

TestPoint™
Internet Toolkit

Program and documentation copyright (C) 1999 by Capital Equipment Corporation. No parts of this book may be reproduced or transmitted in any form or by any means, electronic, optical, or mechanical, including photocopying and recording, or by any information storage and retrieval system, without permission in writing from Capital Equipment Corporation.

The software accompanying this manual is licensed to the user by Capital Equipment Corporation. The software is copyrighted (C) 1998 by Capital Equipment Corporation. A single-user software license allows the user to install and run the software on a single computer, plus limited rights of distribution as follows: TestPoint applications written using the TestPoint objects contained in this software package can be freely distributed as runtimes, including the support files TPMAPI.DLL, TPSEVER.EXE, and TPTCP.DLL. Network licenses are described on separate documentation accompanying network-server-based packages.

Limited Warranty

Capital Equipment Corporation (CEC) warrants the physical diskettes or CDROMs and documentation enclosed herein to be free of defects in materials and workmanship for a period of sixty days from the purchase date. The entire and exclusive liability and remedy for breach of this Limited Warranty shall be limited to replacement of defective diskettes or documentation, and shall not include or extend to any claim for or right to recover any other damages, including but not limited to, loss of profit, data, or use of the software, or special, incidental, or consequential damages or other similar claims. In no event shall CEC's liability exceed the purchase price of the product.

TestPoint and TestPoint Internet Toolkit are trademarks of Capital Equipment Corporation.

TestPoint Internet Toolkit
Second edition

Capital Equipment Corporation
900 Middlesex Turnpike, Building 2
Billerica, Massachusetts 01821
(978) 663-2002

Chapter 1. Installation	1-1
Enabling the Toolkit.....	1-3
Chapter 2. Introduction.....	2-1
Chapter 3. Tutorial on Protocols	3-1
Internet and Intranets.....	3-2
TCP/IP	3-3
Computer Addresses and Names.....	3-4
Ports.....	3-6
Web Browsers and HTTP.....	3-7
E-mail.....	3-8
Chapter 4. Setting Up Your Network.....	4-1
A simple 2-computer setup.....	4-2
Network Administration, Gateways, Firewalls, and Security	4-7
Choosing Addresses	4-10
Installing and Configuring TCP/IP in Windows.....	4-11
Testing a TCP/IP connection.....	4-12
Configuring Web Browsers.....	4-13
Chapter 5. TestPoint Web Server	5-1
Overview	5-2
Web Server Example.....	5-4
URLs - Names of Objects to View.....	5-6
Directly Viewing a TestPoint Panel - No Configuration Necessary	5-10
Creating Web Pages for your Server.....	5-11
Web Server Options	5-13

Chapter 6. TestPoint E-mail Client	6-1
Overview	6-2
E-Mail Example	6-3
Using the E-mail Object	6-4
MAPI vs. SMTP Mail Protocols and Addresses	6-6
Settings	6-7
Chapter 7. TestPoint TCP/IP	7-1
Overview	7-2
TCP/IP Example	7-3
Using the TCP/IP Object	7-4
Clients, Servers, and Ports	7-6
Settings	7-7
Actions	7-10
Events	7-13
Handling Multiple Client Connections	7-15
More Examples	7-17
Chapter 8. Distributing Runtime Applications	8-1

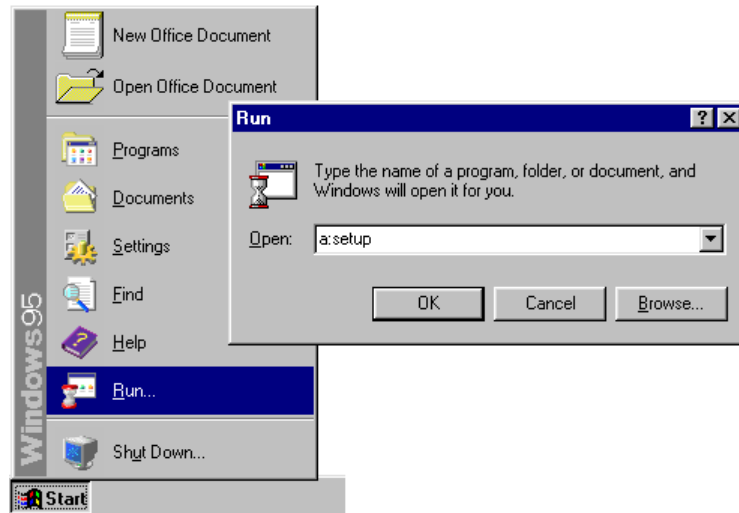
Chapter 1. Installation

System Requirements

TestPoint version 2.0 or higher
Windows v3.1 or higher (including Win95, Win98, or WinNT)
Network interface
80386 or higher processor
8MB system memory
1MB free disk space

Setup

Select the Start menu and Run, or File/Run and choose your floppy disk drive and the SETUP program. For example:



Install the software into the directory containing your copy of TestPoint (usually C:\TESTPT).

See Chapter 4 for details on setting up your network for TCP/IP protocol and web access.

Enabling the Toolkit

If you are installing this toolkit as an addition to an existing TestPoint system, you will need to enable the new features through the registration dialog window.

After installing the toolkit:

1. Run the TestPoint editor.
2. If you have TestPoint version 3.3 or earlier, you will also need to open a file that uses the internet toolkit features. Use File/Open and select the TCPIP.TST example from the WEB directory.

You will get a message about features that have not yet been registered and enabled. This message will give specific instructions on how to enable your software and register it.

Registering your software can always be done via web, email, or FAX. You'll need information provided on the TestPoint screen about your computer or key ID and software serial number(s), along with your name, company, phone number, and e-mail address.

Registration via:

web	www.test-point.com
email	registration@test-point.com
FAX	978-663-2626

Help is also available at: support@test-point.com

Details of enabling/registration:

To obtain technical support and permanently enable your software, you will need to register at the TestPoint registration data center. You will obtain a software code that you enter into TestPoint to permanently enable your copy.

For TestPoint v3.4 and later, enabling your software is done directly through the message window that appears when you run TestPoint. This gets you up and running immediately. You must then send in your registration information to obtain a permanent enabling code.

For TestPoint v3.3 and earlier, a code number is required before your toolkit software is fully enabled. Normally this code is delivered with your software package. If it is not, contact us and we'll get it to you right away.

While your software copy remains unenabled, you will still be able to use the features of the toolkit, but you will not be able to save applications which use these features.

Chapter 2. Introduction

- **View TestPoint results from *any* Web browser**
- **Monitor tests remotely**
- **E-mail enable your TestPoint applications**
- **Send results or alarms via e-mail**
- **Build client/server applications**

Your TestPoint Internet Toolkit includes the following components:

TestPoint Web Server

- Allows any running TestPoint application to be viewed on web browsers: whole panels, or just one object.
- Supports JPG and GIF formats for graphical views, and text format for views of data values.
- Also sources disk-based web pages, which may be created with any HTML editor, and can include embedded TestPoint images.
- Images can automatically update at any desired time interval (browsers which support "server push", such as Netscape, only).

TestPoint E-mail Client

- Send text e-mail from any TestPoint application.
- The e-mail client object can be used in your custom application to send any desired text, including measured results or alarm notifications, to any e-mail address.
- Supports both SMTP (internet) and MAPI (Microsoft Windows) e-mail interfaces.

TestPoint TCP/IP Interface

- Connect to any other computer via TCP/IP protocol (the internet standard).
- Works as both client and server, so two TestPoint applications on different computers can communicate.
- Can also directly communicate with any TCP/IP-based program on any network-connected computer. With an internet gateway/router, that can be any computer in the world, or you could use only your corporate intranet.
- With the TCP/IP object, you can build remote-control panels in TestPoint, to not only monitor a TestPoint program on another computer, but modify its settings remotely if desired.

Chapter 3. Tutorial on Protocols

Internet and Intranets

The Internet is a world-wide phenomenon that no longer requires any introduction. What began with the U.S. Department of Defense Advanced Research Projects Agency (ARPA) and a number of universities has now grown to millions of users, ranging from engineers to schoolchildren. The Internet connects computers together via phone lines, dedicated wires, and high-speed fiber-optic links.

Intranet is a somewhat newer term, referring to the use of internet technology within one organization for communication and information sharing. Corporate intranets can be on a single local-area network, or may span multiple offices, using the public internet as a carrier.

The intersection of inter- and intranets and TestPoint comes when data acquisition, measurement, and test results may be usefully shared across networks. Many programmers and engineers have attempted to build custom solutions to provide remote access to their measurement systems. Use of internet standards technology, the TestPoint Internet Toolkit, and commonly available tools like Web browsers and e-mail make remote access easy to implement.

TCP/IP

TCP/IP is an acronym standing for the two fundamental communications standards used on the internet to connect computers together. TCP (Transmission Control Protocol) governs how data is sent back and forth and broken into segments for routing through networks. IP (Internet Protocol) governs the addressing and logical links.

TCP/IP is only one of various network protocols that commonly are used in local networks. IPX, for example, is a protocol used on Novell networks.

Networks may be physically connected by a variety of means, from Ethernet local networks, to phone dial-up lines, etc. These form the "physical layer" and the "data-link layer". TCP/IP protocol works on top of these layers to provide computer-to-computer connections. Applications and job-specific protocols like SMTP (for e-mail) or HTTP (for web browsers) operate on top of the TCP/IP layer, calling on it for its services.

Computers running Microsoft Windows can be configured with multiple network protocols installed. TCP/IP is provided as part of the Windows 95, Win98, and Windows NT operating systems, and it is available as a free add-on from Microsoft for Windows for Workgroups 3.11.

Computer Addresses and Names

IP address numbers

Every computer using TCP/IP protocol must have an IP address. IP addresses are 32-bit numbers, usually referred to by giving four separate bytes as values from 0 to 255, separated by dots. For example:

187.154.54.100

Every computer being connected via TCP/IP must have a unique IP address, to distinguish the origin and destination of messages. Addresses are assigned by a central agency (at this time it is the Internet Network Information Center, or InterNIC, at <http://ds.internic.net>) that allocates groups of addresses and keeps track to make sure they are unique. Addresses for local networks are often assigned in groups of 256, with the first three address numbers being pre-assigned, and the last number ranging from 0 to 255.

Private IP addresses

It is also possible to have a local network that does not directly connect to the global internet, in which case it will not have any address conflicts no matter what addresses are chosen. To simplify matters, a special group of addresses has been pre-defined for such networks, never to be used on the public internet. Such **private addresses** are of the form:

192.168.0.xxx

where "xxx" can range from 1 to 255.

Host Names and Name Servers

You can also refer to computers by name. The name is translated into an IP address by the software.

One way to provide translation from names to addresses is to create a local file on your computer (usually in the Windows directory) called "hosts". This file contains one or more lines giving address and host name, like this:

```
192.168.0.1    gateway
192.168.0.100 server1
```

Another method is to have a **name server**, which is a computer configured to respond to requests to look up a name and return an IP address. Such servers use a protocol called **DNS**. Your local network may have a DNS server, or you may be configured to use a DNS server at your internet provider's site. You configure your choice of one or more DNS servers in the TCP/IP installation and configuration screens of Windows (see Chapter 4).

Gateways, Firewalls

It is also possible to have a network that uses private addresses and still communicates with the internet via a special computer configured as a "gateway" and "firewall". This computer is assigned a real internet IP address and carries out all communications on behalf of all the local network computers, forwarding the messages back and forth as they arrive.

Such firewalls can also provide additional security to the local computers, avoiding outside access to private data and files, or regulating outgoing access as well.

Ports

Every computer using TCP/IP can have one or more "ports" to which other computers can connect. Each port has a number, between 0 and 65535, by which it is addressed. Each port also has a single local program which manages that port and communicates on that port. In this way, a single computer can have multiple programs, each supporting different communications and protocols.

A server program on a single port can have multiple active connections.

There are some standard port numbers usually reserved for certain programs and protocols. Here are a few of the common ones:

80	HTTP (web servers)
25	SMTP (e-mail servers)
21	FTP (file transfer protocol)

Web Browsers and HTTP

The World Wide Web and browser programs like Netscape or Internet Explorer are the most well-known application on the internet. These browsers allow you to view text and pictures from other computers, and click to follow links between documents.

Web browsers communicate with the computers containing web pages (the servers) using a protocol called **HTTP** (HyperText Transfer Protocol). HTTP is simply a set of rules for the queries and responses that move web pages around the internet. When you enter a web site location into your browser, it simply sends an HTTP "Get" command to the server, which in turn sends back the contents of the requested page.

Web pages themselves begin as text files, containing a mixture of plain text and **HTML** (HyperText Markup Language) commands. HTML commands can specify type fonts, colors, embedded picture files, links to other pages, and other features of a web page.

E-mail

Electronic, or e-, mail is simply a method of sending messages from one computer user to another, including the ability to store and forward these messages for later retrieval.

Internet e-mail uses a protocol called **SMTP** (Simple Mail Transfer Protocol), again handled by a server computer. When you send SMTP mail, you are giving a series of commands, followed by your message, to the SMTP server computer. This computer then sends your mail on to its destination.

Microsoft has also defined a mail interface for its Windows operating system, called **MAPI** (Mail Application Programming Interface). MAPI is a standard set of operating system functions that can be used by a program to send mail. MAPI mail can go to other users of a local network postoffice, or it can be forwarded through a "mail gateway" program to users on the internet or other mail systems.

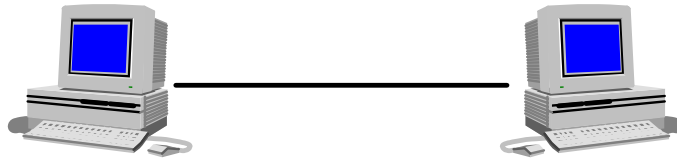
Chapter 4. Setting Up Your Network

We recommend you begin, if possible, by setting up a simple 2-computer test network, as described in this chapter. This avoids any complexities of setup related to a large network, internet gateways, dial-up lines, and interaction with other corporate network resources and servers.

This chapter also gives information and guidelines on using the TestPoint Internet Toolkit in a more general network setting. If you have a **network administrator**, you should always contact that person regarding network configuration issues and installing any new server computer onto the network.

A simple 2-computer setup

We recommend starting with this setup, if at all possible.



You will need:

- 2 computers with network interface cards
- Network cable (and network hub box if required, it depends on the type of network boards and cabling you have chosen)
- Windows for Workgroups 3.11, Windows 95, Win98, or Windows NT
- If you are using Win 3.11, you will need the Microsoft TCP/IP-32 drivers, downloadable from Microsoft for free. (<http://www.microsoft.com/kb/softlib/mslfiles/TCP32B.EXE>)
- TestPoint
- TestPoint Internet Toolkit
- A web browser (preferably Netscape, but can be Internet Explorer)

Step 1: Networking the computers

Install Windows, if not already installed, on both computers.
Install the network cards, according to the manufacturer's directions.

Connect the two computers, according to the directions for your chosen network cards. (For Ethernet using 10-Base-2, you directly connect the two computers, and you need a "terminator" plug at each end of the cable. For Ethernet using 10-Base-T, you need a network "hub" box into which you plug both computers.)

Configure Microsoft Network on the computers, according to Microsoft's instructions.

In Win 3.11, you use the "Network Setup" icon in the "Network" group.

For Win95 and above, you use the Start/Settings/Control Panel menu, and double-click on the "Network" icon.

Use the Help button as needed for instructions.

Step 2: Check that your network is correctly configured

In Win95 and above, use the "Network Neighborhood" icon to see if the computers can communicate with each other.

In Windows for Workgroups 3.11, use File/Run/Browse/Network, and see if the other computer shows up in the list.

If you have a problem, please refer to the network card instructions and Microsoft help files. Up to this point, there is nothing TestPoint-specific, so it is not within the technical support expertise of CEC or TestPoint resellers.

Step 3: Install TCP/IP protocol on the computers

Return to the Network configuration screen.

Windows for Workgroups 3.11

In Windows for Workgroups 3.11, double-click the "Network Setup" icon in the "Network" program group.

Push the "Drivers" button.

Push "Add Protocol".

Select "Unlisted or Updated Protocol".

Insert a disk containing the TCP/IP-32 drivers, or specify the directory where you have downloaded these drivers from Microsoft. You will now be in the configuration screen - continue below under "Configuration".

Windows 95, 98, NT

In Windows 95 and above, you go to Start/Settings/Control Panel, and double-click on "Network".

If you already have TCP/IP protocol installed, just double-click on it to configure it.

To install TCP/IP, push the "Add" button.

Select "Protocol".

Select "Microsoft", then "TCP/IP".

Configuration

In the TCP/IP configuration screen, enter an IP address for the computer.

For the **server** computer, enter **192.168.0.1**.

For the **client** computer, enter **192.168.0.2**.

Set the "subnet mask" to 255.255.255.0

Because we are using only two computers, we don't need any DNS (name server) computer addresses, nor do we need any gateway computer addresses.

Step 4: Test your TCP/IP configuration

Open a DOS command window on the client computer and type the command:

ping 192.168.0.1

You should see a response, with time indicated, from the other computer, similar to this:

```
G:\>ping 192.168.0.1
Pinging [192.168.0.1] with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time<10ms TTL=32
Reply from 192.168.0.1: bytes=32 time<10ms TTL=32
Reply from 192.168.0.1: bytes=32 time<10ms TTL=32
Reply from 192.168.0.1: bytes=32 time=50ms TTL=32
G:\>
```

If you have a problem, please refer to the Microsoft help files. Up to this point, there is nothing TestPoint-specific, so it is not within the technical support expertise of CEC or TestPoint resellers.

Step 5: Other software setup

Install TestPoint on the server computer.

Install the TestPoint Internet Toolkit on the server computer.

Install your Web browser on the client computer.

Open the Notepad editor and create a file with the following content, and save it as "hosts." (no extension) in the Windows directory:

```
server      192.168.0.1
client     192.168.0.2
```

Put this file on both computers.

This will allow you to use the name of the computer, rather than a number, to refer to the computer.

Summary

You have completed the setup and are ready to go to the next chapter and begin trying out the TestPoint Web Server, or the other components of the TestPoint Internet Toolkit.

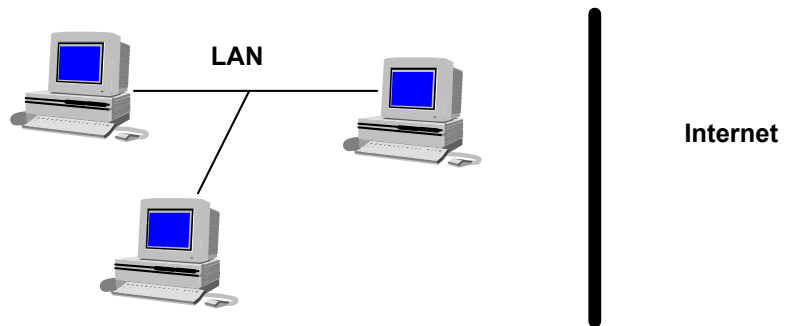
Note that the 2-computer network thus created is not appropriate for direct connection to the public internet (because it uses reserved "private-only" IP addresses), nor is it necessarily the correct setup for integration directly into your corporate network. **Consult your network administrator before connecting into a wider-area network.**

Network Administration, Gateways, Firewalls, and Security

The first task when you go to use the TestPoint Internet Toolkit on your full network is to determine your network setup. In many businesses, you will already have a **network administrator**, and you should contact that person regarding issues of computer configuration, IP addresses, and servers. If you do not have a network administrator, you can fulfill this role yourself by following the guidelines in this chapter.

No Internet connection

If your local network is **not** to be connected to the public internet:

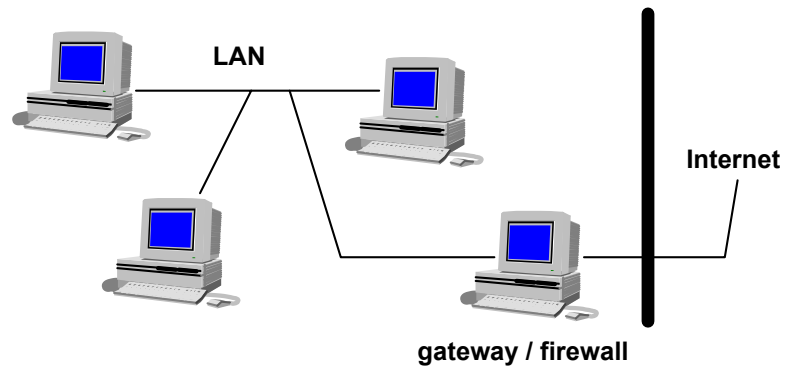


you can use private IP addresses, as described in the next section. You simply have to install TCP/IP protocol drivers on each computer that is to use these facilities, and assign each a unique address.

In this situation there is no access from outside the network, so security issues are only related to internal access. Anyone internal to the network who knows the IP address of a TestPoint Web Server can access it - but note that access is read-only, and only applies to the web pages and images you have defined for this server.

Internet via gateway

If your network is **connected** to internet via a **firewall** or other gateway computer that hides internal computers from the internet:

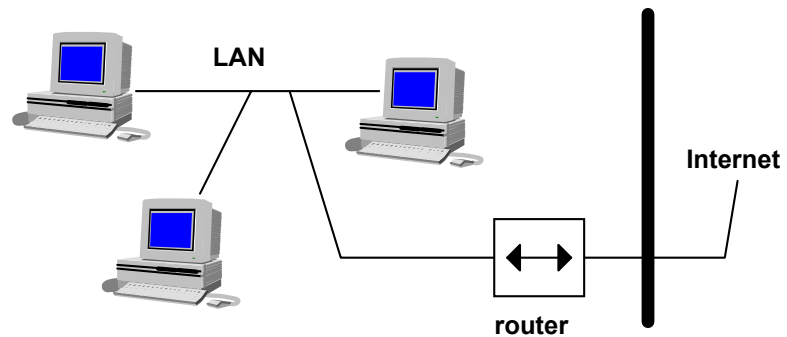


you can follow the same procedure, using private IP addresses. An example of such a setup would be a local network using Microsoft Network drivers, and one computer running an internet gateway program like WinGate (<http://www.wingate.net>).

Security issues here are generally the same as for a network with no internet connection. The firewall computer enforces your security from outside access, and it is up to your choice of firewall/gateway software and hardware. Note that some of these products may allow outside access with passwords and other features, so your own employees can access resources while travelling.

Directly on Internet

If you have one or more computers directly routed to the internet:



then each computer must have a valid, unique IP address that has been assigned for your use by InterNIC (ds.internic.net) or the current authorization agency.

Such computers are publicly addressable by anyone on the internet, so security becomes a major issue in this configuration. Generally, you do not put most of your computers directly on the internet this way, but instead use a firewall as shown in the previous section. You may, however, wish to have one or more publicly accessible servers connected directly. Again, the TestPoint Web Server provides only read-only access and only to the designated web page files and the currently running TestPoint applications.

Security

TestPoint Web Server does not execute programs or write to files. It provides only read access (the HTTP "Get" command), and only to files in its defined Web subdirectory and to screen images.

The E-mail object only sends messages.

The TCP/IP object is fully programmable, so its usage in an application and its security is up to you, the application developer.

Choosing Addresses

Each computer which will be using TCP/IP protocol or hosting or accessing a web server must be assigned a unique IP address.

Private IP addresses

The simplest situation is where your local network is either not connected to the internet, or is connected only through a gateway which hides the local IP addresses. In this case, you can assign IP addresses from the range:

192.168.0.1
to 192.168.0.255

Just make sure you keep the addresses different for each computer.

Public IP addresses

If your computers can be directly addressed, or for any computers that can be addressed this way (like your gateway computer), you must obtain an assigned IP address. The authority which assigns such addresses is the InterNIC (ds.internic.net), but most Internet access providers can obtain address assignments for you as part of their services.

Generally, if you have a number of computers, you can obtain an entire block of 256 addresses for your own use.

Another way to obtain addresses is to have one dynamically assigned to you. This is often the case when you have a dial-up connection to the internet. Your internet connection provider may assign you an IP address each time you connect.

Installing and Configuring TCP/IP in Windows

TCP/IP is just one of the network protocols you can install in Microsoft Windows. Depending on your local network configuration, you may already have one or more protocols installed.

TCP/IP protocol comes with Win95, Win98, and WinNT. You simply have to go to the Control Panel and double-click on the Network icon. Choose "Add" to add a new protocol, and select the Microsoft protocol "TCP/IP".

For Windows for Workgroups 3.11, you can obtain a free TCP/IP protocol driver from Microsoft at <http://www.microsoft.com/kb/softlib/mslfiles/TCP32B.EXE>

There are also other, third-party, providers of TCP/IP drivers.

When you have added the TCP/IP protocol, you will be given a configuration dialog that lets you enter your IP address. You can also specify the address of one or more DNS name servers (see Chapter 3) and a gateway computer (used when connecting to IP addresses outside your network - that is, on the Internet).

Testing a TCP/IP connection

There is a test program provided with Microsoft's TCP/IP drivers, called PING.EXE. This program may be used to test your TCP/IP installation, once you have at least two computers configured with TCP/IP protocol.

Just use the PING command together with the IP address of another computer. For example:

```
PING 192.168.0.100
```

You should get back an indication that a response was received, along with the response time. If you instead get a "Request timed out" message, recheck your TCP/IP installation and your IP address settings and network connections.

Note that the PING command will not work through some gateway/firewall combinations, but it should always work fine within your local network.

Configuring Web Browsers

You will also need to make sure your Web browser is set up correctly before you can access a local TestPoint Web Server. Even if you can access the Internet, you may need to modify your settings.

Netscape

Netscape has a setting under the Options/Network menu for "Proxies". A proxy, in this case, is a computer which will forward your web requests on to the actual destination. When you enter the name of a site, like www.microsoft.com, the proxy is contacted instead of directly trying to find that name (which, of course, does not exist on your local network).

If you are set for "No proxies", you will be able to access local servers directly by IP address or name.

If you are set for "Manual Proxy Configuration", you should view your settings and enter the IP addresses and ports for your local servers that will be running TestPoint Web Server into the box labelled "No proxy for:".

If you are set for "Automatic Proxy Configuration", you will need to check with your network administrator, because your proxy settings are coming from an external file.

Internet Explorer

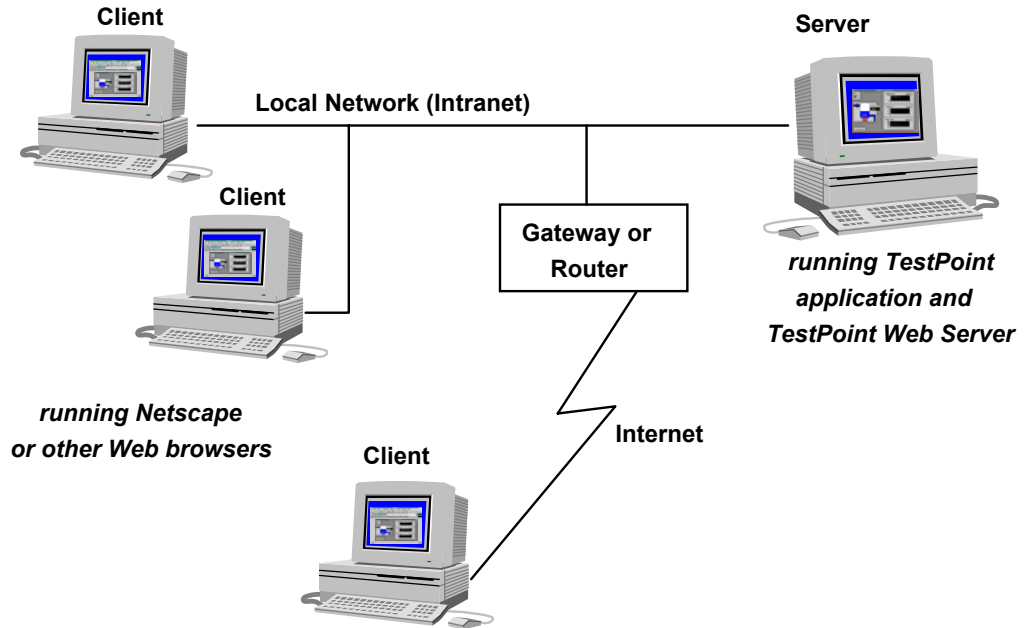
Internet Explorer also has an option for using "Proxies", as an alternative to directly accessing a server, or dialling a server on a phone line. This option is accessed through the View/Options menu.

If you are using a proxy server, you must go into your Proxy Settings and enter the IP addresses for your local server(s) that will be running TestPoint Web Server into the box under "Exceptions". You can also choose to check the box labelled "Do not use proxy server for local intranet addresses".

Chapter 5. TestPoint Web Server

Overview

The TestPoint Web Server is a program you can run on any Windows-based PC (Win 3.1 or higher), along with running one or more TestPoint applications, so that results and screens from those applications can be viewed on other computers.



There is no need to modify the TestPoint applications in any way in order to use them with the Web Server.

You can view entire application panels (whole windows), or any specific object on a panel, either in graphical format, or text format. You can also specify an update interval to dynamically update the image in the Web browser (only Netscape browsers currently support this feature, known as "server push").

Web Server Example

A web example is provided with the Internet Toolkit. To run this example, just double-click on the icon labelled "Example (use with server)", and also on the icon for the Web Server:



This will run a demo application written in TestPoint which simulates a process-monitoring application with multiple strip charts and numeric displays.



You'll need to make sure your network has TCP/IP protocol installed and configured (see Chapter 4 for further information).

Now go to another computer on your network and run Netscape.

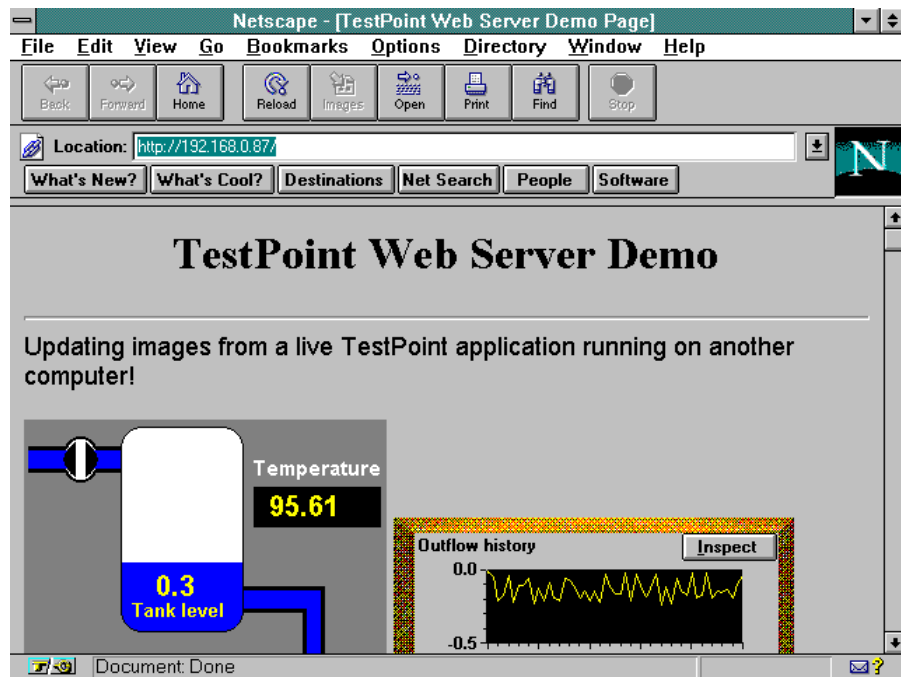


Make sure you have configured Netscape either to use "no proxies" or to specifically exclude the server computer you are trying to reach. Proxy setup is found in the Options/Network menu of Netscape.

Enter the correct address for the server computer (**which will depend on your specific network setup and addresses**). For example, to access our test lab server, we would enter:

`http://192.168.0.100`

You should see a screen similar to the one shown below, with displays that update every 4 seconds. (Note that the updates only occur if your browser supports "server push", which generally means using Netscape and not Internet Explorer).



If you have problems:

- Check the computer address you entered.
- Check that TCP/IP is setup and configured - see Chapter 4. Use the "ping" command to check the connection as described in Chapter 4.
- Check the Netscape setup to make sure "proxies" are disabled.
- Make sure both TPSEVER and the Control application are running on the server.
- Check that you have images enabled in Netscape.

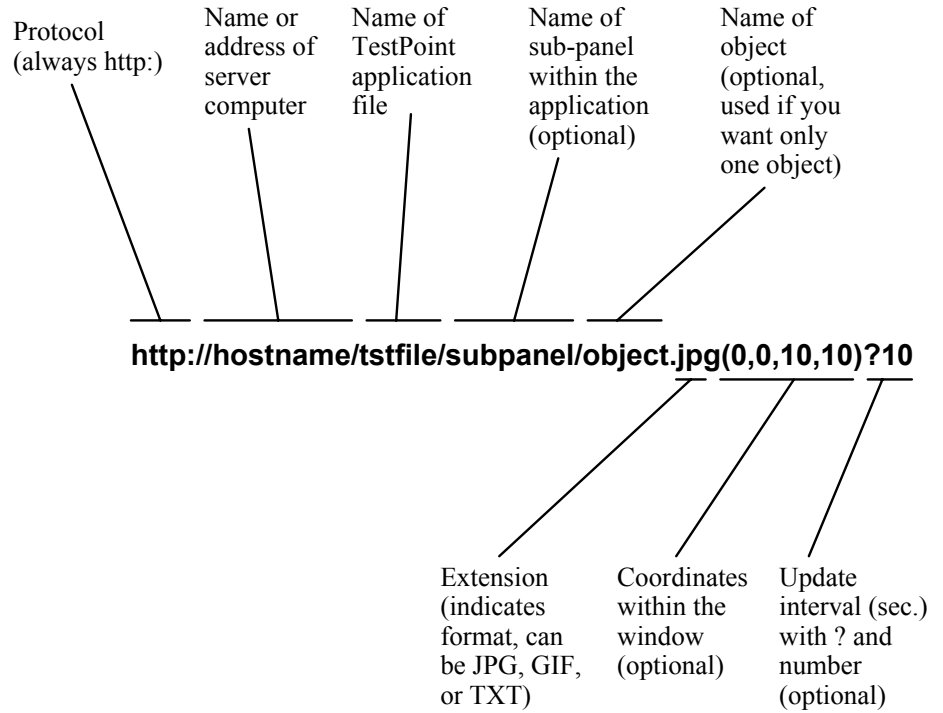
URLs - Names of Objects to View

A URL (Uniform Resource Locator) is simply the string you type into your browser to get to a specific web page or FTP site. An example of a URL is "http://www.ccc488.com". URLs generally consist of the name of an access protocol, such as HTTP, followed by a host, or computer, name, and further information that depends on the particular host and information being accessed. Most URLs just end with the pathname of a disk file containing a web page.

The TestPoint Web Server supports URLs referring to both web pages on disk, and also to TestPoint objects and panels. Here are a few examples:

http://host/tstfile.jpg	Entire panel of given application, JPG format
http://host/tstfile/object	One object in a TestPoint application
http://host/file.htm	A disk file containing a web page
http://host	The default, or home page, disk file on the server (index.htm)
http://host/image.jpg	A disk file containing a JPG image
http://host/tstfile?10	A TestPoint panel, updated every 10 seconds
http://host/\$screen\$	The entire screen of the host computer
http://host/tstfile/\$top\$(10,10,300,200)	A specific coordinate window within the top panel of a TestPoint application

URLs for TestPoint applications



The **tstfile** portion of the URL is the actual name of the TestPoint application file, without the .TST extension.

Subpanel and **object** names are the actual TestPoint object names (which normally show up as the captions for these objects).

Coordinates, if used, are in units of display pixels (dots).

Extensions indicate the desired format. TXT format returns the data value of the object as text. JPG or GIF formats return a screen image of the object (if it has a screen window). Note that choosing JPG or GIF does not guarantee that particular format, since some display setups and number of color choices can only be rendered by one format or the other. The default format is GIF.

URLs for disk files

A disk file URL is simply the path and filename, relative to the directory containing the web pages. This web "root" directory is defined as part of the TestPoint Web Server settings, and is normally C:\TESTPT\WEB.

A URL is interpreted to refer to a disk file only if the first part of the path name is **not** the name of a running TestPoint application file.

So, for example, the URL:

`http://host/mypage.htm`

would normally refer to the disk file:

`c:\testpt\web\mypage.htm`

Only files within the pre-defined web "root" directory can be accessed via the TestPoint Web Server, for security reasons. This keeps viewers from reading files in other places on the computer, or potentially on other computers in the network.

Default file URL (home page)

If you give a URL consisting of only the protocol and host computer name, a default home page will be returned. For the TestPoint Web Server, this will be the file **index.htm** in the web directory.

Special URLs

There is a pre-defined reserved name "\$screen\$", used to refer to the entire screen area of the host computer, to allow for complete screen dumps.

You can also use "\$top\$" to refer to the top panel of any TestPoint application. This is useful if you want to specify options such as window coordinates, as in this example:

`http://host/tstfile/top(10,10,300,200)`

URL examples to use with the supplied CONTROL.TST

(substitute the correct name or IP address for your server computer)

<code>http://server/control</code>	The entire main panel of the application
<code>http://server/control/Temperature</code>	Just the numeric display object named "Temperature"
<code>http://server/control/Temperature.txt</code>	The "Temperature" object in text format instead of graphic
<code>http://server/control/Temperature?4</code>	The "Temperature" object, updated every 4 seconds
<code>http://server/control/\$top\$(10,135,269,324)</code>	A rectangular section of the main panel of the application

Directly Viewing a TestPoint Panel - No Configuration Necessary

You can view the entire panel of any running TestPoint application remotely, simply by running the TestPoint Web Server on the same computer.

You just go to any remote computer, run Netscape, and enter a URL referring to the server computer and the TestPoint application filename, such as in the example below:

`http://server1/appl`

There is no need to create web page files, or know anything about HTML (HyperText Markup Language, the format for web pages).

Note the following items when attempting web server access:

- Both the server and remote browser computer must have TCP/IP protocol installed and correctly configured, with valid computer addresses. (See Chapter 4)
- If you wish to refer to the server by name, as in the example above, you must have a local "hosts" file or a name server. (See Chapter 4)
- Your browser should be configured to access the server computer directly if you are on the same local network. For Netscape, this means no "proxy" is used to access the server. (See Chapter 4)

Creating Web Pages for your Server

If you want to create customized displays to view in your browser, containing a mixture of text and graphics, or more than one TestPoint object, you will need to create your own disk-based web pages. Web pages are simply disk files containing text that conforms to the HTML (HyperText Markup Language) specification.

Some sample web pages are provided with the Internet Toolkit, and installed in the WEB subdirectory: index.htm, other.htm, and features.htm. You can modify these or create your own.

Web pages are most easily created by using an HTML editor. Many of these are available either free or for a low price, such as Microsoft Front Page, HoTMetaL, and others. You can also create HTML in any text editor if you enter the HTML keywords manually.

Here are some places to look on the internet for HTML editors and information (of course, web sites come and go - you may also want to do your own search on HTML):

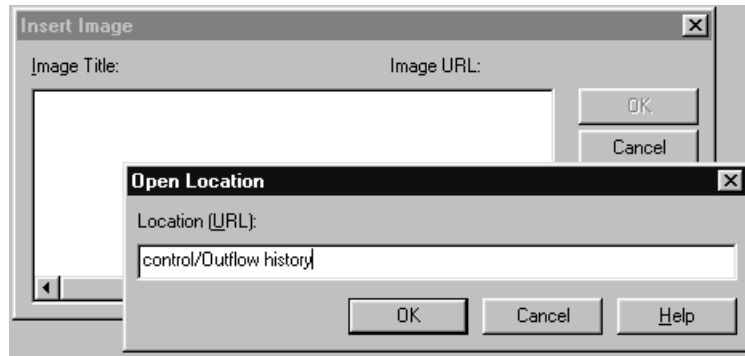
<http://www.microsoft.com/msoffice/frontpage>

<http://www.sq.com/products/hotmetal/hm-ftp.htm>

<http://sdg.ncsa.uiuc.edu/~mag/work/HTMLEditors/windowslist.html>

When creating web pages, just follow the normal rules explained in any book on HTML or the instructions of your HTML editor. Whenever you want to insert a reference to a TestPoint screen image, just insert an image reference and use the URL rules mentioned earlier in this chapter.

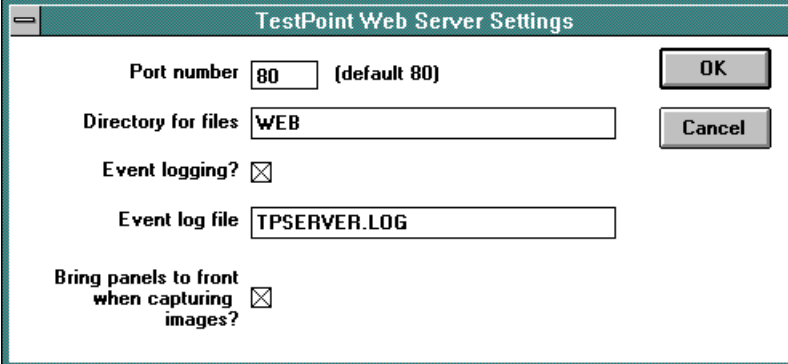
For example, in the Microsoft Front Page editor, you can insert an image reference to the "Outflow history" graph of the CONTROL.TST application using Insert/Image and this dialog box:



These references can leave off the "http://hostname/" portion of the URL because they are **relative** URLs, referring to items on the same server as the HTML file. This lets you use the same HTML file on any host computer without having to edit the image references.

Web Server Options

There are a number of configurable settings available in the TestPoint Web Server. These may be accessed through the Options/Settings menu of the server program:



The screenshot shows a dialog box titled "TestPoint Web Server Settings". It contains the following fields and options:

- Port number:** A text box containing "80" with "(default 80)" to its right.
- Directory for files:** A text box containing "WEB".
- Event logging?:** A checked checkbox.
- Event log file:** A text box containing "TPSERVER.LOG".
- Bring panels to front when capturing images?:** A checked checkbox.

Buttons for "OK" and "Cancel" are located on the right side of the dialog.

The settings are described below:

Port number

The TCP/IP port number (see Chapter 3) on which the server will operate. This should almost always be port 80, which is the default port for web servers. However, if you want to run the TestPoint Web Server on the same computer as some other web server, at the same time, you can put the TestPoint Web Server on any desired alternate port number. In that case, you must specify the port number when accessing the server, as in:

http://server:8080

Directory for files

Specifies the directory used for web files, when a file URL is received. This is always a subdirectory relative to the location of the TPSERVER program itself.

Event logging?

Specifies whether to log server events, such as connections and requests, to a file.

Event log file

Specifies the name of the event log file.

Bring panels to front when capturing images?

When this setting is checked, the server will force the TestPoint panel window to the frontmost position before any attempt to capture a graphic image from it. If you do not turn on this option, and you bring up some other window on the screen of the server that overlaps the application panel, then that window's contents will be captured instead.

Chapter 6. TestPoint E-mail Client

Overview

The TestPoint E-mail object lets you create e-mail enabled applications. These applications can send text e-mail upon any desired event, containing any combination of constant text and measured results.

For example, you might have an application that sends a summary of measured results to an e-mail distribution list every 8 hours. Or, you might want your application to send an e-mail upon an abnormal measurement condition.

The TestPoint E-mail object can use either the SMTP (internet) or MAPI (Windows) e-mail interfaces, providing flexibility in integrating into your particular e-mail setup.

E-Mail Example

Examples are provided with the Internet Toolkit. Go to the TestPoint editor and open the file EMAIL.TST, from the WEB directory. This application shows how to send an e-mail message when a pushbutton is clicked, with the action list:

```
1) Send          E-mail          to: "test@cec488.com",  
                                     subject="example from Internet  
                                     kit", message="Test message  
                                     text."
```

The application is initially set for MAPI (Windows-based) e-mail. Just edit this action line and change the **to:** parameter to a valid local e-mail address (such as your own), and run it. Or, if you are set up for internet e-mail access with an SMTP server, you can change the settings of the e-mail object and send internet e-mail (see the later section in this chapter for details on the object's settings).

Another example is provided in EMAIL2.TST, which shows how periodic summaries of measured results might be sent via e-mail.

If you get a warning message about needing to enable features when you open EMAIL.TST, refer back to Chapter 1 for installation instructions.

Using the E-mail Object

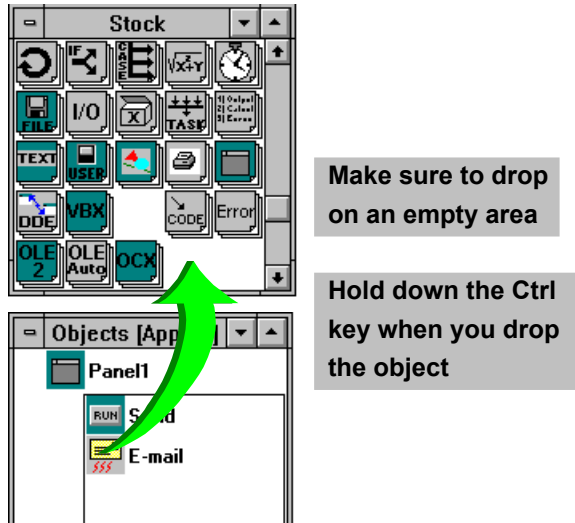
Copy and Paste

The EMAIL.TST example is also the place to go when you want to create your own applications. Simply follow these steps:

1. Open the EMAIL.TST example
2. Select the e-mail object
3. Use Edit/Copy
4. Close the EMAIL.TST example
5. Paste the object into your own application, using Edit/Paste

Adding the E-mail object to the stock

If you expect to be using this object frequently, you may wish to add it to the stock window of the TestPoint editor. To do this, simply drag the object from the "Objects" window to an **empty** area in the "Stock" window, hold down the Ctrl key, and drop it into the stock:



If you have only full rows of objects in the stock window, scroll the stock window all the way down, and then drop the object into the last pixel or two at the very bottom of the stock window.

MAPI vs. SMTP Mail Protocols and Addresses

There are two basic mechanisms for sending e-mail, the SMTP (Simple Mail Transfer Protocol, or internet) method, and the MAPI (Mail Application Programming Interface, or Windows) method.

SMTP protocol requires that your computer communicate via TCP/IP network protocol with an SMTP server computer. You will need to know the address and port number (usually 25) of your SMTP server to use this protocol. SMTP protocol is used for internet e-mail.

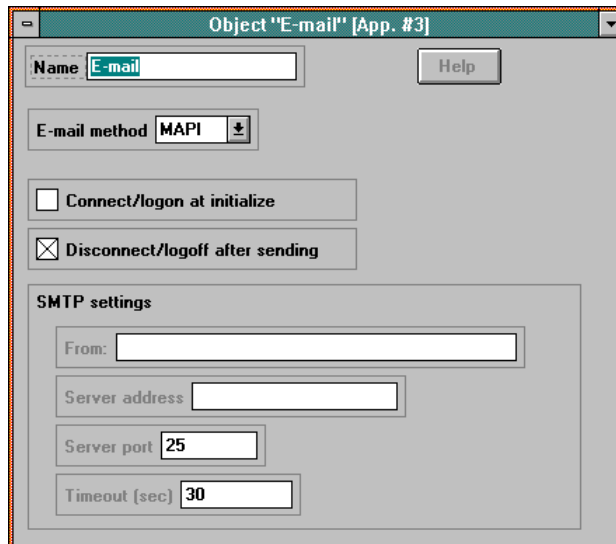
SMTP e-mail addresses are of the form **user@host.com** or **user@host.net**, or similar.

MAPI protocol is used by Microsoft Mail and Microsoft Exchange, which come with Windows. MAPI typically communicates with a local network e-mail "postoffice" server. MAPI mail can also communicate with internet and other forms of e-mail if your local network e-mail installation includes an "e-mail gateway" program for these forms of mail. In this case, you may be able to use internet-style e-mail addresses, but the MAPI protocol.

MAPI e-mail addresses are usually just names in your e-mail address book.

Multiple addresses can be separated by semi-colons.

Settings



Object "E-mail" [App. #3]

Name: E-mail Help

E-mail method: MAPI

Connect/logon at initialize

Disconnect/logoff after sending

SMTP settings

From: []

Server address: []

Server port: 25

Timeout [sec]: 30

E-mail method

You can choose MAPI or SMTP protocol. MAPI protocol is a set of built-in Windows function calls specifically for mail, configured through the e-mail setup of Windows. SMTP protocol is an internet protocol sent via TCP/IP network connections to a server.

Connect/logon at initialize

If this setting is checked, the E-mail object will connect or logon to the server immediately upon entering Run mode. This can potentially save time later when you execute a "Send" action to send your e-mail. However, it also holds a connection open to the server, and if your server has limited connections or resources you may wish to avoid this setting. This setting is off in the default setup provided in the Internet Toolkit.

Disconnect/logoff after sending

If this setting is checked, the E-mail object disconnects or logs off from the server as soon as a message is sent. (It will reconnect for the next message). This avoids holding network connections open. However, closing the connection also means having to re-open the connection for the next message, which may cause delay later. If you are sending lots of e-mail messages quickly, you may want to turn this setting off.

(SMTP) From:

For SMTP mail, you need to provide a "From:" address (your internet-style e-mail return address). Note that for MAPI, the sender is known based on your e-mail configuration in Windows.

(SMTP) Server address

For SMTP mail, you must provide the TCP/IP address of the SMTP server computer. This can be a numeric address or a computer name (if you have this name listed in a "hosts" file or have a name server - see Chapter 4 and/or ask your network administrator).

(SMTP) Server port

For SMTP mail, you must provide the TCP/IP port number of the SMTP server program. This is normally port 25.

(SMTP) Timeout

This setting provides the timeout used in the network communications for SMTP mail. It should be long enough to allow the server to respond. Note that in cases where you are accessing a server through an on-demand dial-up connection, you may have to set a fairly long timeout.

Chapter 7. TestPoint TCP/IP

Overview

The TestPoint TCP/IP object provides communications features that let you build client/server applications across your local network or the internet.

TCP/IP is the basic communication protocol of the internet (see Chapter 3). It can be used within a local-area network or between computers in different divisions and buildings of one company (an "intranet"), or even across the internet between any two computers.

With the TestPoint TCP/IP object, you can create a **server** application which waits for **clients** to connect to it, and then you can send and receive any desired stream of bytes (TestPoint string data) between computers.

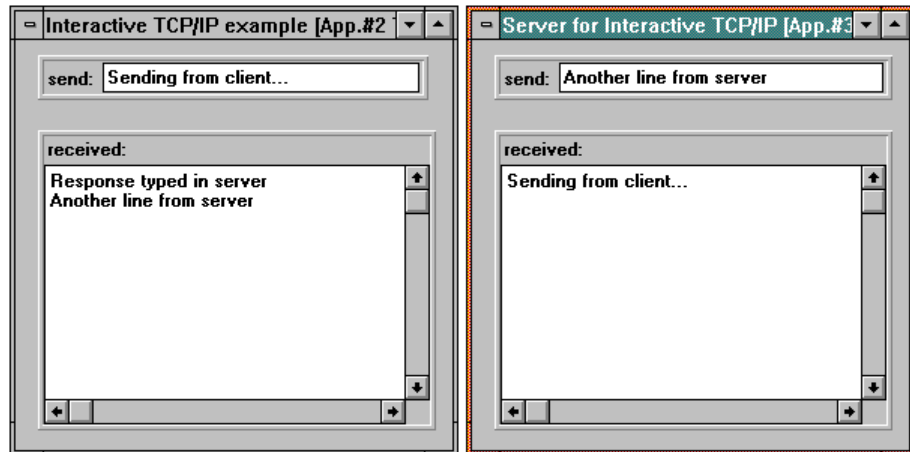
Applications could include:

- Remote monitoring and control panels for a TestPoint-based measurement system.
- Interconnected applications in distant locations that communicate data while operating.
- Communications between TestPoint-based applications TCP/IP-based measurement hardware, such as Keithley's SmartLink product family.

TCP/IP Example

A set of TCP/IP example programs are provided with the Internet Toolkit. To run these examples, you must have the TCP/IP protocol installed and configured on your system (see Chapter 4).

Begin in the TestPoint editor, by opening both the **TCPIP.TST** and the **CHATSERV.TST** examples from the WEB directory. Go into Run mode, and try typing into the "Send:" fields on each of the panels.



You will see that this pair of example programs provides a client/server "chat" capability, sending data strings between two programs. The examples are initially set up to run on the same computer, because the special server address of "127.0.0.1", which refers to "this computer" is used. If you want to run this pair of programs on two computers, simply run CHATSERV on one computer, then open TCPIP on the other and change the server address setting in the TCP/IP object to match the correct address of the server computer. Then you will see communication between two computers, using the same code!

If you get a warning message about needing to enable features when you open TCPIP.TST, refer back to Chapter 1 for installation instructions.

Using the TCP/IP Object

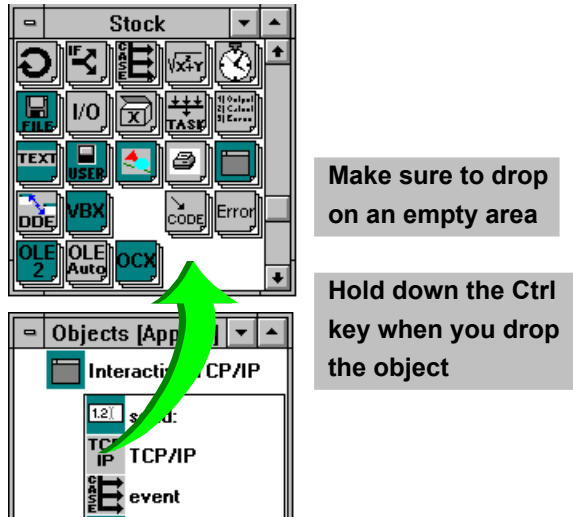
Copy and Paste

The TCPIP.TST example is also the place to go when you want to create your own applications. Simply follow these steps:

1. Open the TCPIP.TST example
2. Select the TCP/IP object
3. Use Edit/Copy
4. Close the TCPIP.TST example
5. Paste the object into your own application, using Edit/Paste

Adding the TCP/IP object to the stock

If you expect to be using this object frequently, you may wish to add it to the stock window of the TestPoint editor. To do this, simply drag the object from the "Objects" window to an **empty** area in the "Stock" window, hold down the Ctrl key, and drop it into the stock:



If you have only full rows of objects in the stock window, scroll the stock window all the way down, and then drop the object into the last pixel or two at the very bottom of the stock window.

Clients, Servers, and Ports

A **server** application is designed to accept connections from one or more client programs, usually running on other computers.

A **client** application initiates a connection to a waiting server.

Note that communication is bidirectional - both servers and clients can send and receive data.

Note also that a single TestPoint application can contain multiple TCP/IP objects, and be a client for multiple connections, or a server for multiple connections simultaneously.

A **port number** is a unique identifying number for a particular TCP/IP "place to connect to" on a given computer. Only one program (or TCP/IP object) can make use of any one port number at a time on one computer.

Settings

The screenshot shows a dialog box titled "Object 'TCP/IP' [App. #4]". It contains the following fields and options:

- Name:** TCP/IP
- Mode:** Client (dropdown menu)
- Port #:** 20000
- I/O timeout:** 5
- Open at initialize**
- Cause events on:**
 - Receive**
 - Connect/Close**
- Client-mode-only settings:**
 - Server address:** 127.0.0.1
 - Server port #:** 20001
 - Connection timeout (sec):** 10

Mode

The TCP/IP object can be configured for either **server** mode or **client** mode. In server mode, upon opening the connection the object goes into a "listening" state, waiting for clients to connect to it. One server can be connected to more than one client. In client mode, upon opening the connection, an attempt is made to link up with the selected server (see server settings below).

Regardless of mode, both client and server can send and receive data in either direction.

Port #

This setting specifies the TCP/IP port number used by this object. There can be only one program and object at a time using a given port on a single computer. Objects on different computers communicating with each other can use the same or different port numbers.

I/O Timeout

This setting gives the timeout period in seconds for output and enter operations.

Open at initialize

If this setting is checked, the TCP/IP connection will be opened immediately upon entering Run mode. If not, you may explicitly open the connection using the Open action. Note that connections are always closed upon exiting Run mode (at the termination of the application).

Cause events on

You can configure the TCP/IP object to cause an event, and thereby run its action list on **Receive** and/or **Connect/Close** occurrences. Enabling connect/close events allows events upon accepting a client connection (for server mode), completing a connection (for client mode), and closing of a connection (either mode). Enabling receive events will execute the action list whenever data is received over the connection.

Server address

For client mode operation, you must specify the TCP/IP address of the server you wish to connect to. This can be a fixed address setting, either numeric or a computer name (if you have names configured - see Chapter 4), or it can be set at run time by dragging this setting directly to an action list to create an action line that changes the setting, and then opening the connection as the next action line.

Note: if you use a server name rather than a numeric address, you may have to put that name in the local "hosts" file on your computer, because Microsoft TCP/IP may not access other name server computers to look up the name when making a connection.

Server port #

For client mode, you must specify the TCP/IP port of the server you wish to connect to. In the special case where the server is on the same computer as the client, the server port will always be different than the client object's port. If server and client are different computers (the more common situation), the port number can be any valid port.

Connection timeout

This setting specifies the allowed timeout period for a connection to complete, in client mode. If your connection may go through an on-demand dial-up connection or the public internet, you may wish to set a fairly long timeout period.

Actions

Open

This action opens the connection. However, if the connection is already open, no action will occur.

For **servers**, opening the connection means that the server becomes active in a "listening" mode, waiting for clients to connect to it. If you have enabled connect/close events, the server will run its action list whenever a client connects to it.

For **clients**, opening the connection actually communicates with the selected server, establishing a link between the computers.

Output to

This action sends data to the other computer. Data is always sent as a TestPoint STRING data type, which can be any series of data bytes up to 32767 bytes in length. If you give a string value as the data to send, it is of course sent just as it is. If you provide a numeric value, it is first converted to a string (for example, the value 3.52 is converted to the string "3.52"). A vector is converted to a string as well, with CRLF line separators between the values.

You can select any desired terminating sequence to be sent after the data. This can be "none" if you wish to send only the data, or you may wish to use CRLF, or CR, or some other unique terminating character.

Note that there are communications buffers between sender and receiver, so completion of this action does not mean that the other computer has actually completed reception of the data.

Enter from

This action receives data from the other computer. You can specify a desired maximum number of bytes to receive (up to 32767), and a terminating character sequence such as CR, CRLF, or "none". Input will stop on either the byte limit or the terminator, whichever occurs first. Received data is always a TestPoint STRING data type. If you wish, you may convert the result to other types using TestPoint math functions.

The received string becomes the value of the TCP/IP object.

Close

This action closes the connection, returning this TCP/IP object to an idle state.

Get # bytes available

This action sets the data value of the TCP/IP object to the number of input bytes available (in the incoming data buffer of this connection).

Test if connected

This action sets the data value of the TCP/IP object to 0 (false) or 1 (true) to indicate whether there is an open connection.

Receive bytes (low-level)

This action receives input bytes, similar to the Enter from action. However, this action directly accesses the lower-level TCP/IP function calls, and does not check for terminating character strings. It simply returns the available input data, **up to** the given number of bytes. It can return less than the chosen number of bytes, if less bytes are currently available, or even 0 bytes if no data is available.

This action may be more efficient in execution than the Enter from action for situations where higher data throughput is critical, because no additional buffering or string matching for terminating characters occurs. However, this action is also harder to use, because there is no obvious input record boundary (unless you have specifically programmed your client and server application to use fixed-length data transmissions).

Select client

This action may be used in server mode when there is more than one client connection. Normally, if you are processing input data in the action list of the TCP/IP object, in response to a received data event, your Enter from and Output to actions will automatically be routed to the client computer that caused the event. However, if you want to specifically select a client connection (by connection number), you can do so with this action. See the section on multiple clients in this chapter.

Get client connection list

This action may be used in server mode. It sets the data value of the TCP/IP object to a vector containing all the active client connection ID numbers. See the section on multiple clients in this chapter.

Events

The TCP/IP object gets events based on activity on the TCP/IP connection. The object settings allow you to control whether you want events on connection open/close occurrences, or on reception of data, or both.

Whenever a TCP/IP event occurs, the action list of the TCP/IP object executes, and the value of the TCP/IP object is preset to indicate what event occurred and on what connection ID number. The data value is a TestPoint LIST data type consisting of the ID number and an event code.

Possible event codes are:

- 1 Data received
- 8 Connection accepted (server mode)
- 16 Connection completed (client mode)
- 32 Connection closed (either mode)

So, for example, if connection number 4 just received some data, the value of the TCP/IP object when the action list executes would be a LIST containing two numbers: 4, and 1.

Handling only receive events

If you choose object settings with only Receive events enabled, and Connect/Close events disabled, you know that the action list will run only when data is received, so you can start right out with an Enter from action.

Handling all events

On the other hand, if you want to do processing when connections start and end as well as when data is received, you will need to check which event has occurred, so you can do the correct processing. This

is most easily done using the **Case** object, with the following formula setting:

select(tcpip,1)

which chooses just the event code information from the TCP/IP data value. Your action list would be of the form:

- 1) Select event tcpip=TCP/IP
- 2) When event is 1
- 3) Enter from TCP/IP up to 256 bytes, stop at CRLF
- 4) ...(do input processing here)
- 5) When event is 8
- 6) ...(do connection processing here)
- 7) End event

Handling Multiple Client Connections

TCP/IP servers can handle more than one client connection at a time. Each client connection is assigned an ID number.

Every connection is independent of the others. For example, client A might be sending a message to the server at the same time the server is sending to client B.

When you write a server application, you'll need to decide how you wish to handle multiple connections and conversations. It's just like talking in a crowded meeting - rules are needed to keep things organized.

One client / one server

The simplest client/server model is one-to-one, where you build your applications to only make one connection at a time, running only a single copy of the client application. CLIENT.TST and SERVER.TST show this. The server simply needs to respond to received-data events by sending back an appropriate response.

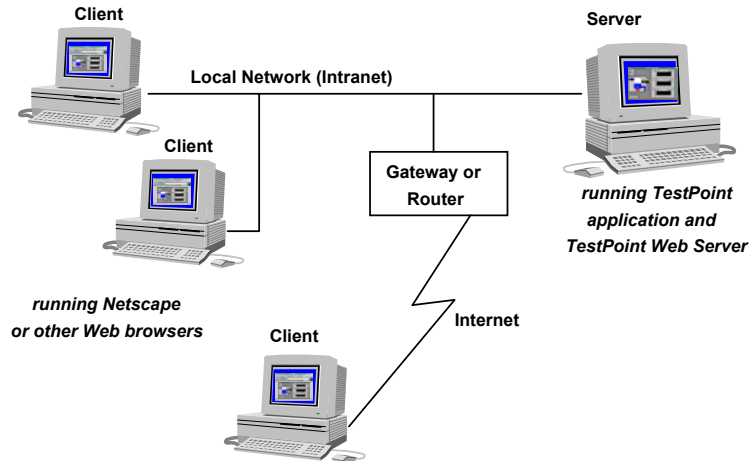
Multiple clients, query and response model

The next simplest model allows for multiple clients, but the server is designed so that it simply waits for input messages from the clients, and generates responses to these messages.

This situation is handled automatically by the TCP/IP object. The object itself, when in server mode, keeps track of all the client connections. When a receive event occurs (in this model, you would always want to enable events on receive in the settings), the TCP/IP object sets the "current client number" to the one that sent the message. In the action list for the TCP/IP object, when you execute an Enter from or Output to action, it will be directed to the correct client.

Multiple clients, general model

In a more general situation, you might want to send out information from the server to one or more clients, without receiving queries from them first. You might send these messages under certain measurement conditions, or to signal the clients about events occurring on the server system.



An example is provided in the file:

BRDCAST.TST

In this case, you will need to create your own logic for deciding which clients to route messages to. The TCP/IP object provides two actions to help you in designing such an application: **Get client connection list** and **Select client**.

Get client connection list returns a vector containing all the currently active connection ID numbers for clients. If you wish to send something to all clients, you can simply loop through these values, as in the BRDCAST.TST example

Select client sets the "current client" to which the Output to and Enter from actions will apply.

More Examples

SERVER.TST and CLIENT.TST

This pair of example programs work together, showing how you can create cooperating client/server applications in TestPoint.

The SERVER program continuously makes measurements (simulated), and waits for connections and requests via TCP/IP from clients. It allows clients to query measurement values and also set control parameters, depending on the message contents. A response is always sent to any incoming message.

The core of the SERVER example is just a Case statement for different incoming message command keywords.

The CLIENT program provides a user interface for retrieving the remotely measured values, and also for setting the control switches.

BRDCAST.TST

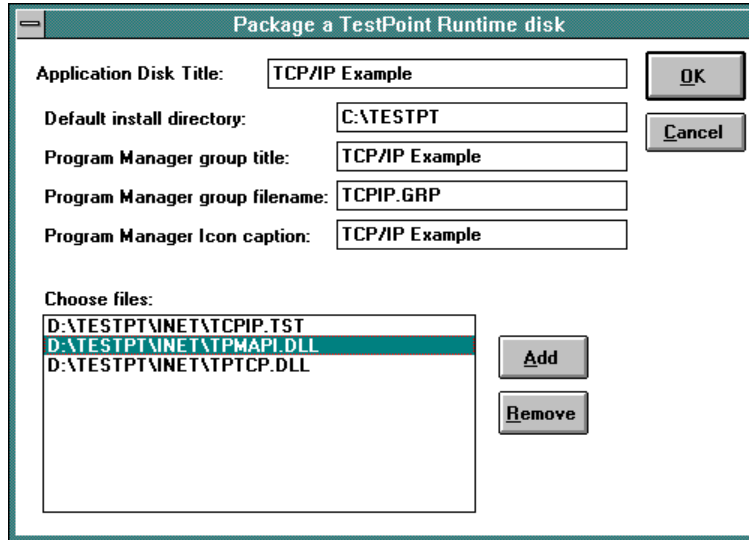
This example shows how a server can communicate with multiple client computers. You can run BRDCAST as the server on one computer, and then run multiple copies of TCPIP.TST on other computers as clients. When you press the "Broadcast" button on the server, all the clients will receive a message.

The core of this example is the use of the "Get client connection list" action, and a loop through all the clients using the "Select client" action and "Output to".

Chapter 8. Distributing Runtime Applications

If you build TestPoint applications containing either the **E-mail client** object or the **TCP/IP** object, and wish to distribute these to other computers and other users as runtimes, you can do so free of royalties or other restrictions.

All you need to do when packaging your application in the TestPoint Utilities/Make Runtime Disk dialog, is to use the "Add Files" button to add the necessary support files to your distribution disk:



The additional files you will need are:

TPTCP.DLL
TPMAPI.DLL

A

Address, 3-4, 4-10

Administration, 4-7

B

Browser, 3-7

C

Client, 7-6

Coordinate, 5-7

D

Disk file, 5-8

Distribution, 8-2

DNS, 3-4

Domain, 3-4

E

E-mail, 3-8

Editor, 5-11

Extension, 5-7

F

Firewall, 4-7

Format, 5-7

Front Page, 5-11

G

Gateway, 4-7

GIF, 5-7

H

Host Name, 3-4

Hosts File, 3-4

HoTMetaL, 5-11

HTML, 3-7, 5-11

HTTP, 3-7

I

Install, 1-2, 4-11

IP Address, 3-4

J

JPG, 5-7

M

MAPI, 3-8, 6-6

P

Port, 3-6, 7-6

R

Router, 4-7

Runtime, 8-2

S

Screen, 5-9

Server, 7-6

Setup, 1-2, 4-11

SMTP, 3-8, 6-6

Subpanel, 5-7

T

TCP/IP, 3-3, 4-11, 7-6

TXT, 5-7

U

URL, 5-6