is our DSL modem.

Here, we see a little bit of packet loss being added to every hop - our wireless network is losing a few packets (about 1 to 2 percent, it looks like), and this impacts everything this computer does - including our VoIP connection.

The call quality would be generally good here (probably better than acceptable - up to the "good" range, really). The latency is fine and there is very little jitter, but there is a little packet loss. There is a problem, though - at 8:53pm, our call was interrupted - it looks like hop 1 lost a bunch of packets all together and during that period, we were unable to hear anything. Let's zoom in on that a bit.
See the period at 8:53 / 8:54 where we start getting a lot more packet loss, and then all the hops show a big block of lost packets - a period where it's likely no packets were getting through.

Here, the solution might be to move the wireless access point, or switch to wired on that computer.

**Reporting problems, when you find them**

If you're using PingPlotter, it's almost certainly because you're experiencing some kind of problem - and when you find something that you think might be the cause of that problem, you need to communicate that to the right party.

We [cover this topic](#) in some depth in our Getting Started Guide. The piece we want to stress here is that the data in PingPlotter doesn't really mean anything unless you correlate it with a network problem (like poor VoIP quality). It's of paramount importance that your complaints include a description of how this problem is affecting you. Don't just send a graph from PingPlotter expecting them to be able to figure out what was wrong.

One great way of doing this is to put comments in the PingPlotter graph itself using the "Create Comment" feature of the time graphs. Make comments every time your VoIP quality is bad. Make comments when you give up on a conversation because they can't hear you at all (but you can hear them just fine - how frustrating!).

**Troubleshooting PingPlotter**

If you get "Destination Unreachable" at something beyond hop 3 or so, but can access that site via a web browser.

Some sites do not respond to ICMP echo requests. See our knowledge base article for instructions on how to configure PingPlotter to use TCP packets instead of ICMP.

If you get "Destination Unreachable" at hop 1 for all targets

Make sure your software firewall (ie: ZoneAlarm, etc) is configured to allow PingPlotter to have access to the network.

If you only have the final hop visible - and all intermediate hops are empty

We cover this in this [knowledgebase article](#).

**Other questions**

**Jitter**

Jitter is the amount of variation in latency. If one packet takes 100ms and the next one takes 200ms,
there's 100ms of jitter there. PingPlotter Pro offers jitter calculations and graphs, but PingPlotter Standard (and the 2.x line) still gives you an easy way to see the jitter by looking at the smoothness of the time graph, zoomed in a little. Here's an example:

![Graph showing 100ms of jitter](image)

This is zoomed in enough for us to see the individual samples - and we can see that none of them come in with the same latency. Adjacent samples here often have latency variations of 100ms - and just about every one has latency variation of 30ms or higher. Just looking at this graph, we can see a lot of jitter. Compare that to the first picture in this article - where the line was completely flat. We're looking for the flat lines, not big variations with red stuck in everywhere.

**Other resources**

This article introduces some concepts and ideas about VoIP troubleshooting. There are other resources online that provide more depth (albeit not within the context of PingPlotter).

- www.whichvoip.com/voip-troubleshooting.htm - a great page with real-world solutions and suggestions. Targeted to residential, but useful everywhere.
- www.voiptroubleshooter.com is a great site that has an enormous amount of content on which symptoms relate to what kinds of network problems. This site has a relationship with Telchemy, a leading VoIP provider of call quality monitoring tools.
Part VI

Advanced Features
6 Advanced Features

6.1 Command Line Arguments

You can have PingPlotter do a few things automatically on startup by specifying command line parameters. You can put these parameters in a shortcut - or enter them from a DOS command line window.

`pingplotter.exe [File to Load] [/TRACE:[Address To Trace]] [/SAVE] [/SINGLEINSTANCE] [/NEWINSTANCE] [/INIFILE:(filename)]`

Loading a file at startup

If you enter a parameter without a / on it, PingPlotter will try and find a file by this name - and load it if it's found.

Example:


/TRACE

This option will start tracing automatically when PingPlotter loads. If use the option to load a save file on startup, then tracing will begin to this address. Otherwise, Add a colon (:) and the IP Address or server name you want to trace to.

Example:


or

`pingplotter.exe /TRACE:www.pingplotterpro.com`

SAVE

This option can only be used when you specify a file name on the command line. If you use /SAVE, then any new traces (to the original address specified in the save file only) will be saved to that file on shutdown automatically (without asking you if you want to save).

Example:


/SINGLEINSTANCE

If PingPlotter is run with this parameter, then it checks to see if another copy of the program is running. If it is running, then it exits.

Also, if you passed in an address to trace (via the /TRACE:address option above), then this address will be passed to the currently running version. The currently running one will start tracing to that address.

Example:

`pingplotter.exe /SINGLEINSTANCE /TRACE:www.pingplotterpro.com`
Note: PingPlotter Pro defaults to Single Instance - where loading a new save file or starting a new trace will use the already running instance, if there is one. To change this behavior, use the /MULTIINSTANCE (or /NEWINSTANCE - which is equivalent), which will open a new instance for that command.

/INIFILE

You may want to start PingPlotter with a different set of parameters and setups. This is particularly useful if you’re auto-starting PingPlotter and having it trace automatically. In this situation, you may want to have different trace intervals, or graph times, or whatever. Starting PingPlotter with an alternate INI file allows you to save multiple setups and use these different setups as needed.

Example:

pingplotter.exe /INIFILE:alternatesetup.ini

/?

Show this help screen.

Example:

pingplotter.exe ?

6.2 Custom time graph intervals

You can change, and also add, additional timeline intervals for the timeline graph if the defaults available from the right-click menu don't fit your needs. For example, you may want to look at results farther out than the default maximum value (which is 7 days).

The default set isn't written in the .INI file, however you can add an entry and it will override the default values.

For example, if we wanted to add "2 hours" and "14 days" intervals to the right-click menu, your .INI file entry would be what you see in the example below. It's important to note that the values are in minutes.

It's also important to note the "Count" value. If the value was set to "Count=12" in the example below, the value for "Interval13" wouldn't show up in the menu.

[TimeGraphIntervals]
Count=13
Interval1=1
Interval2=5
Interval3=10
Interval4=30
Interval5=60
Interval6=120
Interval7=180
Interval8=360
Interval9=720
Interval10=1440
Interval11=2880
Interval12=10080
Interval13=20160
6.3 Installer Options and MSI

The PingPlotter install is MSI-based, and wrapped with a bootstrap that helps do upgrades.

Normally, the best way to install PingPlotter is just to launch the installer by downloading and double-clicking.

In some cases, though (particularly for deployment to multiple computers), it may be helpful to change the way PingPlotter is installed.

The bootstrap has several options to extract the MSI, log the install to file, debug, etc. To see the options, launch the installer from a command line and pass it a /? parameter - this will list the parameters that can be used.

License Entry

PingPlotter license are stored in the registry in the following location:

```
[HKEY_LOCAL_MACHINE\Software\Nessoft\PingPlotter Pro\User]
"UserName"="Your Username"
"RegistrationCode"="Your License Key"
```

If you need to automate license key entry, you can write these values into the registry.

*** 64 bit windows warning *** - Since PingPlotter is 32 bit application, on a 64 bit machine, this needs to be written to:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Nessoft\PingPlotter Pro\User]
```

Custom Options

There are several other options available through the command line interface of the installer. Please contact our support team with your needs, and we'll help you build the right install package.

6.4 Whois Setup

If you'd like to add additional whois servers to have available from the trace graph's right-click menu, you can edit the pingplotter.ini file. Currently PingPlotter only supports xxx.yyy style addresses. Not, for example, xxx.yyy.uk. You can also change the default servers directly in the .ini file instead of changing them from the Options/Internet menu.

To change the default settings, add the following to your pingplotter.ini file:

```
[Internet]
StandardWhoisHost=whois.networksolutions.com
BlockIPWhoisHost=whois.arin.net
```

Also, you can add additional servers with a list of commands like the following:

```
[Internet]
AddlWhoIsServers=Internic Whois Lookup, whois.networksolutions.com, name,
Arin (IP) Whois Lookup, whois.arin.net, ip
```

Basically, each whois server is defined by 3 settings - Description (as it appears on the menu), Server address (IP or name), and lookup type (Name or IP). The last setting specifies if it queries the whois server for the IP address, or the name of the specific hop.

If you're adding a server that's not specified in PingPlotter by default, add that server as the first server on
the AddIWholsServers line (i.e.: don't include the default servers shown in the example above, since they're always included no matter what you put in the INI file). If you have more than 1, just add 3 more comma delimited sections. There is no limit to the number of sections you can include, though more than a few will make the menu a bit unwieldy.

6.5 Running from a USB drive

PingPlotter Pro can be run from a removable drive (like a USB key) without having to be installed, or modifying anything on the "host" machine where you're running PingPlotter. This is great if you administer a number of PCs and want to be able to run PingPlotter without having to install it, and/or enter a license key.

There are several differences in operation when PingPlotter is running from a USB drive in special USB mode.

- The license key is stored on the USB drive, not in the system registry. This means you don't have to enter the license key on each machine you're running on.
- All settings that would normally change the registry will no longer work. For example, you cannot install as a service when running from a USB drive. Also, the .pp2 file association is not tied to PingPlotter automatically.
- All PingPlotter settings are written to the .ini file on the USB key (in .exe directory - so make sure it's writable!).
- PingPlotter does not check in the user's profile for override settings.

To run PingPlotter Pro in this special mode, follow these steps:

1. Install PingPlotter Pro normally on your computer's hard drive (not the USB drive).

2. Copy the entire directory where PingPlotter Pro is installed to your USB drive. This should include all subdirectories that may exist. You can put this in any directory on the USB drive, but we'll assume this is in f:\PingPlotter for now. The executable to launch PingPlotter would then be f:\PingPlotter\pingplotter.exe

3. Uninstall PingPlotter Pro from your computer's hard drive.

4. Create an empty file named "license.dat" into the directory where pingplotter.exe exists on the USB drive (in our example, this is f:\PingPlotter).

5. Launch PingPlotter Pro from the USB drive. When prompted, enter your license key. PingPlotter will put your license information into the license.dat file.

6. Create an Autorun.inf (outside this scope of this topic, but email us if you need help) or hook up the f:\PingPlotter\pingplotter.exe file to your USB drive's menu system.
Part VII

Reference
7 Reference

7.1 Options

The options dialog is the main way to configure PingPlotter Pro.

There are a few interesting behaviors that may not be obvious.

**Applying (and identifying) Pending Changes**

First, any changes made to these settings do not take affect until you either hit the "Apply" or "OK" button. The Apply button will become enabled when there are any changes pending to be applied.

The topics in the tree view with the * marks (asterisk) are the ones that have changed. Hitting the Apply key will write these changes to PingPlotter and put them into affect. Hitting the "OK" button will do the same, but will also close the options dialog.

Hitting cancel will back out the changes that have not been applied.

You can make changes to multiple areas before applying them.

**Creating / Manipulating named configurations**

PingPlotter Pro supports [multiple named configurations](#). Right clicking the “Default Settings” option above will allow you to create more.

If you delete a named configuration that is currently in use by a target, that target will automatically change to use the first configuration in the list.

**Making "on the fly" changes while still using PingPlotter**

If you want to make changes in the options dialog will still interacting with the main PingPlotter Program, hold the control key down while you launch the options dialog. The options dialog will stay on top, but it will allow you to interact with other areas of PingPlotter at the same time. As soon as you hit the “Apply” button, these changes will go into affect - without you having to close the options dialog.
### General Options

#### Put icon in tool tray?
Enabling this option will turn on a tray icon at all times. When PingPlotter is minimized, it will only be visible in the tray. When it’s not minimized, it will show on the taskbar and the tool tray. Alert conditions can be surfaced through the tool tray as well, in which case the icon will change to red and a message might appear.

#### Show "Round Trip Time" row?
The "Round Trip" row of PingPlotter duplicates the information from the final hop and makes it evident that the host is reachable and what the round trip time and latency is. Before adding this option, we got a lot of questions “What’s the round trip time?”.

#### Prompt to save on exit if lots of new data exists
When PingPlotter is closed and has been collecting data, you might not want to just throw away all that data. If this checkmark is on and there are at least 75 unsaved samples in memory, PingPlotter will ask if you want to save it when closing.

#### Minimize PingPlotter when Windows "close" command is used
If you normally run PingPlotter all the time, you might not want it to close if you accidentally hit the "close" button on the application (ie: the X button). Turning on this option will make PingPlotter minimize instead of close. To close PingPlotter, use the "File -> Exit" menu option, or if PingPlotter is minimized to the tray, use the right-click "Close" command. Note that using the close command from the taskbar will not close PingPlotter, as that is equivalent to using the X button.

#### Include settings name in target / host descriptions
When tracing to the same target with different engine settings (see our [named configurations](#) documentation for more details), the only way to distinguish between the settings is by the named configuration. If you're only using one named configuration (or if you're tracing to all different targets), then this is not so helpful. Turn on this option to show the named configuration on all tabs and time graphs. For the summary graph, turn on the "Settings" column to show the named configuration used for that target.
Summary Graph Settings

Summary graphs are exclusive to PingPlotter Pro.

Automatically show final destination on default summary screen

If this option is on, when you start a new target in PingPlotter Pro, that target will automatically add itself to the summary graph screen. The summary graph screen is a handy way to see the current status.

Automatically show timeline graphs for targets added to summary screen

When a new target or router is added to the summary screen, having a time graph automatically show up can be handy. The downside of this is that a long list of targets quickly fills up the screen and becomes less useful. Turn off this option if you find yourself regularly turning off a lot of the automatically added graphs. You can turn them back on at any time manually anyway.

Workspace Settings

Workspaces are exclusive to PingPlotter Pro.

Save collected data with workspaces.

A handy option that is usually turned on - this makes it easy to save a workspace and reload with everything just the way it was. If disabled, reloading the workspace will *not* reload the collected data, just the targets and layout. The data will be saved in a directory named the same as your workspace (without the file extension).

Auto-save/load active workspace when PingPlotter is closed/opened.

A great way to have PingPlotter remember everything you were working on and automatically pick it back up when you reload. When you close PingPlotter, everything is saved. When you re-open, it's loaded again.

If you have this on and want to clear everything and start fresh, you can do that via the workspace menu.

Auto-save active workspace every:

Manually key a time into this field, like "30 minutes" or "6 hours". Click out or tab out of the field to "verify" what you entered and have PingPlotter interpret it. 30-60 minutes is a pretty good number to use here. Numbers like "1 second" mean that your computer will probably be constantly saving things, and won't have much free time for anything else - especially if you get a lot of data collected.
7.1.2 Internet Options

The "Internet" settings control the connectivity for whois, update checks, etc.

**Standard Whois Lookups**
Right-clicking a hop on the upper "route" graph on any hop that has a DNS name defined gives us the option of looking up that name and seeing who the owner is. We use the "Whois" protocol for this, and we talk to the server specified here. Default: whois.crsnic.net

**IP Block Lookups**
Internet IP Addresses are assigned ownership, and sometimes it's interesting to find out who owns them. This is the server that does this for us – default: whois.arin.net

**Version Checking**
PingPlotter can check with the PingPlotter servers occasionally to see if there's a new version available. This can be done automatically, or it can be done manually by hitting the "Check Now" button.

**Proxy Setup**
The Proxy Setup is currently used for only one thing – doing the version check. PingPlotter doesn't use these settings to do the trace in any way.
7.1.3 Web Server Options

The PingPlotter Pro integrated web server allows access to PingPlotter from a remote machine. The user interface and functionality of this web server is designed to give the owner of this machine access to trace information remotely.

Enable built-in web server

PingPlotter Pro includes its own web server for this purpose. You can, however, use IIS as the web server and point at PingPlotter Pro "www" directory and use IIS instead. If you are going to customize the scripts, it is recommended that you use IIS instead of the PingPlotter Pro built in web server.

When enabled, you can access PingPlotter Pro's web interface through this URL:

http://localhost:7464/

Server Port

The default port for PingPlotter Pro is 7464 (or PING on a telephone keypad). Change this to something more appropriate if applicable.

Web server security settings

For best security, you'll want to require a username and passport to access PingPlotter Pro's web interface. If you turn off security, then anyone with a browser who has network connectivity to your machine will be able to manipulate your PingPlotter Pro sessions (including adding new targets and closing existing targets).

Served File List

For best security, every file that the PingPlotter Pro's built-in web interface serves up needs to be added to this list of files. There is no way to "wildcard" in the files - each needs to be explicitly added.

When adding, look at examples of the existing ones. The "HTTP Name" should have a leading slash (/). The Actual file name should not, unless it's based in the root directory. There is no requirement that the actual file be in the PingPlotter Pro www path - you can point at any file.

Adding a plus character (+) to the end of the "Actual File Name" indicates that this is a script file (executable) and that PingPlotter Pro will parse it as a command / script file. The absence of a + means the file will be served straight-up, no parsing.
7.1.4 Windows Service Options

PingPlotter Pro offers some startup options that allow you to monitor with PingPlotter Pro continuously - even if you don't log in and if you reboot occasionally.

See the topic on Running as a service [page number] for more details.

Note that when running PingPlotter Pro as a service, the user interface might behave slightly different that when running as an application. This is particularly true on any file browse dialogs. Make sure you specify absolute paths on any locations.

Installing as a service

The "Status" shows the current service status. Please read these instructions [page number] for more details on how to configure this.

Workspace to load

When starting PingPlotter as a service, you can specify a pre-configured list of targets and user interface configurations in the form of a "Workspace" file. This workspace is created by using PingPlotter Pro in normal mode, then using the "Workspace" -> "Save Workspace" command. When saving this workspace, make sure you save to a known path, and one that is accessible to the "Service" user.

Automatically update workspace on shutdown / reboot

If you regularly change the list of monitored targets, and you want PingPlotter Pro to remember those changes, turn on this option. When enabled, PingPlotter Pro will automatically update the configuration when you reboot or manually stop the service.

If you always want PingPlotter Pro to restart with the "same" targets, then disable this option. To update the workspace manually, just use the "Workspace" -> "Save Workspace" menu option.

Save collected data on shutdown, reload on startup

If you want to continue tracing, automatically, when you shutdown and reload the PingPlotter Pro service, turn on this option. Each auto-save event that happens for a target will write a service reload file, as will stopping the service normally (via a reboot or manually stopping the service). When this option is enabled, restarting the service will look for these save files and automatically resume tracing.

Save File location

All service data files (used for resuming the trace) are stored in this directory. It's important that this directory is readable and writeable by the "System" user, so it should be on the local machine. When running as a service, all auto-saved data files will be copied into this directory.
Email Settings

The email setup dialog is used to set up emailing for alerts. If you're not using alerts, or you're not interested in having the alert system email you, then setting this up is not required.

Return Address

All outgoing emails will have a return address specified, and this is the address that is used. Please make sure you specify a valid address here since this is where all the bounce messages will come from. Some ISP SMTP servers only allow emails sent out with a "from" address of their domain as well, so if you're having problems getting the SMTP server to work, make sure you're using a valid return address.

Attach PingPlotter Savefile (.pp2) on Email Alert

When an alert goes out, data will be attached. The data that is attached encompasses all the alert period since the "last" similar alert was sent out. This is a global setting for all email alerts. Note that you'll need PingPlotter installed on a machine to read the attached data.

Include "alert" samples in text message

If this option is checked, any outgoing alert emails will include text showing the samples that failed the alert. As each alert email goes out, the past history is marked as being sent already so that you won't get duplicate data in reoccurring emails.

SMTP Server

The SMTP server is the server that your outgoing mails will go through. This may have been given to you by your ISP or your mail administrator.

Server Port

The default port for most SMTP servers is 25. If you connect to your SMTP server via a different port, then enter that port here. Leaving this blank will use port 25. If you're using STARTTLS/SSL, then this might be port 587 or some other port as supplied by your email server/provider.

SMTP Authentication

Some SMTP servers require a username and password to be able to deliver mail. If this is the case with your server, turn on the "Use SMTP Authentication" checkbox, and then enter your username and password. The password is saved in your PingPlotter.ini file using a basic XOR encryption scheme – this will keep your password hidden, but this encryption method is "crackable" if someone really wants to figure it out by looking at your .ini file and reverse engineering it.

Using SSL for SMTP

PingPlotter Pro also supports use of SSL (STARTTLS) for SMTP. For details on this, see our web site.

Named Configurations

Each configuration can be associated with one or more targets, and control the way data is collected and displayed for those targets.
Named configurations allow you to use different settings for different targets within the same instances of PingPlotter Pro. We talk about this in quite a bit more depth in our Named Configurations topic.

The "Configuration Label" here is used throughout PingPlotter to associate a target to a configuration. The description is informational only.

### 7.1.7 Display Options

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color HOP column by speed</td>
<td></td>
</tr>
<tr>
<td>Color graph background</td>
<td></td>
</tr>
<tr>
<td>Warning (yellow) speed (ms)</td>
<td>200</td>
</tr>
<tr>
<td>Critical (red) speed (ms)</td>
<td>500</td>
</tr>
<tr>
<td>Draw X on graph for current sample</td>
<td></td>
</tr>
<tr>
<td>Draw line to show Min/Max range</td>
<td></td>
</tr>
<tr>
<td>Automatically Scale to Sample Times</td>
<td></td>
</tr>
<tr>
<td>Scale to</td>
<td>500</td>
</tr>
<tr>
<td>Don't graph packet loss</td>
<td></td>
</tr>
<tr>
<td>Packet Loss Scale at %</td>
<td>30</td>
</tr>
<tr>
<td>Add jitter graph above time graphs?</td>
<td></td>
</tr>
<tr>
<td>Jitter Graph Scale</td>
<td>60</td>
</tr>
<tr>
<td>Jitter &quot;Target&quot; line</td>
<td>35</td>
</tr>
<tr>
<td>Graph Height</td>
<td>32</td>
</tr>
<tr>
<td>Hide if area less than</td>
<td>90</td>
</tr>
</tbody>
</table>

The "Display" settings control the general display format of PingPlotter’s graphs, including scaling, coloring, and other general values.

**Color HOP column by speed**

If you want to minimize on-screen color, turning this off will hide the green, yellow and red background color on the hop (first) column.

**Color graph background**

Some 256 color graphics drivers might not look very good with the colors on the graph. Turn off this switch to use a non-color background.

**Warning and Critical speed limits**

Changing these values sets the Green / Yellow / Red threshold for the graphs. This is dependant on your expected performance. For a modem, 200 ms might be quite good, while for a T1, it could be considered bad.

**Draw X on graph for current sample**

When examining current data, sometimes its handy to graph the most recent sample collected. Turn
Draw line to show Min/Max Range
When showing just a few samples, this can be really handy to see the range of latencies. As you increase your window, though, a single bad sample can make this line stretch the scale of the graph.

Graph Scale
Normally, setting the graph scale to automatic works pretty well. Sometimes, you might get a few samples “way” out of range, though, that stretches the scale. This is especially likely as you increase the value for samples to include.

Packet Loss
Most often, graphing the packet loss is a handy, easy way to see lost samples. 30% seems to work great for highlighting just the right of loss in most cases, but you’re certainly going to run into cases where you want to change this to something lower (as low as 1) or higher (any number is valid – even over 100).

Jitter
PingPlotter Pro allows you to graph jitter correlated with the time graph. In many cases, jitter is apparent when examining the standard PingPlotter time graphs, so the jitter graph is only displayed when there is enough room. The settings here allows you to control when that is displayed, and what it will look like.

Jitter Graph Scale and Target line
These settings are represented in milliseconds, and control the range of jitter values that will fit into the graph. Note that this also applies to the web interface, so if you want the jitter graphs to show up in the web interface, these settings need to fit the height of the web interface time graphs.
7.1.8 Engine Options

The "Engine" settings control what and how PingPlotter sends data.

Packet Type
The "Packet Type" settings allows you to pick what kind of data you want to send to tailor PingPlotter to your network needs. PingPlotter supports 4 packet types:

- **ICMP using Windows DLL (default).** This method is the traditional method and matches the data that the Windows TRACERT command uses. It works on all Windows operating systems, and is a good balance of reliability and capability.

- **ICMP using Raw Sockets.** This method sends the same data that the Windows DLL does, but does so by manually creating the packets in PingPlotter. This sometimes works in cases where ICMP.DLL doesn’t, but requires administrative rights on the PC - and doesn’t work on all versions of Windows.

- **UDP Packets (Unix-Style).** Instead of sending ICMP packets, this sends UDP packets, which might be able to get through some firewalls that are blocking ICMP. This uses ports 33434 – 33500 and closely mirrors Unix’s traceroute command. This requires administrative user rights.

- **TCP Packets.** If a firewall is blocking ICMP packets, it’s sometimes possible to get a response using TCP packets instead. TCP is the protocol used for all web browsers in addition to FTP, Telnet and others.
requires Administrative user rights, and on most operating systems also requires a helper library.

**Timeout Speed**
When PingPlotter sends out a packet, it waits a certain amount of time for a response. The longer it waits, the more resources it needs to use (to keep sockets open), but the more likely that it will get a response. 9999 is a pretty high number – very few packets take longer than this to return. Some people use 1500 here.

**Time interval between hop traces**
PingPlotter sends out packets without waiting for previous packets to return. This doesn't mean, however, that it sends out all packets at once (you don't want to do this, or it might overwhelm your bandwidth). This is in ms (or 1/1000s of a second).

**Packet Size**
The Packet Size can make a considerable difference in latency performance. Normally, you want to use a relatively small number here. The default is 56 bytes, but in some cases you might need to lower this (especially on TCP port 80 packets, which sometimes get dropped unless they are 40 bytes). 1500 is lot of data, and should be used with great care. A 1500 byte packet means PingPlotter will be sending out 30-50 K per second worth of data, which can cause its own problems (and makes measuring latency more challenging).

**ToS/DSCP header byte**
Some networks use this byte to prioritize data. If that's the case, you can manipulate this value to see how the network responds. This is relatively advanced course material – ie: if you don't know what this means, just leave it at 0.

**Starting Hop**
Sometimes hop 1 or hop 2 might never respond. Rather than continuing to pound away at these hops and never getting a response, it sometimes makes sense to just ignore the first hop or 2. This is totally normal on lots of cable modems, and can happen on any connection – where the first hop is always "silent". Ignoring the first non-responding hops will save some resources.

**TCP Specific Settings**
When using TCP packets, you can specify which target port to use. Usually, you'll want to use port 80 here, but you're welcome to use any reasonable port. Windows firewall blocks creation of TCP packets, so you'll need to use WinPCap to create packets under that OS (and possibly others). See our online page with more details at [http://www.pingplotterpro.com/winpcap.html](http://www.pingplotterpro.com/winpcap.html).

**Use non-threaded Name Lookups**
This makes PingPlotter do all DNS lookups in the main application thread. If you're having DNS lookup problems, it might be worth trying.

**Maximum concurrent requests**
Setting this too high can cause PingPlotter to crash, especially on old operating systems. 45 is proven safe. If you have a really fast trace interval, you may need to increase this number to support the trace interval. Usually, you want to lower the "Timeout Speed" setting before raising this, as that might be adequate.

**Cargo**
Every packet has a minimum size (28 bytes for ICMP/UDP, 40 bytes for TCP). Anything in addition to that core size is padded with a value. Modify these settings to pad with the value you're interested in. Different packet "cargo" values might help expose network problems.
7.1.9 Auto-Save Options

The "Auto-Save" settings allow you to automatically save data or graphical images at an interval of your choice.

Auto-save data

Turning this setting on will cause PingPlotter to save a .pp2 file at the interval you choose. You should specify the directory you’d like to save in, and you should also include some variable as part of the name – otherwise every save will overwrite the previous save.

Auto-save image

Automatically saving an image is an interesting way to create a history, or to update a web page with data automatically. Note that the written image will always look like whatever’s on the PingPlotter screen – so if you’re looking at some period besides the current period, “that” image will be saved (possibly multiple times).

Changing the “File Type” here also controls the default file type when saving from the main screen – so even if you’re not auto-saving, you can change this value to set that default.

Variables: Both the data and the image filenames use variable substitution.

Maximum samples held in memory

Long-term monitoring sessions might start to fill up memory, making things run slower. If you want to purge off old samples when memory starts to get full, set this. Make sure you’re auto-saving the old data, though, or it will be lost forever!
7.1.10 Route Change Options

Route Change Exclusion Masks

List the IP "exclusions" for which you want to ignore route changes.

Example: if you have a normally oscillating route that oscillates between 192.168.1.1 and 192.168.2.1, then enter "192.168.3.1" in the exclusion list above. Basically, mathematically or the addresses that should be handled as equivalent. If you want to ignore ALL route changes, enter the word ALL.

By default, PingPlotter tracks all route changes. Sometimes, these changes might be normal and might add confusion (ie: too much data) rather than clarity. PingPlotter supports "masking", or excluding certain changes.

Route Change Exclusion Masks

Route change masks can be manually added. Alternately (and much easier) is to add a mask by right-clicking on the oscillating router that you’d like to stop signaling a route change.

Show route change indicator when list hidden

When the route change list is hidden, the normal behavior is to show an indicator there when the route changes. If you don’t want to know about route changes, you can turn off this option.

7.1.11 Plugins & Scripting

PingPlotter Pro has a built-in scripting engine which allows extensions to its capabilities. There are a handful of default extensions that ship with PingPlotter Pro, and you can add more, or create your own.

The list of scripts that show in this dialog is controlled by the list of .ppx files the exist in your (PingPlotter Pro)\scripts directory. By default, this path is:

c:\program files\PingPlotter Pro\scripts

Any .ppx file found in that directory will show up in this list. The .ppx file itself controls if it defaults to be disabled or enabled. Once it shows up on the list, you may need to manually enable (or disable it) through here.
You can completely disable the scripting engine, or just do any individual script.
To create a new script, just copy one of the existing ones from the script directory, and save it with a new name. This script will then show up as a new one, and you can then edit it to do your bidding.

7.2 Menus

7.2.1 File Menu

1. **New Target...** - (Pro version only) This will create a new empty target area where you can trace to a new instance. See the documentation on "Tracing to Multiple Targets" for more details.

2. **Load Sample Set...** - Loads a previously saved sample set. The default extension for PingPlotter saved sample files is .pp2, or PingPlotter savefile format.

3. **Save Sample set...** - Allows you to save the current sample set to an external file. These files are saved in .pp2, or PingPlotter's savefile format.

4. **Save Image...** - Saves the current graph in .png, .gif or .bmp format. See the Autosave section for information on how to automate the saving of graph images.

5. **Export to Text file...** - Exports trace data to a comma delimited text file. Click here for an explanation of the export options available from PingPlotter..

6. **Send Email** - This option will launch your email client and automatically create an email with the current sample set, image or both.

7. **Exit** - Exits PingPlotter. By default you'll be prompted to save your current sample set if you haven't done so already.
7.2.2 Edit Menu

1. **Copy as Image** - Copy the current graph to the clipboard as an image. From here, you can paste the image into your favorite graphics program or an email.

2. **Copy as Text** - Copy the current graph to the clipboard as text. Hold down the shift key when clicking the Edit menu to toggle between copying all the collected data details, or copying a summary.

3. **Alert Setup...** - Create and Edit alert configurations. See the Alert setup options for more details.

4. **Options...** - Go to the configuration and options setup area.

7.2.3 View Menu

1. **Put Icon in Tool Tray** - When this option is on, PingPlotter will move to the tool tray when minimized.

2. **Show Round Trip Time** - Don't show/show the Round Trip time below the trace data graph.

3. **Ignore First Hop(s)** - The default is "trace all hops". From this menu option you can skip hops 1, 2, 3, 4 and 5. Say you're hooked up to a DSL modem and don't want the modem to show in your traces. You'd set this option to Start at Hop 2.

   If you wish, you can also choose to only trace the final hop.

4. **Show Timeline Graphs** - Don't show/show the Timeline Graphs (the bottom history graphs).

5. **Customize View...** - this will allow you to enable / disable columns from the upper trace graph.
6. **Graph Current Sample** - Turns on / off the “X” on the trace graph for the current sample. By "Current", this means the most recent sample in the selected set.

7. **Graph Min/Max range** - Turns on / off the Min/Max range lines in the upper graph. This can be handy to turn off for long-term monitoring sessions where the maximum range might push the graph scale too high.

### 7.2.4 Workspace Menu

Workspaces are used in PingPlotter Pro only, and are a list of targets, screen locations and settings that make it easy to continue a monitoring session later. We discuss this in some detail in "Tracing to Multiple Targets" and "Running as a service".

1. **Load Workspace...** - This will load a workspace - which is a list of targets, trace intervals, named configurations and screen locations. PingPlotter workspaces use the extension `.pws` by default. Loading a workspace will stop tracing and close any targets that are currently active.

2. **Clear Workspace** - Closes all tabs and stops tracing - including the summary tab! See the Summary Graphs topic for more information on how to create a summary tab.

3. **Save Workspace** - Save the current workspace. If you loaded this workspace, or saved it this
session already, then the same name will be used and the workspace will be overwritten. If PingPlotter doesn't know what name to use, it will prompt you.

4. **Save Workspace As..** - Save the current setup to a new workspace file.

5. **Start All Targets** - Stop all targets that are currently being trace.

6. **Stop All Targets** - Resume tracing all targets.

7. **List of targets** - At the bottom of this menu is a list of all targets / tabs being traced. You can easily switch between them by using the Workspace menu to find the one you need.
Part VIII

More Information
8 More Information

8.1 Support

We offer a number of great support resources for PingPlotter that might be able to answer your questions without having to ask us. We’re always happy to get an email from you as well though.

Your first step in solving your problem should be to visit the support site on the PingPlotter web site (http://www.pingplotterpro.com/support.html). That site is updated often - more often than this static help file on your hard drive. This site has links to our FAQ, Tutorial, Support forums, knowledgebase and other useful information.

Support Page:
http://www.pingplotterpro.com/support.html

Getting Started Guide:
http://www.pingplotterpro.com/gsg/

Tutorial and Product Manual:
http://www.pingplotterpro.com/tutorial/

Knowledgebase:
http://www.nessoft.com/kb/

Support forums:
http://www.nessoft.com/forums

Email support:
support@pingplotter.com

8.2 Purchasing

PingPlotter Pro requires a license (and a license key) to run after the evaluation period has expired.

We offer a variety of ways to purchase, including credit card, PayPal, check, money order, or via purchase order (after approval). The very easiest way to order is with a credit card online – you’ll get a license key immediately, and you won’t have to wait for your order to be processed by hand (as other methods require).

To purchase a license, visit our online order page at:
http://www.pingplotterpro.com/order.html

Thank you!

8.3 Related and interesting links

Occasionally, we run across an interesting topic, link or web site. We have no affiliation with any of these sites, and some of them may not even be up. Enter at your own risk (arrrggghhh - pirate thoughts abound after reading the Traceroute and Pirates article).

Sorry, if you're reading this on a piece of paper, the links don't show up - you wouldn't want to type them in anyway, so just visit our web page, or go to the electronic version of this manual and click the links.
• **Some thoughts on speed of light latency** - great article about why a good ping is important.
• **DSL reports** (your one stop shop for xDSL information)
• **808hi.com** (tons of information for dialup modem users)
• **Speedguide.net** has equipment reviews, tweaks, etc.
• **Daryl's TCP/IP primer** - interesting information about TCP/IP
• **The ping page** (links to lots of stuff related to ping)
• **The story of PING** - where it all started...
• **Traceroute (and Pirates)** - humorous "How does traceroute work?" article. Sam Spade seems down, and this is too good of analogy to lose, so we can find it here on the Internet Wayback Machine.
• **NetMon.org** (lots of cool tools and links)
• **The museum of broken packets**
Index

- A -
Agents 18
Alerts Setup 28
Alerts Setup: Event Notification 34
Alerts Setup:Event - Launch an executable 36
Alerts Setup:Event - Log to file 37
Alerts Setup:Event - Play a Sound 35
Alerts Setup:Event - Send Email 35
Alerts Setup:Event - Tray Icon Change 38
Alerts Setup:Troubleshooting Alerts 33
Annotation 12
Applying changes 80
Auto-save 43, 91

- B -
Browser 15

- C -
Cargo 89
Clipboard 17
Closing Target 12
Columns 87
Column Widths 7
Command Line Arguments 74
Configuration Label 86

- D -
DNS Name 6
Docking 12
DSCP 89

- E -
Engine type 14
Examples 61
Exporting 17, 41

- F -
Flash Drive 77
Floating 12

- H -
History Graph 7

- I -
Image creation 91
Introduction 2

- J -
Jitter 61
Jitter Display 87

- L -
Latency 4
License Key Entry 76
Licensing 98
Long term monitoring 43

- M -
Masking 92
Memory footprint 43, 91
Menu
  Edit 94
  File 93
  View 94
  Workspace 95
Minimize instead of close 81
Monitoring 44
Multiple Targets 12

- N -
Named configurations 14
- O -
Options  80, 87

- P -
Packet size  89
Packet type  89
Ping Thresholds  87
Proxy server  83

- R -
Registration  98
Remote Tracing  18
Removeable Drive  77
Reporting  17, 41
Reset and Restart  6
Resizing Time Graph  9
Resuming on reload  44
Round Trip Time  81
Route change masks  92
Route Changes  6

- S -
Samples to Include  6
Save on exit  81
Scripting  92
Scrolling  10
Service  39
Shortcut Keys  10
Show on summary graph  81
SMTP Password  86
SMTP Server  86
SMTP Username  86
Sort  9
Summary Graphs  9
Support  98

- T -
Time Graph  7
Time Graph Annotation  12
Time Graph Interval  75

- U -
Unix Agent  18
USB Drive  77

- V -
Version check  83
VoIP Troubleshooting  61

- W -
Web interface  15, 84
Whois  76
Whois lookup servers  83
Windows Agent  18
Windows Service  39
Windows service options  85
Workspaces  25, 44

Timeline Graph  7
Timeline Graphing  10
Timeout speed  89
Tool Tray  81
ToS  89
Trace Address  7
Trace Graph  7
Trace Interval  6
Trace Route  4
TTL  4