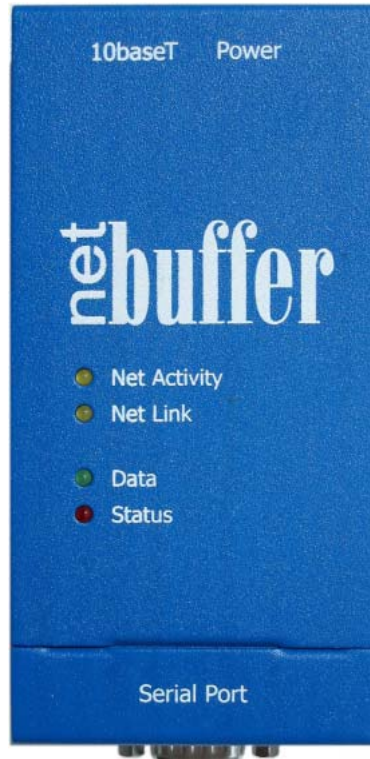




# NetBuffer Manual



Friday, 19<sup>th</sup> December 2003

Firmware Version 2.20 2003-Mar-17 and above

Issue 09

## Document History

Issue	Date	Comments
01	15-Sep-00	Initial Release. Pre firmware 1.13.
02	03-Nov-00	Release for firmware 1.13. Added SMTP size limit and "data pause" function, and in-bound TCP limiting.
03	10-Aug-01	Release for firmware 1.16. Added "Forced Linear" memory option, "bi-directional" option, WAP support, and IP in the user email trigger line. Page layout change.
04	22-Jan-02	Release for firmware 1.17. Added "if data" email option and the 1% full change. Added Telnet login options for TCP1. Minor text changes.
05	05-Jul-02	Release for firmware 2.01. Many changes. Release 04 preserved for 1.1x firmware.
06	23-Jan-03	Release for firmware 2.11. SNMP traps corrected and added.
07	01-Apr-03	Release for firmware 2.20. SNMP traps completely redefined.
08	22-Aug-03	Release for firmware 2.30. Addition of SMTP port. Reordered JavaScript variable list. Minor text changes.
09	19-Dec-03	Logo & page size changes. Added approvals and safety information.

Scannex Electronics Ltd, UK  
t: +44(0)8707 48 65 65  
f: +44(0)8707 48 67 67

<http://www.scannex.com> & <http://www.scannex.co.uk>  
[info@scannex.com](mailto:info@scannex.com) & [info@scannex.co.uk](mailto:info@scannex.co.uk)

# Table of contents

<b>Table of contents</b>	<b>iii</b>
<b>Safety</b>	<b>1</b>
Deutsch	1
Sicherheitshinweise	1
<b>Overview</b>	<b>2</b>
<b>Networking Overview</b>	<b>3</b>
Hardware Layer	3
IP Layer	3
UDP (User Datagram Protocol)	4
TCP (Transmission Control Protocol)	4
Other protocols	4
<b>Features of the NetBuffer</b>	<b>5</b>
Serial interface	5
Collection and delivery	5
Storage	6
Network	6
<b>Configuring the NetBuffer</b>	<b>7</b>
PC network configuration	7
Setting the IP	7
Method 1 – Use NBDDiscover	7
Method 2 – Use ARP/PING	8
Method 3 – Use the web	8
Checking the IP address	8
Configuring your web browser	9
Browsing the NetBuffer	9
Manual entry	9
Using NBDDiscover	9
What you should see	10
<b>Configuring the NetBuffer</b>	<b>11</b>
<b>Setup: Network</b>	<b>12</b>
<b>Setup: Web Password</b>	<b>13</b>
<b>Setup: Encryption</b>	<b>14</b>
<b>Setup: Operating Mode...</b>	<b>15</b>
Data Transfer	15

Serial -> TCP2	15
Serial -> FTP only	15
Serial -> email	15
TCP1 -> Serial	15
TCP1 -> TCP2	15
TCP1 -> FTP only	16
TCP1 -> email	16
Data pause	16
Source	16
Data	16
Memory	17
TCP1 IP+port	17
TCP2 IP+port	17
Live UDP output	17
RX/TX	18
Baud rate	18
Protocol	18
Autobaud	18
<b>Setup: Login TCP1...</b>	<b>19</b>
Rlogin servers	20
<b>Setup: FTP</b>	<b>21</b>
Port	21
Username	21
Password	21
Filename	21
<b>Setup: Email</b>	<b>22</b>
SMTP	23
Port	23
Domain	23
Username	23
Password	24
email to	24
Limit	24
Filename	24
Suffix	24

Every _____	24
...if _____	24
When _____	25
Quiet _____	25
Failures _____	25
<b>[Status/Index] menu item _____</b>	<b>26</b>
<b>[Engineer Menu] menu item _____</b>	<b>27</b>
Live Access: Activate _____	27
Live Access: Terminate _____	28
Reboot _____	28
Terminate: TCP1, TCP2, FTP, email _____	28
Memory: Wipe all data (V2.11+) _____	28
Memory: Clear flags (V2.11+) _____	29
Other: System Log _____	29
Up Time _____	29
Power _____	29
Reboots _____	29
Brown Outs _____	29
Traps _____	29
Events _____	29
STCs _____	29
Other: Internal Diagnostics _____	30
Other: Stack Dump _____	30
Other: Autobaud _____	30
<b>[Lock] menu item _____</b>	<b>31</b>
<b>WAP status _____</b>	<b>32</b>
<b>Suggested Configurations _____</b>	<b>33</b>
Collect RS232 data and deliver to a specific PC or NetBuffer _____	33
Collect RS232 data and wait for collection via TCP/IP _____	33
Create an RS232 link over IP with two NetBuffers _____	33
Deliver alarm notifications by email _____	33
Deliver periodic data by email _____	34
Stacking NetBuffers for larger memory or bridging _____	34
<b>Automated Software Considerations _____</b>	<b>35</b>
Web status page _____	35
Configuration by custom software _____	35

<i>email reference</i> _____	<b>36</b>
<b>NetBuffer LEDs</b> _____	<b>37</b>
<b>SNMP Trap reference</b> _____	<b>38</b>
<b>JavaScript variable reference</b> _____	<b>40</b>
<b>Web address reference</b> _____	<b>43</b>
Basic urls _____	43
POST command urls _____	43
<b>Common SMTP error codes</b> _____	<b>44</b>
<b>Power supply</b> _____	<b>45</b>
<b>Glossary of Network Protocols</b> _____	<b>46</b>
<b>Approvals</b> _____	<b>48</b>
FCC Rules Part 15 - Computing Devices _____	48
Industry Canada Regulatory Compliance Information for Class B Equipment _____	48
Australia and New Zealand users _____	48
EMI statement _____	48
European Union (EU) Statement _____	48
EMI/Safety Requirement _____	48
<b>Frequently Asked Questions</b> _____	<b>49</b>
Is data lost when using FTP? _____	49
When should I disabled autobauding? _____	49
When should the serial pin out be forced? _____	49

## Safety

- For North America and Canada use this product only with the provided UL-Listed and CUL-Listed NEC Class II power supply. Elsewhere use this product only with the provided power supply evaluated to Limited Power Source (LPS)
- Avoid contact with liquids and do not use if suspected damp
- Do not open the unit, no user serviceable parts inside
- Use indoors only

Note: Use of a DC power supply other than the one supplied with the NetBuffer voids the warranty and can damage the NetBuffer.

*A note about Power Connection, Surge Protectors, and lightning.*

Power surges on power lines, such as those caused by lightning strikes, can destroy or damage the NetBuffer. Therefore, we recommend that the DC Power supply is connected via surge protectors.

## Deutsch

Diese Endeinrichtung ist in Konformität EMC-Richtlinien  
89/336/EEC und 93/68/EEC

## Sicherheitshinweise

- Benutzen Sie dieses Produkt nur mit dem zur Verfügung gestellten Netzgerät, das zu begrenzter Energiequelle (LPS) ausgewertet wird
- Vermeiden Kontakt mit Flüssigkeiten
- Öffnen Sie nicht die Maßeinheit
- Nicht im Freien verwenden

## Overview

The NetBuffer provides a sophisticated way of collecting and delivering data across an Ethernet 10-baseT network, or across the public Internet.

Data can be collected from a serial (RS232) source, or a raw TCP/IP connection. Up to 16Mbytes of data is stored in the NetBuffer's non-volatile flash memory. That data can be collected by using an FTP client, or can be delivered by the NetBuffer to a TCP/IP connection, to the serial output, or by email attachment.

Even if the data is not being delivered by email, the NetBuffer can generate email messages that alert to specific conditions – data source problems, memory full, or a “quiet” data source.

The NetBuffer can optionally encrypt the data delivered using a 40-bit session key derived from a private secret of up to 96-bits. Configuration data can be protected using a ‘challenge/response’ mechanism based on the private secret.



## Networking Overview

This overview is by no means exhaustive! In order to make the network function it is built from several layers, which are simplified below:

### Hardware Layer

The hardware layer defines the electrical and basic means for transferring data. In the NetBuffer's case it uses 10-baseT. Physically you need to connect the NetBuffer by a CAT-5 style cable into a LAN hub, switch, or router.

In addition, the Ethernet electronics define the lowest structure of transmission of data. Devices on an Ethernet network have a "MAC" address that is a 48-bit (or 6 byte) globally unique address. One device can direct packets of data to another by using the destination's MAC address. Usually you do not need to know the MAC address of a device. For the NetBuffer, its serial number is the last three hexadecimal digits of its MAC address. The first three will always be "00-02-AE".

### IP Layer

The "Internet Protocol" is the next layer of protocol. This allows for a many-to-many connection of data between devices. Many different 'conversations' can be carried on between one device and many devices. It also provides a checksum protection on the data packets to allow detection of data loss.

This protocol requires the assignment of unique "IP addresses" to each device connected on a LAN, and indeed on the Internet as a whole. An IP address is usually shown as four decimal numbers separated by dots, as in "192.168.0.1". Groups of IP addresses are managed in a "subnet" – typically a collection of up to 256 computers. Devices require a "subnet mask" to be programmed to determine if a particular IP address is local (ie on the same physical network), or remote (for instance a machine on the other side of the world on the Internet). For a subnet of 256 computers the subnet mask would be "255.255.255.0". Addresses that are not local must be directed through a "gateway". The IP address of the gateway also needs to be programmed into the device.

Conversations are started between devices by asking "which MAC is responsible for this particular IP address". When, and indeed if, the reply arrives the device can physically direct the data through the hardware layer. This resolution of addresses is managed by the "ARP" protocol (Address Resolution Protocol).

Normally, on the Internet, devices are identified to humans by a name, as in "www.microsoft.com". However, this is just down to presentation – the computer or system still has to find out the IP address before it can converse. Normally the question "What is the IP address for this given name?" is handled by the DNS protocol (Domain Name System), or the NetBIOS name query service (for local networks). The NetBuffer can use these protocols too.

In addition to the IP address a device needs a "port number" to converse to. The port number is actually a number between 0 and 65535. So, in theory at least, one port on one device can hold 65536 concurrent conversations with any one other device. Of course, there may also be other concurrent conversations with other destination devices, and conversations from other ports!

The ports in the range 0 to 1023 are either assigned to specific services, or are used by well known mechanisms that are published in the "RFC"s of the Internet (Request For Comments). For instance, the web based HTTP services are assigned port 80, while FTP is assigned port 21. Outbound email services normally use port 25, while inbound email services typically use port 110. The other port numbers can be used for you own uses.

On top of IP is normally one of two protocols: UDP or TCP.

## UDP (User Datagram Protocol)

The UDP, or UDP/IP, protocol allows the transfer of chunks of data that are sent but not confirmed. The data is sent from one device to another, or to a group (as in the case of a "broadcast"). The device that sent the UDP data packet has no knowledge that the data was actually received, or even that the other machine is even there!

That being said, UDP is very useful in many circumstances. Many other protocols use UDP as the means to communicate. For instance, DNS uses UDP packets. DNS itself handles the timing mechanisms that allow for retransmission and timeouts.

Some other networked buffer products use UDP as the main means of data transfer, perhaps with a custom protocol managing the timeouts and negotiations. However, UDP transmission over the Internet, and over some WANs, can get very slow – particularly for large data transfers. The NetBuffer, on the other hand, can use UDP for a local "live" presentation of the incoming data but uses TCP for transferring the data.

## TCP (Transmission Control Protocol)

TCP, or TCP/IP ("Transmission Control Protocol over Internet Protocol"), is a negotiated conversation between two devices. It involves the request of one device asking for a conversation of another, and negotiating sequence and acknowledgement numbers.

TCP itself has the necessary timeout and retransmission mechanisms to make it 'reliable', and can handle packets that are lost, delayed, or even duplicated in transit along the network. Because of these properties, TCP/IP is "the" protocol that makes the Internet work the way it does.

Incidentally, you may hear of "sockets" mentioned in connection with TCP/IP communication. A socket is the combination of IP address and port of one device plus the IP address and port of another. When the socket is established, data can be transferred in both directions.

## Other protocols

Many other higher level protocols use TCP/IP as the means to transfer data. Such protocols as HTTP (as used for web browsing), FTP (for transferring data), and SMTP (for email) all use TCP/IP.

These other protocols, and more that make the system function, will be described in more detail in this manual.

## Features of the NetBuffer

### Serial interface

The NetBuffer has a single RS232 V24 interface that has a D-9 male connector. The port is automatically detected for DCE/DTE configuration and outputs handshake control signals on the appropriate pins. The NetBuffer also “knows” if there is no physical connection, or if there is a short on pins 2 and 3. This detection process is done electronically, and does not require data to make the decision.

In situations where there is no receive pin, or where there is data on both pins, the NetBuffer can be programmed to remain in DCE or DTE mode. For example, if the NetBuffer is used to only send serial data to a device that has no transmit itself, or where the NetBuffer is connected into a Y-lead configuration, the DCE/DTE needs to be fixed.

Complex algorithms and software allow the automatic detection of baud rate and parity configuration. The NetBuffer supports baud rates between 300 and 115200, and parity/word lengths of 7 even, 7 odd, 8 even, 8 odd, 8 none (note that “7 none” is not supported unless data arrives with 2 stop bits). This process requires a pause of 16 bit times (about 1½ bytes of data) between the bit-width measurement and the parity detection.

Where autobauding is not required, this facility can be disabled. One situation that requires this is where the NetBuffer is transmitting serial data into a PC COM port. Many operating systems do a “probe” of the serial port on boot-up – such a probe would trigger the autobauding process, but not give enough data to complete it.

Hardware flow control is supported. If the input controls are physically not connected, the NetBuffer takes them as “asserted”. DSR (Data Set Ready) and CTS (Clear To Send) have to both be asserted before the NetBuffer will transmit data. On the other hand, the NetBuffer always asserts DTR (Data Terminal Ready) through a “soft-drive”, while controlling RTS (Request To Send) to indicate that it can or cannot accept data.

A pulse can be issued on the RTS line via the web interface. This pulse can be in multiples of 50ms, ranging from 50ms to 12.75 seconds. See the URL “RTS*nnn*”.

The data flow is controlled by a timed hardware handshake. If CTS is unasserted for less than 5 seconds and then asserted, data transmission on the RS232 port will resume after 500ms. However, if CTS is unasserted for more than 5 seconds, data will resume 5 seconds after CTS remains asserted. If either CTS or DSR are unasserted for more than 20 seconds, the serial port is considered “inactive”, and an error trap is generated.

When using baud rates up to 19200 it is not normally necessary to have hardware flow control enabled. However, when using the higher baud rates CTS/RTS flow control is essential to prevent missed data while using the network functions of the NetBuffer.

### Collection and delivery

The NetBuffer allows collection from serial or from a TCP/IP source. Incoming data is placed in storage. At the same time data can be delivered to the serial port, to TCP/IP or email. Alternatively, the data can be collected from the NetBuffer by means of FTP.

While the destination is connected to TCP/IP or serial it is possible to communicate out to the source. This provides a two way link that is effectively buffered by the flash memory size in one direction, and unbuffered in the other direction.

## Storage

The NetBuffer has a large flash memory that provides 10 year retention without the use of batteries or power supplies.

The memory is arranged as a single, circular memory. If the memory is filled to the brim, it will erase a chunk (16k) of the oldest data to make room for new data from the source. If the delivery destination is connected (TCP/IP or serial) the memory becomes "linear", that is any new data will be lost if the storage becomes full. If the connection to the destination is dropped, the memory reverts back to the "circular" mode.

The NetBuffer can also be configured as a purely "linear" memory. When the memory is full, new data is lost. This is especially useful when 'stacking' NetBuffers or providing two NetBuffers across a network link (where the one nearest the data source can be in the normal, automatic circular mode, and the second in forced linear).

**Warning** As a result of the nature of flash storage, the figure reported as 'used' within the status web page and the FTP directory may include pages of flash that are reserved or defective. As a result, the figure may be larger than the data actually stored. However, when the data is transferred it will be the correct size.

## Network

All configuration and status information is accessed through the network interface. Most information is provided through a JavaScript web-based interface, allowing industry standard software to be used.

It is possible to configure all the settings with standard tools, however, we provide a useful tool for Windows<sup>®</sup> platforms, named NBDDiscover, for configuration of all NetBuffers on the local area network.

The list of protocols supported is: ARP, ICMP (ping), UDP, DHCP, DNS, TCP, FTP, HTTP, SNMP (traps only), SMTP, and NetBIOS (name service queries only). How these protocols are used will become clearer in discussing how to configure the NetBuffer.

# Configuring the NetBuffer

## PC network configuration

In order to communicate with the NetBuffer your PC needs the TCP/IP protocol installed, and a suitable Ethernet adapter and hub or switch.

In addition, it is preferable that NetBIOS services be enabled over TCP/IP on your PC. This will allow communication with a NetBuffer based on its device name, rather than its IP address. That device name can be programmed in the web-based setup screens and gives a more 'human readable' form to the NetBuffer's address.

## Setting the IP

To communicate with the NetBuffer it should be brought into the same subnet as the machines that will be talking to it. So the first step is to identify what IP and subnet your PC is running.

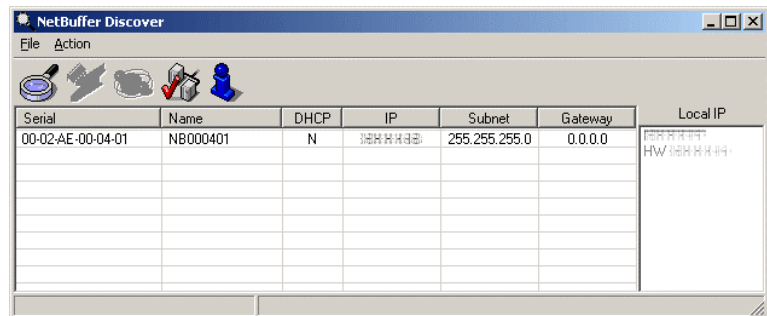
On most PCs, call up an MS-DOS command prompt and type "IPCONFIG", or "IPCONFIG /ALL" to show all installed network adapters, their IP addresses and subnets. If you are using Windows 95, type "WINIPCFG" to display the settings. Make a note of these details, or obtain the information from your network administrator.

There are three ways to set the IP address.

### Method 1 – Use NBDiscover

For Windows® machines (Win95, 98, NT4, 2000, XP) use the NBDiscover.exe. Locate the application and run it (it does not need to be installed or configured). This tool shows all NetBuffers on the local network with their serial number (MAC address), device name, DHCP flag, IP, subnet and gateway. It has the advantage of being able to 'see' all NetBuffers, even if they are not currently on the PC's subnet. However, it will **not** be able to see NetBuffers that reside on another network connected through a router or gateway. (In addition, some managed switches may be programmed to not pass on the UDP broadcast requests.)

Press the magnifying glass, or F5 key, to locate all connected NetBuffers. Double click with the mouse, or use the arrow keys and press Return/Enter to edit the settings for the selected NetBuffer.



Serial	Name	DHCP	IP	Subnet	Gateway	Local IP
00-02-AE-00-04-01	NB000401	N	192.168.1.10	255.255.255.0	0.0.0.0	192.168.1.10

If the LAN has a DHCP server (Dynamic Host Configuration Protocol) you can select "Use DHCP" and the NetBuffer will ask for an IP, gateway address, and subnet mask. If the server is configured to reply with extra details (SMTP addresses etc) these will also be accepted by the NetBuffer. While the NetBuffer is waiting for a response from the DHCP server the Data and Status LEDs will blink twice (Data-Data-Status-Status-Data-Data...).

If you wish to assign an IP address manually, enter the address in the dialog box and press "Program". The NetBuffer will assume this address immediately. Be sure to assign a unique IP address for your network!

**Warning** The NetBuffer will only accept changes within the first five minutes of being powered up. After that time all change requests are refused.

## Method 2 – Use ARP/PING

If you are not using a Windows<sup>®</sup> machine you can use this method. It involves telling the computer you are using that "MAC address = IP address", and performing a ping to the NetBuffer. If it has been less than 5 minutes since the NetBuffer has been powered up the change will take effect.

1. Obtain a command prompt
2. Type: `ARP -S ipaddress MACaddress`
3. Type: `PING ipaddress`
4. After a successful ping, type `ARP -D ipaddress`

Where *ipaddress* is the IP address you want the NetBuffer to become, and *MACaddress* is the NetBuffer serial number/Ethernet MAC address.

e.g. "ARP -S 192.168.0.234 00-02-ae-00-00-01", followed by "PING 192.168.0.234", followed by "ARP -D 192.168.0.234".

You can view which MAC addresses are currently in the computer's cache with the command "ARP -A". Some confusion can arise if there are multiple IP addresses for the same MAC address – it is best to delete any duplicates with the "ARP -D *ipaddress*" command, or wait for the ARP cache to flush.

**Warning** The NetBuffer will only accept changes within the first five minutes of being powered up. After that time all change requests are refused.

## Method 3 – Use the web

This method has limited uses. If you know the IP address of the NetBuffer, and your computer can talk to it, you can use a standard browser to change the IP address with the "Network" web page.

Any changes made via the web interface only come into effect when the NetBuffer is rebooted.

It is possible to set your computer to talk directly to the NetBuffer, even if it is on another subnet, by using the "ROUTE" command utility. It is much easier to use the NBDDiscover tool if at all possible.

## Checking the IP address

Use the PING utility to see if the NetBuffer is seen by the PC. Call up a command prompt and type: "PING *ipaddress*".

e.g. "PING 192.168.0.1"

You should see the replies listed – indicating that the PC can see the NetBuffer.

## Configuring your web browser

Before you talk to the NetBuffer it is worth checking your browser settings. In particular, check the "proxy" settings. Some machines, particularly in an office or enterprise environment, will be set to use a proxy server to allow multiple machines to use a single connection to the Internet or WAN. In order to talk to the NetBuffer it is sometimes necessary to tell the browser to avoid using the proxy server.

For Microsoft's Internet Explorer, select from the menu "Tools" and then "Internet Options". Select the "Connections" tab. If you are using IE5+, press the "LAN settings" button.

If the check box "Use a proxy server" is checked, make sure that "**Bypass proxy server for local addresses**" is checked. Then select the "Advanced" button, and make sure that your subnet is entered in the exceptions list box. For instance, if your machine is 192.168.0.1 and your subnet is 255.255.255.0, enter 192.168.0.\* in the exceptions list.

Netscape Navigator users, select the menu "Edit", then "Preferences". Under the "Category" list, expand out the "Advanced" item, and select "Proxies". Then in the right hand pane, if the "Manual proxy configuration" is checked, click "View". In the exceptions list, enter the IP exceptions as for Microsoft Internet Explorer.

## Browsing the NetBuffer

### Manual entry

In the browser's location or address line, enter "http://*ipaddress*", or "http://*devicename*" (if you have NetBIOS enabled).

e.g. "http://192.168.0.223", or "http://mainpbx".

### Using NBDDiscover

Alternatively, run NBDDiscover. Then select the NetBuffer you require, and press the world/web icon, or press Ctrl-F10. NBDDiscover will then automatically run your default browser with the IP address listed for that NetBuffer.

## What you should see

In either case you should see a page headed "Status", with a menu of links along the top of the screen. The title of the web page will show the NetBuffer name and serial number. The web page should be similar to the following (for firmware 2.00+ only!):

The screenshot shows a web interface for a NetBuffer device. At the top, there is a menu bar with links: [Status/Index], [Email Now!], [Lock], [Engineer Menu], and [Setup Menu]. The page title is "NB000401" 00-04-01. The main content is titled "Status" and is divided into several sections:

- Source:** Serial Port (Input & Output, Binary). A table shows settings: Baud (19200), Protocol (70), DSR (1), CTS (1), RX pin (0), and Autobaud (0). Below the table, it says "0-00:09:56 since data".
- Up time:** 0-00:09:56
- Bytes:** 0 / 4186112 (0%)
- Status:** Circular
- Destination:** email. A table shows: Active (0), Last State (-), Code (0), and Next ID (00000000).
- Encrypted:** (no data shown)
- FTP:** A table shows: Active (0) and File size (0).

At the bottom, it says "Version 2002-Apr-18 2.00, built 2002-04-19 10:10".

**Note:** The top menu bar changes according to the setup of the NetBuffer. [Status/Index] is always visible, [Email Now!] is only visible when the email service has been configured. [Engineer Menu] and [Setup Menu] are not visible when the NetBuffer has been "WebLocked".

You need to have JavaScript (this is distinct from "Java") enabled to view the pages correctly. Normally this is enabled by default and works with most browsers installed today, and all browsers that are current. The background will be pale green during the first five minutes of powering up, then revert to grey when locked.

**Note:** The default username and password for the protected web pages is "admin" and "secret" (case-sensitive).

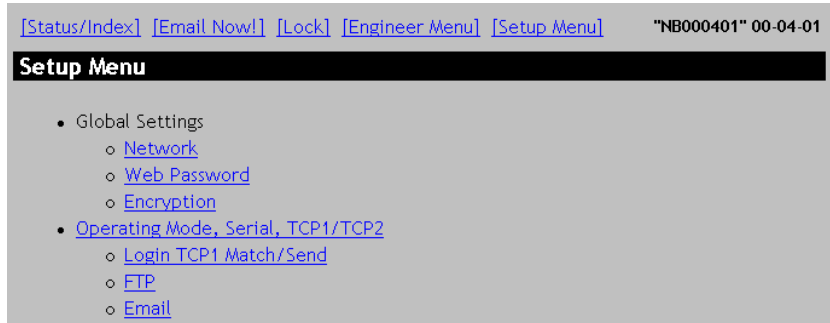


## Configuring the NetBuffer

<http://.../sm>

The NetBuffer has a menu item “[Setup Menu]” that includes all the configuration pages that need setting up. Please note that the Encryption setup can only be accessed within the first five minutes of a reboot/powerup.

When setting up the NetBuffer for the first time, follow the list, from top to bottom. Configure the ‘Global Settings’ first: Network, Web Password, and Encryption.



When those pages have been configured, move onto the Operating Mode, Login TCP1, FTP, and Email settings.

## Setup: Network

<http://.../network>

This page allows the viewing and setup of the device name, DHCP setting, IP address, subnet mask, gateway, SNMP destination, and two domain name servers.

The Name should be unique in the LAN. This is used for NetBIOS name queries from a PC or another NetBuffer.

The NetBuffer does not check to see if the name you enter is unique – that is up to you! In addition, the Name is used in the source email address for any SMTP deliveries (hence, do not use the “@” character within the NetBuffer Name field).

Changes to the Name, and the primary and secondary DNS servers become effective immediately. However, the other network based settings come into force when the NetBuffer is next rebooted.

If an SNMP destination is entered as a name (rather than a dotted IP address) that requires DNS resolution, the NetBuffer will check the name-to-IP mapping every 10 minutes.

The screenshot shows a web browser window displaying the 'Network' configuration page for a device. At the top, there are navigation links: [Status/Index], [Email Now!], [Lock], [Engineer Menu], and [Setup Menu]. The device identifier 'NB000401' and version '00-04-01' are shown in the top right. The page title is 'Network'. The configuration fields are as follows:

Name	NB000401	Unique identifier
IP	<input type="radio"/> DHCP <input checked="" type="radio"/> Static	DHCP=automatic IP assignment
IP address	192.168.0.234	IP address (changes effective on powerup)
Gateway	0.0.0.0	IP address of the gateway. Default=0.0.0.0
Subnet	255.255.255.0	Default=255.255.255.0
SNMP	255.255.255.255	Name/IP address to get traps. Default=255.255.255.255
Domain Name Servers		
DNS 1	255.255.255.255	IP address
DNS 2	0.0.0.0	IP address

At the bottom of the form are 'Save' and 'Cancel' buttons, and a version string: 'Version 2002-Apr-18 2.00, built 2002-04-19 10:10'.

## Setup: Web Password

<http://.../pwd>

This web-page sets the user name and password for accessing the protected NetBuffer web pages. These details are separate from the FTP user name and password.

[Status/Index] [Email Now!] [Lock] [Engineer Menu] [Setup Menu] "NB000401" 00-04-01

### Web password

New details (case sensitive)

Username

Password

Version 2002-Apr-18 2.00, built 2002-04-19 10:10

The defaults are "**admin**" and "**secret**".

The first five minutes after power up are considered "open", and any username will be accepted. This allows resetting of the username/password. If you need to 'lock' the NetBuffer, as if five minutes had elapsed, use the "Lock" web-page.

All user names and passwords are **case sensitive**.

# Setup: Encryption

http://.../pk

[Status/Index] [Email Now!] [Lock] [Engineer Menu] [Setup Menu] "NB000401" 00-04-01

## Encryption & Authentication

Web  Locked  Normal *Default=Normal*

Features

Secret  *24 digit hex, write-only*

Version 2002-Apr-18 2.00, built 2002-04-19 10:10

**Note** This page is only accessible during the first five minutes of a reboot. The background will be pale green during this time.

This is the setup page for encryption and authentication options.

**Note** Implementation, use, and setup guides for encryption and authentication are available in the separate document "*NetBuffer Encryption and Authentication.*"

- The "Web" option, when 'Locked' will prohibit changes to the NetBuffer unless an unlock response code is given. Use this facility when the web pages are accessible over the Internet, or another public LAN/WAN.
- The "Features" selection allows a choice of:
  - o "Disabled" - Sends data 'as-is'. Use for compatibility with firmware 1.xx, or where encryption is not required or allowed.
  - o Authenticate & Plaintext - Sends (and requires) 40 byte encryption header, but after that sends the data 'as-is'. Use where data encryption is not permitted (e.g. for legal reasons).
  - o Authenticate & Encrypt - Sends (and requires) 40 byte encryption header, and then sends 40-bit encrypted data.
- The "Secret" edit box, if blank, will leave the internal private secret unchanged. To modify the secret, enter a large hexadecimal number, up to 24 digits. E.g. "6542fda78bc0f746dc3b7a23". A 24 digit hex number corresponds to a 96-bit secret (4 x 24).

**Use** The encryption and authentication technology requires licensing.

## Setup: Operating Mode...

<http://.../mode>

This is perhaps the most important page for setting up the NetBuffer.

You are presented with a "Data Transfer" mode, settings for TCP1, TCP2, Live UDP, and the serial port baud rate and protocol.

[Status/Index] [Email Now!] [Lock] [Engineer Menu] [Setup Menu] "NB000401" 00-04-01

### Operating Mode

Data Transfer: Serial -> email *Left side has the buffer*

Data pause: 0 *x 50ms. 0=off*

Source:  Input & Output  Input only

Data:  ASCII  Binary *ASCII strips D7*

Memory:  Linear  Circular *Default=Circular.*

IP settings

TCP1 IP+port: [ ] :2001 *Active=name/ipaddress.Listen=blank/#name/#ipaddress*

TCP2 IP+port: [ ] :2002 *Port=0 disables*

Live UDP output: [ ] :0 *Port=0 for no live output*

Serial port settings

RX/TX: Auto *Default=Auto*

Baud rate: 19200 *bits per second*

Protocol: 7 Odd *data length and parity*

Autobaud:  Enabled  Disabled *Default=Enabled*

Save Cancel

Version 2002-Apr-18 2.00, built 2002-04-19 10:10

## Data Transfer

There are 7 modes of operation:

### Serial -> TCP2

Collects data from the RS232 port and stores in the flash memory buffer. It will allow FTP collection (if enabled – see the "FTP" web page) or will deliver the data to a raw TCP/IP connection. That connection may be to another NetBuffer or a PC.

Depending on the settings for TCP2 the NetBuffer can actively try to deliver the data to the far machine, or sit and wait for another machine to connect to the NetBuffer.

### Serial -> FTP only

Collects data from the RS232 port and waits for it to be collected by an FTP client.

### Serial -> email

Collects data from the RS232 port and delivers it by email. Various triggers can be set in the "email" web page for delivery. FTP collection or inspection is also possible.

### TCP1 -> Serial

Collects data from a raw TCP/IP socket, stores it, and outputs to the RS232 port. FTP collection or inspection is also possible.

### TCP1 -> TCP2

Collects data from a raw TCP/IP socket, stores it, and outputs to another raw TCP/IP socket. FTP collection or inspection is also possible.

## TCP1 -> FTP only

Collects data from a raw TCP/IP socket and stores it, ready for collection by FTP.

## TCP1 -> email

Collects data from a raw TCP/IP socket and delivers it by email. FTP collection or inspection is also possible.

If you are unclear which mode to use some suggested configurations are discussed later in this manual.

## Data pause

The NetBuffer can be programmed to look for a pause in the incoming data stream before 'sealing' the pointers. This is useful in situations where data arrives in packets, records, or chunks and you wish to only see whole chunks when using FTP or email delivery.

The value represents the number of 50ms timer ticks before sealing the pointers. The minimum suggested value is 2, while the maximum is 240 – corresponding to 12 seconds.

Set a value of zero to indicate no data pause detection.

## Source

(Default is "input only") Selecting "input & output" will allow data to be transmitted to the data source, as well as received. When set to "input only", the NetBuffer will only collect data.

Additionally, if the data source is a serial device, and it has unasserted DSR (Data Set Ready) to indicate 'do not send anything to me', selecting "Input Only" will ignore this handshake line (along with the CTS line) when considering if the device is 'alive'. When the option is set to "Input & Output", both CTS and DSR **must** be asserted (or not connected) for the red status light to extinguish, and for the NetBuffer to consider the device active and alive (for the SMTP traps and the email alerts).

This option only has effect when delivering to TCP2 or Serial, since email and FTP cannot send data back to the NetBuffer.

The selection of "Input Only" and "Input & Output" also has a bearing on encrypted delivery via TCP2. Please see the section on Encryption and Authentication.

The Engineer's "Live Access" is unaffected by this option – it is always bi-directional, or input & output.

## Data

Selecting "ASCII" will convert all incoming data (whether from Serial or TCP1) to 7-bit ASCII only data. This is particularly useful since some data sources output serial data in '7-data bit plus mark parity'. Such data, if viewed in 8-bit mode, looks like garbage until the top bit is zeroed. (Programmatically, the stored data is 'AND'd with 0x7f as it is stored in the flash memory.)

The web status page shows the current data mode.

When "Binary" is selected, the data is stored as received, in full 8-bit mode.

**Warning** If the data source is binary, or contains ASCII data in the range 0x80-0xff that needs to be preserved, you must select "Binary" for this option.

## Memory

Selecting "Forced Linear" will force the NetBuffer to lose new data when the memory is full, if the handshaking or flow control is not observed. The default is "Circular" which selects the automatic circular/linear memory mode.

For more information see pages 6 and 33.

## TCP1 IP+port

If the mode is "TCP1 -> something", enter an IP and port number. If the NetBuffer needs to connect to a device, enter the device name or IP address (e.g. "main.pbx.company.com" if using DNS, or "mainpbx" if using NetBIOS, or "192.168.0.192" for a direct IP).

On the other hand, if the NetBuffer should sit and wait for another device to connect to it, leave the IP field blank. If you want only one IP to be able to connect, prefix the entry with the hash "#" character. You may enter a dotted IP, a NetBIOS name, or DNS name after the hash. The NetBuffer will sit and wait for that machine to connect to it. Any other IP will be refused a connection into the NetBuffer.

The port needs to be filled in. Typically use a port above 1024. This port has to agree with the port of the far device, otherwise no connection will be established.

Once connected, the NetBuffer will execute the "Login" sequence if set, or just start collecting data.

## TCP2 IP+port

If the mode is "something -> TCP2", enter a name or IP and a port.

The same rules apply as for TCP1.

Once connected, the NetBuffer will output any stored data to the far device.

## Live UDP output

If you need a non-guaranteed local delivery of data – perhaps for testing purposes – enter a name/IP and port.

When data arrives from the source port, and is stored, the packets of data will also be sent across the LAN to the device(s) specified by this field. If all local machines should see the data, enter "255.255.255.255" as an address – a broadcast. Again, you may enter a fully qualified domain name, a local NetBIOS name, or a dotted IP address.

A port should be chosen that does not conflict with other UDP software or protocols.

To see the packets, the PC(s) need to have custom software running to pick up the data and present it.

If a name is entered that requires DNS resolution, the NetBuffer will check the name-to-IP mapping every 10 minutes.

## RX/TX

This selection controls the serial port pin-outs. The default is "Auto" where the NetBuffer will look for a negative signal (less than -1V) to decide which is the receive pin. If this is not appropriate (e.g. there is no active receive pin, or there are two receive pins), choose one of the other options to force the pin-out.

If a forced pin out is chosen, and the NetBuffer cannot detect a negative input on the selected receive pin the status will show "2?" or "3?"

## Baud rate

Select the required baud rate from between 300 and 115200.

**Note:** The NetBuffer will run at quarter speed when using the 300 and 600 baud settings.

Under normal conditions, with Autobaud Enabled, you do not need to set this, however, by manually entering the correct speed and protocol for a new installation the NetBuffer will not lose data when autobauding for the first time.

If you set a baud rate that does not match the incoming data rate, the NetBuffer will automatically autobaud (assuming Autobaud is enabled).

## Protocol

Select the required combination of data length and parity setting.

## Autobaud

Use this setting to enable or disable autobauding. Under normal circumstances, ensure that autobauding is **enabled**.



## Setup: Login TCP1...

<http://.../login>

The login section is designed specifically for modes that source data from TCP1. It is used when the device you need to collect

data from has a simple authorisation sequence before it gives you data. For instance, it may require a username and password.

In this web page you can program up to four "match strings", and four "send strings". Each match string may be up to 16 characters long, while the send strings can be up to 63 characters.

When the TCP/IP socket first connects, the NetBuffer waits to look for "Match 1". When it sees that string it sends "Send 1". Then it waits for "Match 2", then sends "Send 2", all the way up to 4. (Blank strings are skipped.)

There are special characters for use in the match/send string sequence. The "#" (hash) character in any send string will be converted to a carriage/return (0x0d, 0x0a) sequence. The "\$" (dollar) character in either a match or send string will be interpreted as the null character (0x00).

Additionally, the "/" (forward slash) can specify hexadecimal values. E.g. "/20/01/FF". If a sequence of hex is needed, enclose them within curly braces "{" and "}", e.g. "{2001FF}". If you need to use one of the special characters literally, prefix with the "/", namely: "/{", "/}", "/", "/#", "/\$".

**Note:** The equals sign and the single and double quotes must be quoted as hex. e.g. "=" must be represented as /3D, and " as /22, and ' as /27

If the TCP1 is connecting to a telnet server, then enter "/~" as the first characters of Match1. The NetBuffer will negotiate any telnet options the server requires. When using this option you *must* provide a match string as well. E.g. "/~login:" – will negotiate telnet options if present and then look for the string "login".

When the sequencing gets past 4 the connection is considered "done", and data logging begins.

If the TCP/IP socket closes, or is reset, the whole match/send process is repeated when the socket is reconnected.

As an example, you may want to put "user" in Match 1, with "guest#" in Send 1 (this adds a carriage return/linefeed combination as if you pressed the Return key), and "password" in Match 2, with "letmein#" in Send 2.

	Match	Send	
1	<input type="text"/>	-> <input type="text"/>	/~=Telnet (Match 1 only)
2	<input type="text"/>	-> <input type="text"/>	#=CR/LF
3	<input type="text"/>	-> <input type="text"/>	\$=0x00
4	<input type="text"/>	-> <input type="text"/>	/nn=hex or {hex}

(see the manual)

Save Cancel

Version 2002-Apr-18 2.00, built 2002-04-19 10:10

Obviously, the entries you put in here depend on the type of TCP/IP datasource you are trying to connect to.

The status of the login sequence is shown on the Status web page, under the section "Login" in the "Source TCP1" area. An "M" prefix indicates the NetBuffer is waiting for a string match, while "S" indicates it is trying to send a string. The numerical suffix shows which step the sequencing is at. If the sequencing is complete, "Okay" will be shown (the JavaScript variables "lmi" and "lsi" will be at 255 in this state).

All information sent by the remote TCP device or server will be stored in the memory of the NetBuffer.

## Rlogin servers

If the remote TCP device is expecting the RLOGIN protocol, you need to put the following information in the Send1 line:

```
$username1$username2$vt100/9600$
```

The "username1" should be the client side username (put in the NetBuffer's device name if you are unsure). "Username2" should be the user name the server (the remote TCP device) will authenticate against. The string "vt100/9600" informs the server of the terminal type and line speed.

The rlogin server will send a null (remember it will be converted to a "\$" if you need to match against it). If the rlogin server needs a password, it will send a string, typically "Password:", or something similar. Place this in Match2, and put the password in Send2. e.g. "letmein#"

## Setup: FTP

http://.../ftp

The FTP page allows the configuration of the NetBuffer's FTP server.

### Port

[Status/Index] [Email Now!] [Lock] [Engineer Menu] [Setup Menu] "NB000401" 00-04-01

**FTP**

Port  *0=FTP disabled*

Username  *leave blank for anonymous FTP*

Password  *leave blank for no password*

Filename  *as seen by FTP client*

Version 2002-Apr-18 2.00, built 2002-04-19 10:10

Enter the port the NetBuffer should "listen" on. Typically this is port 21 for standard FTP. You may want to 'hide' the FTP on another port (e.g. 43210). Or you can disable the FTP server altogether by entering 0.

### Username

The FTP server only allows for one username/password combination. Enter a username here, or leave it blank to accept any user. The default is "user".

### Password

Enter the password for the user. The default is "password". Leave this field blank to allow any password.

### Filename

The NetBuffer presents one file to the FTP client. The default is "download.dat".

**Note** Data collection does **not** stop when an FTP client is logged into the NetBuffer. Every time the client requests a directory, with the "LIST" command, the current file size is 'frozen' and presented to the client. When deleting the file, the "DELE" command will erase only this 'frozen' data portion.

As a result any FTP client does not have to worry about loss of data, it merely needs to login, do a directory (optional), retrieve the file, and optionally delete the file.

**Warning** When writing FTP clients, perform requests in this order: LIST, RETR, DELE. The LIST is optional, however, do **not** request a LIST between the RETR and the DELE – this will obtain new 'frozen' pointers and therefore delete data that has not yet been downloaded.

## Setup: Email

<http://.../email>

The NetBuffer can email the data, or just notification messages to one email destination. In order to send the email there must be an SMTP server that is accessible by the NetBuffer (either a local server, or one on the Internet that can be accessed through a router).

If the data transfer mode is "something ->

email" the data is attached. Any other data transfer mode will merely send a notification email.

All emails arrive with an informative subject line that contains a sequentially unique number, the NetBuffer device name, and a list of reasons for the email (note that there may be only one reason, or several – this is important if you are developing custom email processing software). The email also contains a simple HTML message body that gives some simple diagnostic information, plus an attachment of an HTML status document – which is actually the 'frozen' status page.

Any data sent will be attached as a base-64 encoded MIME attachment, and will use the filename set in the "FTP" web-page. You have the choice of appending the unique email ID to the filename.

If there are problems with sending the email, the last state and error code are displayed in the main "Status" page. The error codes given are the actual response codes from the SMTP server, with the exception of "529" which indicates the NetBuffer could not even gain a TCP/IP conversation with the server defined.

When an email needs to be sent, the NetBuffer will try connecting for 5 minutes. If unsuccessful, it will hold off for 30 minutes before repeating the attempt.

**Note:** If the SMTP server address is left blank, the SNMP traps are still generated for the values specified on this setup page.

[Status/Index] [Email Now!] [Lock] [Engineer Menu] [Setup Menu] "NB000401" 00-04-01

### email

*Connection*

SMTP  *Name or IP of server*

Port  *default=25*

Username  *blank=no authorisation*

Password

Domain  *Login domain. eg. "scannex.com"*

email to  *Full address of recipient*

*Data Attachment (Serial->email or TCP1->email)*

Limit  *kB. 0=no limit*

Filename  *(same as FTP)*

Suffix  .8-digit-ID  (none) *Suffix unique ID*

*Triggers*

Every  *minutes. 0=not timed*

... if  Have Data  Always *(Applies to "Every"). Default=Always*

When  *% full. 0=not used, 1=any data*

Quiet  *minutes. Detects a quiet data source. 0=not used*

Failures  Notify  Ignore *Source failures and powerup*

Version 2002-Jun-21 2.01, built 2002-06-20 16:37

## SMTP

Enter a name or IP of the SMTP server. If you are using a local SMTP server, you can also enter a NetBIOS machine name.

For some proxy servers you need to put the address of the *proxy* machine, and not the actual SMTP server. For others, including routers, enter the actual address of the SMTP server.

If the NetBuffer was configured using DHCP, the SMTP server address can be passed with the DHCP server reply. You may need to setup the DHCP server to do this.

## Port

Enter a port address for the SMTP server. The default is port 25. However, some ISPs perform transparent proxy of SMTP port 25 traffic. Consequently, this value can be used to "piggy back" across the internet and email directly into your SMTP server.

Obviously, the SMTP server should be set to listen on this non-standard port. Alternatively it may be possible to set the SMTP server's incoming firewall to port-forward from the non-standard port to port 25 - allowing a mix and match of standard and non-standard ports.

## Domain

This is the fully qualified domain that will be used for the SMTP server communication. When software talks to an SMTP server it says "HELO domainname". Many SMTP servers ignore the domain name at this stage.

However, the NetBuffer also uses the domain name for the "from" part of the email. It tells the SMTP server that the email is from "devicename@domainname". The *devicename* is set in the "Network" web-page of the NetBuffer. (Be sure not to use the "@" symbol in the devicename, or other illegal email address characters.)

e.g. "MainPBX@mycompany.com".

Many SMTP servers will validate the domain (at least) and disallow email from bogus domains. An error code 501 usually results from an incorrect domain name.

## Username

If the SMTP server requires authorisation to send emails, enter a username here. If blank, the NetBuffer connects with the "HELO" command, otherwise it connects with the "EHLO" command and issues an "AUTH LOGIN" sequence.

Typically, SMTP servers will allow emails to be sent to the same domain without requiring authorisation. However, if the email is on another domain and the SMTP server is public, authorisation is normally required. This mechanism is designed to prevent spam senders hijacking an SMTP server for their own needs.

On the status page, a last state of "Auth" and a code of 535 indicates an authorisation failure, i.e. wrong username or password. A last state of "Rcpt to" and a code of 550 indicates that authorisation is required – fill in the appropriate username and password.

The maximum number of characters is 63.

## Password

If you set a username for SMTP server authorisation, enter the password here.

The maximum number of characters is 63.

## email to

Normally you should enter a full email address here, complete with domain name.

If it is a local email, just the recipient (with no domain) may suffice.

## Limit

Some mail servers have a limited email capacity. Use this field to indicate a maximum size of source data, in kilo-bytes. The actual email size will be approximately 1.5 times this maximum size (allowing for the MIME encoding overhead).

If your SMTP server is limited to one mega-byte, enter a maximum value of 600 – which limits the email output to about 900kb.

When an email is triggered, a series of emails are generated. Each fragmented email has in the subject line the text “, **Fragment**”. The last email in the set will not contain this flag.

Enter a value of zero to indicate no limit to the email.

## Filename

Enter the filename for the data attachment (if in the ->email mode). This is the same setting that appears on the FTP page.

## Suffix

If this is “.8-digit-ID”, then any attached data will be suffixed with the 8-digit hexadecimal email identifier. The identifier is incremented every time the NetBuffer succeeds in delivering an email (either with or without data attachments).

## Every

If you want notifications and/or data delivery on a timed basis, enter the number of minutes here. 1440 is every 24 hours, while 720 is every 12 hours. The timing starts from the point the configuration is submitted, since the NetBuffer has no real time clock facility.

The subject line will contain “, **Time**” for an email triggered by this event.

Enter “0” to disable this feature.

## ...if

If this option is “Have Data” the timed “Every” trigger will only send an email if there is at least some data in the NetBuffer. This is useful where data needs to be delivered at regular intervals during the week, but not at weekends (because there is no data output).

The default is “Always”.

## When

To deliver data when the NetBuffer becomes full, or to notify when that happens, enter a percentage.

The subject line will contain ", **Full**" for an email triggered by this event.

If the mode is "Serial->email" or "TCP1->email", and the value is "1" an email will occur within 8 seconds of *any* data being received. This option allows the "Every" to be used as an infrequent confirmation-of-operation email (say every 24 hours), while the "When=1" will make the NetBuffer email as soon as data is received. When using this option, set the "Data pause" option in the Mode page to prevent slicing of the data. (See the section on Suggested Configurations for examples of use.)

When email is used for notification only, emails are sent out every hour while the memory is above the set limit.

Enter "0" to disable this feature.

## Quiet

You may have noticed the status page contains the time since data was last received. This allows a "quiet" notification/delivery – where an email is generated if no data was received from the source within n minutes.

Once this is triggered, the email will be triggered periodically at the interval specified by the entry.

As soon as data is resumed, another email is sent to that effect.

The subject line will contain ", **Quiet**" or ", **Data**" to indicate this.

Enter "0" to disable this feature.

## Failures

If this is "Notify", an email will be triggered when the data source is connected, when the data source is disconnected, and also when the NetBuffer is rebooted.

The subject line will contain ", **No Source**" or ", **Source ok**".

The subject line will contain ", **Init**" when the NetBuffer is rebooted, or when the operation mode is changed via the web-interface. You can infer which by examining the "time alive" value in the attached status page of the email.

## [Status/Index] menu item

<http://.../index>

<http://.../default>

The status page gives a quick view of the state of the NetBuffer. It is presented in human-readable form. The page is rendered using JavaScript showing the following sections:

- Top menu items, NetBuffer name and serial number
- "Source" status, including 'time since data'
- 'Up-time' (i.e. the time since the NetBuffer rebooted or powered up)
- Memory status: number of bytes used / number of bytes total
- Memory flags
- "Destination" status, including the optional "Encrypted" or "Authenticated" labels
- FTP status, if enabled and FTP is not the only destination
- Email status, if configured and email is not the only destination
- Firmware version and manufacture date.

[Status/Index] [Email Now!] [Lock] [Engineer Menu] [Setup Menu] "NB000401" 00-04-01

### Status

**Source** Serial Port (Input & Output, Binary)

Baud	Protocol	DSR	CTS	RX pin	Autobaud
19200	7D	1	1	0	0

0-00:09:56 since data

---

**Up time** 0-00:09:56  
**Bytes** 0 / 4186112 (0%)  
**Status** Circular

---

**Destination** email

Active	Last State	Code	Next ID
0	-	0	00000000

Encrypted

FTP

Active	File size
0	0

Version 2002-Apr-18 2.00, built 2002-04-19 10:10

For automated processing, a selection of variables are embedded in the raw data sent back to the PC from the NetBuffer. These variables are easily processed with software. The list of variables is described in detail later in this manual.

The status page is always available to any machine that can access the NetBuffer.

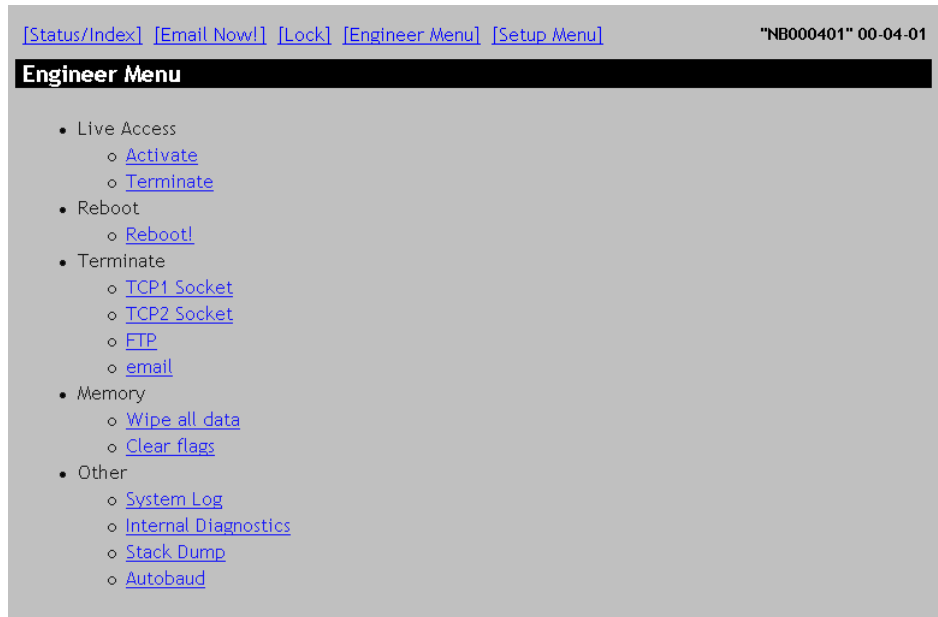
All other pages require authentication, and need a username and password. The default user is "admin" and "secret" for the password. Additionally, if the "WebLock" feature has been enabled every sensitive page is protected by a challenge/response mechanism.

**Note:** When TCP sockets are connected and encrypted or authenticated, the "Connected" box will show a suffix to indicate that encryption or authentication is active on that TCP socket. "\*" = two-way encryption; "\*i" = inbound only encryption (outbound data not allowed); "\*o" = outbound only encryption (inbound data not allowed); "!!E!" = error in configuration or a secret mismatch between the two devices.



## [Engineer Menu] menu item

<http://.../em>



## Live Access: Activate

<http://.../live>

Accessing this web link will trigger an engineers setup mode. To make use of it you need a TCP/IP client on PC ("Telnet.exe" is a reasonable example, though it does tend to "cook" the incoming data by line wrapping and stripping some unprintable characters, "SETelnet" is another example).

Access this url once, then from the *same* PC, connect to **port 87** of the NetBuffer. The NetBuffer will wait for a maximum of two minutes for that connection. Once established the link will remain open, but will timeout if nothing is received from the PC for five minutes.

For example, type "Telnet *ipaddress* 87" from the Start/Run line of Windows.

When connected, the engineer can then type and see live responses on the PC. Note that everything that is seen over this "live" link is also stored. However, if the destination is TCP2 or Serial, then the NetBuffer will block data received from the destination port while the live access is in use.

During the live link connection period, "Live access: In use" will appear on the NetBuffer's status web-page.

## Live Access: Terminate

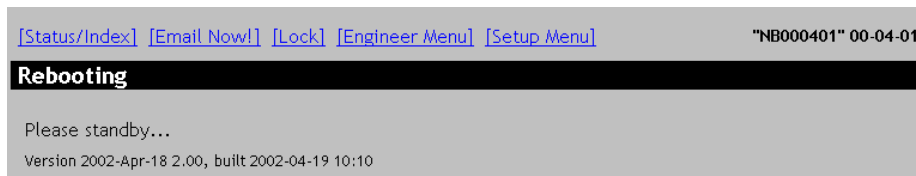
**`http://.../k1`**

Accessing this web link will kill the live access socket – whether the socket is awaiting a connection, or established.

## Reboot

**`http://.../_reboot`**

Use this link to reboot the NetBuffer. The following page will be seen during reboot, which should reload to the main status page when the reboot is complete:



## Terminate: TCP1, TCP2, FTP, email

**`http://.../k1`**

**`http://.../k2`**

**`http://.../kf`**

**`http://.../ke`**

These links allow the brutal termination of any TCP/IP connections to TCP1, TCP2, FTP, SMTP.

Use these with caution! They are included since sometimes PCs can “orphan” a socket, leaving the socket connected to the NetBuffer, and keeping it active even though the application may have been closed on the PC!

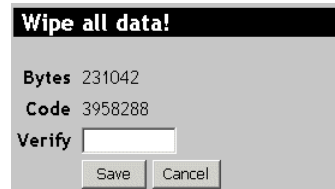
It is worth mentioning that most TCP/IP connections in the NetBuffer are time-limited to two minutes. Those that need to be kept open, such as TCP1 and TCP2, will send a “keep-alive” TCP/IP packet every minute of inactivity to keep the link alive. Consequently, any network disruptions can only really have one or two minutes of down-time before the NetBuffer closes the socket and begins to try again.

## Memory: Wipe all data (V2.11+)

**`http://.../wipe`**

This link wipes all the data in the NetBuffer. Clicking this link takes you to a page that shows how many bytes will be wiped. In addition, a ‘code’ is presented that must be verified. Just copy the code into the input box and click ‘Save’.

This operation is irreversible!



## Memory: Clear flags (V2.11+)

<http://.../cmf>

In version 2.11 the memory flags that show "Overfull", and "Wrapped" are persistent. If the data is being delivered the flags are cleared on completion of the email, or chain of email fragments. (Versions prior to 2.11 automatically cleared the flags when memory was freed.)

For all other delivery methods, the flags must be manually reset with this url.

## Other: System Log

<http://.../diag>

The System Log page contains some useful information in diagnosing fault or power cut scenarios:

### Up Time

The number of days-hours:minutes:seconds the NetBuffer has been alive. This timer is reset whenever the NetBuffer reboots.

### Power

Gives an indication of the power supply state, in millivolts.

### Reboots

The number of times the NetBuffer has been rebooted correctly. Each power up, or "Reboot" request (via the web), will increment this number.

### Brown Outs

The number of power brown outs. A brown out is where the power supply dips to a dangerous level and the NetBuffer assumes it is a power fail. However, the supply has resumed before the NetBuffer died completely, and so a reboot is performed.

If brown outs occur frequently, it may indicate a problem with the plug-top charger, or the mains supply.

### Traps

A trap is a firmware failure. You should never see any!

### Events

This section displays the last sixteen events, in reverse chronological order (i.e. the last event that occurred is at the top of the list).

### STCs

Self Test Code results.

## Other: Internal Diagnostics

<http://.../di>

This information is for diagnostics purposes only. Some of the sections can be helpful in sorting out connection and setup problems.

The section "DNS" shows the name, the resolved IP address, and the Ethernet MAC address (for SNMP) of the device.

The section "DHCP" will show the server's IP and ID, as well as the time outs T1 and T2, as defined in the RFC-2131.

## Other: Stack Dump

<http://.../trace>

Reserved for trap diagnostics.

## Other: Autobaud

<http://.../dabn>

Accessing this url will begin autobauding on the serial port, even if there have been no communication errors.

**Note:** Autobauding will be initiated even if the autobauding feature has been disabled on the mode page.

## **[Lock] menu item**

`http://.../ds`

Accessing this url will close down the five minute open security window of the NetBuffer. Any requests to protected pages will then require authorisation.

During the "open security" window, all web pages are returned with a **pale green** background. After that time, when 'locked', the web pages have a **grey** background.

In addition, if the "WebLock" feature is enabled, on the encryption page, this url will stop any further access to the web configuration pages. Further access will require a response code to the NetBuffer's challenge.

## WAP status

(Not shown on the menu)

**http://.../wap**

The WAP page is suitable for ultra-thin browsers, such as WAP mobile phones. It provides a very simple status display of the NetBuffer and contains a link to trigger a user-initiated email.

The name of the NetBuffer is shown as the title page, with an [Email Now!] link further down the page:

*(Image from "Deck-It WML Previewer" by PyWeb.com, with a Nokia Phone image. Trademarks and Copyrights acknowledged.)*



## Suggested Configurations

### Collect RS232 data and deliver to a specific PC or NetBuffer

Set the operating mode to "Serial->TCP2", and program in the IP address (or name) of the PC and the required port in "TCP2".

The NetBuffer will attempt to open a socket to that PC every 4 seconds in order to deliver the data. The PC or NetBuffer must have a "Listen" socket open to obtain the data. As soon as a link is established, the data will be streamed from the NetBuffer's memory to the remote end.

When connected, the remote end can send back data that will be output to the RS232 port.

It is suggested you site the NetBuffer immediately next to the device you are logging. If the network cabling is then damaged, or the intermediate hubs are powered down, the NetBuffer will continue logging.

### Collect RS232 data and wait for collection via TCP/IP

Set the operating mode to "Serial->TCP2" and program a port number in "TCP2". Leave the IP address of TCP2 blank.

The NetBuffer will wait for an incoming connection (from any IP address, or an IP as specified with the '#' prefix mentioned on page 17) on the specified port. As soon as connection is established the data can flow both ways.

### Create an RS232 link over IP with two NetBuffers

Just program each buffer for "Serial->TCP2". Program one NetBuffer's TCP2 settings to be a blank IP (or the other NetBuffer's name or IP address using the '#' prefix mentioned on page 17) and a port number (say 2020). Program the other NetBuffer's TCP2 settings to the IP, or device name, of the first and the same port number (say 2020).

Both NetBuffers will then communicate with each other and provide a transparent RS232 link. A web-browser may be used to interrogate either NetBuffer.

### Deliver alarm notifications by email

This configuration suggestion is for situations where the monitored device outputs data as a result of some event, such as an alarm. The data needs to be transferred immediately, and the system also needs regular emails to confirm connection.

Set the Mode to "Serial->email" or "TCP1->email". Also set the "Data pause" option to a reasonable figure. A value of 200 will wait 10 seconds before registering the data in the memory, and thus triggering the email. This setting prevents the situation where small pauses in the transmission cause the email to be spliced into two, or more, emails.

In the Email page, setup the usual SMTP server, domain, and email to options. Choose UseID = yes. This allows the receiving station to determine if any emails have been lost in transit, or sent out of sequence.

Set the "Every" to 1440 to email a confirmation every 24 hours, and set "...if" to **Always**. This will send a blank email every 24 hours. Now set the "When" to 1 – which will send data as soon as it is received.

Ensure that "Quiet" is set to 0, and "Failures" is set to "Notify" (so you will know if the serial plug gets pulled!).

## Deliver periodic data by email

This configuration is useful for devices that output a block of data at a periodic interval, say every hour. The data needs to be delivered quickly. The situation is similar to the above example, but further checks can be added.

Set the Mode to "Serial->email" or "TCP1->email". Also set the "Data pause" option to a reasonable figure. A value of 200 will wait 10 seconds before registering the data in the memory, and thus triggering the email. This setting prevents the situation where small pauses in the transmission cause the email to be spliced into two, or more, emails.

In the Email page, setup the usual SMTP server, domain, and email to options. Choose Suffix = .8-digit-ID. This allows the receiving station to determine if any emails have been lost in transit, or sent out of sequence.

Set the "Every" to 0 to disable timed emails. Now set the "When" to 1 – which will send data as soon as it is received.

Set "Quiet" to some figure above the periodic output of data. For instance, if the device outputs hourly, set this figure to 90. If the device stops outputting data, you will know that something has stopped. Set "Failures" is set to "Notify" so you will know if the serial plug gets pulled, or if the monitored device is turned off.

## Stacking NetBuffers for larger memory or bridging

By using the "Memory=Linear" on the operating mode page it is possible to create a larger memory. For example, to provide double the storage, first of all connect two NetBuffers to the same local hub. Then configure the first NetBuffer that is connected to the Data Source (either by serial or TCP connection) to deliver the data to TCP2. Enter the IP of the second NetBuffer in the TCP2 IP and enter a port (say 2020). Set the Memory mode to either "Linear" or "Circular" – see the text below.

Now, configure the second NetBuffer to collect from TCP1. Enter the IP for TCP1 as "#" plus the IP of the first NetBuffer, and enter the same port number entered in TCP2 of the first NetBuffer (see the notes on "Create an RS232 link over IP with two NetBuffers" above). Configure delivery as required. **Now set Memory = Linear.**

In this mode the first NetBuffer will normally pass the data straight onto the second. When the second is full, the first (nearest the data source) will fill up. If both become full the data source will be signalled to stop sending.

If the link between the two NetBuffer is severed, or the second NetBuffer goes off line, the first can be programmed to either keep the old data (Memory = Linear), or keep the newest data (Memory = Circular).

This mode is also useful in bridging a network, and placing one NetBuffer on each side of the 'bridge'.



## Automated Software Considerations

### Web status page

To allow for a custom written automated status monitor, the status page (url "index") returns the data as JavaScript/HTML. All the important information is contained in a variable block that can easily be processed in software. It appears immediately after the line:

```
<!--
```

and terminates on a blank line. Following the blank line is the JavaScript that turns these variables into meaningful and 'pretty' HTML for human consumption.

The web variables are listed later in this document.

### Configuration by custom software

The configuration forms (Network, Mode, Login, email, and FTP) post their data back to the NetBuffer. Again, automated software can be used to process the required variables (in the same style as the status page), and to post selected pieces of change back to the NetBuffer.

The POST command should be issued as:

```
POST /_ca
```

The authentication entry must be there, and valid, if the 5-minute security window is closed. (See the Internet RFCs on formatting the Basic Authentication entry of an HTTP request.)

Following that is a blank line and the parameters in the form: p=d&...

Where p is the parameter id, d is the data, and the "&" is used to separate the fields. Finally a CR/LF sequence. Note that the HTTP encoding applies – spaces are encoded as "+", while other special characters are described as "%hh", where hh is a two digit hexadecimal value of the character.

Custom software can then open a socket to port 80 on the required NetBuffer, and transmit the request. The NetBuffer will then send back an "Updated!" or "Not updated" web page in HTML before closing the socket.

## email reference

The NetBuffer sends a number of items in the subject line with each email. Be aware that multiple flags can occur in one email.

<b>Search text</b>	<b>Indicates</b>	<b>Setup Requirements</b>
, Init	NetBuffer has just rebooted, or has had its mode changed.	"Failures" is Notify.
, Full	Full trigger.	"When" is non-zero
, Time	Time trigger.	"Every" is non-zero
, No Source	Data source has disconnected.	"Failures" is Notify.
, Source ok	Data source has been reconnected.	"Failures" is Notify.
, Quiet	Data source has been quiet for an extended time.	"Quiet" is non-zero.
, Data	Data source has resumed transmission.	"Failures" is Notify.
, Fragment	There are more emails following in this set	"Limit to" is non-zero and the mode is "...>email."
, User ( <b>ip</b> )	User requested an email via the web. The IP is shown in dotted form after the text.  e.g. "..., User (192.168.0.11)".  Firmware pre 1.16 just shows ", User"	None.

There will always be an attached html file that is a snapshot of the status page. This contains important JavaScript variables. The file is named in the form:

"DeviceName.status.html"

If the data mode is set to deliver data by email, the attached file will be in standard MIME base-64 encoding, and have the filename set in the "FTP" web page of the NetBuffer. In addition, it may have a suffix of the email ID (as set in the email web-page "Use ID").

## NetBuffer LEDs

There are four LEDs on the NetBuffer: Net Activity, Net Link, Data, and Status.

Net Link will be lit when there is a valid 10base-T connection, while Net Activity will flash whenever network packets are received by the device. If the Ethernet link is disconnected, the Net Link LED will flash.

Data and Status are used to show the following conditions:

- 1) Primary link down (ie the port the NetBuffer is storing data for). The Status LED will be mainly on (3 seconds on, 1 second off)
- 2) Secondary link down (ie the port the NetBuffer is sending to). The Status LED will be mainly off (1 seconds on, 3 seconds off)
- 3) Both links down – Status on all the time.

**Note:** The RS232 is considered “dead” if either, or both, handshakes go inactive for more than 20 seconds. If the cable is unplugged it is “dead” almost immediately. It takes a period of 5 seconds with the line connected and both lines active to consider it “alive”. If the mode is set to “Source = Input Only” only the receive pin is checked for negative voltage, and the handshake lines are ignored.

Data will blink two or three times as data is stored, and remain on while there is at least one byte in the memory.

In addition, the following conditions are represented:

<b>Condition</b>	<b>LEDs</b>
Waiting for DHCP/BOOTP reply	2 blinks on Data, 2 blinks on Status
Hardware Failure 1	4 blinks on Data, 4 blinks on Status
Hardware Failure 2	6 blinks on Data, 6 blinks on Status

## SNMP Trap reference

The SNMP traps are sent to UDP port 162. By default the NetBuffer performs a broadcast to all listening machines. However, you can configure (under the "Network" web-page) a specific IP address to send SNMP traps to.

You can view SNMP traps from NetBuffers on a Windows® machines using the NBTrapView software. All traps are sent as TrapType 6 (Enterprise Specific).

<b>OID (Object id)</b>	<b>Details</b>
1.3.6.1.4.1.6024	Scannex Electronics Ltd top level OID
1.3.6.1.4.1.6024.1.2	NetBuffer

<b>SpecificTrap</b>	<b>Code</b>	<b>Details</b>
6	0x06	Booting
7	0x07	Reboot
8	0x08	Powerfail
9	0x09	IP_Assign
15	0x0f	HeartBeat
16	0x10	Weblock
17	0x11	Secret
18	0x11	Auth_Config
29	0x1d	FTP_Password
32	0x20	Email_success
33	0x21	Emai_fail
40	0x28	Autobaud_start
41	0x29	Autobaud_end
48	0x30	Full
49	0x31	Not_Full
50	0x32	Quiet
51	0x33	Not_Quiet
64	0x40	TCP_Source
65	0x41	TCP_Dest
78	0x4e	TCP_SMTP
80	0x50	C_Source
81	0x51	C_Dest
93	0x5d	C_FTP
94	0x5e	C_SMTP
95	0x5f	C_Live
96	0x60	D_Source
97	0x61	D_Dest
109	0x6d	D_FTP
111	0x6f	D_Live

Along with each trap is delivered the "interesting variables":

<b><i>OID (Object Identifier,</i></b>	<b><i>Type</i></b>	<b><i>Details</i></b>
1.3.6.1.4.1.6024.1.2.1	Serial Number	6 byte string containing the Ethernet MAC/Serial No.
1.3.6.1.4.1.6024.1.2.2	Box name	Up to 16 character text: "NB123456"
1.3.6.1.4.1.6024.1.2.3	Connection Status	A bit field representing the connection states (1 indicates connected): D0 = Source Port D1 = Destination Port D2 = Live Port D3 = FTP

**Note:** Unlike version 1.xx of the firmware, all trap types can be sent to a local broadcast address, or remote address (specified by name or dotted IP).

## JavaScript variable reference

<i>Variable</i>	<i>Notes</i>	<i>Admin Read</i>	<i>Admin Post</i>	<i>Status Read</i>
Actual far TCP1 Address	IP address	-	-	ai1
Actual far TCP2 Address	IP address	-	-	ai2
Actual far TCP1 Port	0-65535	-	-	ar1
Actual far TCP2 Port	0-65535	-	-	ar2
Encryption Enabled	string: "", "Authenticated", "Encrypted"	-	-	cen
Time since last data	d-hh:mm:ss	-	-	di
Time to delivery	d-hh:mm:ss	-	-	dt
SMTP active	0 or 1	-	-	ea
SMTP bytes sent		-	-	eb
SMTP ID		-	-	eid
SMTP last error	RFC result code	-	-	en
TCP1 encryption	Blank = plain text " *" = encrypted/authenticated " *i" = inbound only " *o" = outbound only " <b>!E!</b>" = error	-	-	es1
TCP2 encryption	Blank = plain text " *" = encrypted/authenticated " *i" = inbound only " *o" = outbound only " <b>!E!</b>" = error	-	-	es2
SMTP last stage	text string	-	-	et
FTP user logged in	0 or 1	-	-	fu
Live maintenance channel active	0 or 1	-	-	la
Login Match Index		-	-	lmi
Login Send Index		-	-	lsi
(Menu colour)		-	-	mc
Manufacture Date		-	-	md
(Menu email)	1 to show the [Email Now] link	-	-	me
Frozen bytes		-	-	mf
(Menu locked)	1 to show the [Unlock] link	-	-	ml
Memory status	Characters: L = Linear C = Circular F = Full W = Wrapped (lost old) O = Overfull (lost new)	-	-	mm
Total bytes		-	-	mt
Used bytes		-	-	mu
Serial Autobauding	0 or 1	-	-	ra
Serial CTS line	1 = asserted	-	-	rc
Serial DSR line	1 = asserted	-	-	rd
Serial Receive pin	2/3/0/?	-	-	rn
TCP1 active	0 or 1	-	-	sc1

<b>Variable</b>	<b>Notes</b>	<b>Admin Read</b>	<b>Admin Post</b>	<b>Status Read</b>
TCP2 active	0 or 1	-	-	sc2
Serial Number		-	-	sn
Software Version		-	-	sv
Time alive		-	-	ta
Autobaud enable	0 = disabled 1 = enabled	cab	ab	-
Serial Baud Rate	300-115200	cb	b	rb
Bidirectional	0 = input only from data source 1 = input/output data source	cbd	bd	bd
Use DHCP	0 or 1	cd	d	-
DNS 1	IP address	cd1	d1	-
DNS 2	IP address	cd2	d2	-
Deliver if source fails	0 or 1	cdf	df	cdf
Data pause	50ms chunks. 0=not used. Max= 50 x 50ms = 2.5 seconds	cdp	dp	-
Deliver if quiet for n minutes	6 digits max	cdq	dq	cdq
Delivery every n minutes	6 digits max (694days max)	cdt	dt	cdt
...if data	0 = email deliver always 1 = email deliver if have data	cdd	dd	-
Delivery when > x% full	0-100	cdw	dw	cdw
SMTP domain	63 char max	ced	ed	-
email to	63 char max	cee	ee	-
Append ID to email filename	0 or 1. If "1" then the email ID (8 digit hex) is appended to the filename. eg. "download.dat" becomes "download.dat.1234ABCD"	cei	ei	-
Encryption Key	24-digit hex (write only), blank = no change	-	ek	-
Limit emails	8 chars max (in bytes)	cel	el	-
Encryption level	0 = disabled 1 = authenticate only 2 = authenticate & encrypt	cem	em	-
SMTP password	63 char max	cep	ep	-
SMTP port	1-65535	cer	er	cer
SMTP server	63 char name	ces	es	ces
SMTP username	63 char max	ceu	eu	-
WebLocked	0 = disabled 1 = enabled	cew	ew	ew
FTP Port	Port 0-65535, default=21	cf	f	-
Forced Linear	0 = circular, 1 = linear	cfl	fl	fl
Gateway	e.g. 192.168.0.1	cg	g	-
TCP1 Address	IP address	ci1	i1	pi1
TCP2 Address	IP address	ci2	i2	pi2
IP Address	e.g. 192.168.0.234	ci	ip	-
Subnet Mask	Default 255.255.255.0	ck	k	-
Live Address	IP address	cli	li	-

<b>Variable</b>	<b>Notes</b>	<b>Admin Read</b>	<b>Admin Post</b>	<b>Status Read</b>
Login Match 1	16 characters	clm1	lm1	-
Login Match 2	16 characters	clm2	lm2	-
Login Match 3	16 characters	clm3	lm3	-
Login Match 4	16 characters	clm4	lm4	-
Live Port	Port 0-65535	clp	lp	-
Login Send 1	63 characters	cls1	ls1	-
Login Send 2	63 characters	cls2	ls2	-
Login Send 3	63 characters	cls3	ls3	-
Login Send 4	63 characters	cls4	ls4	-
Operation Mode	SF/ST/TF/TS/TF/SE/TE S=serial,T=TCP,F=FTP, E=email	cm	m	dm
FTP Filename	Max 16 characters	cn	n	-
Serial Protocol	"8N" etc	cp	p	rp
TCP1 Port	Port 0-65535	cr1	r1	pr1
TCP2 Port	Port 0-65535	cr2	r2	pr2
SNMP Address	IP address	cs	s	-
Strip top bit	0 = full 8 bit = Binary 1 = only 7 bit = ASCII	cs7	s7	s7
Serial RX pin	0 = auto 2 = RX from pin 2 3 = RX from pin 3	csr	sr	-
FTP User name	Max 16 characters	cu	u	-
FTP Password	Max 16 characters	cw	w	-
BoxName		xn	x	xn



## Web address reference

### Basic urls

<b>URL</b>	<b>Description</b>
_REBOOT	Reboot the NetBuffer
CMF	Clear memory flags (2.11+)
DABN	Autobaud Now
DI	Internal diagnostics
DIAG	Diagnostics index
DK	"Kill socket" main page
DS	Secure immediately
EM	Engineer menu
EMAIL	Setup the SMTP/email delivery options
EMAILNOW	Deliver email notification (and data if the destination is set to email)
EMAILWAP	Email now link for WAP
FTP	Setup the FTP parameters
INDEX or DEFAULT	Main index
K1	Kill socket 1
K2	Kill socket 2
KE	Kill email connection (abort)
KF	Kill ftp connection (abort)
KL	Kill live socket
LIVE	Enable live access
LOGIN	Setup the login sequence for TCP1
MODE	Setup the basic mode of operation
NETWORK	Setup the network configuration
PK	Set private secret/key
PWD	Set the admin web password
RTS= <i>nnn</i>	Unassert the RTS line for <i>nnn</i> multiples of 50ms, from 1 to 255. e.g. "RTS5" will unassert the line for 250ms, then reassert.
SUM	Setup menu
TRACE	Show stack trace for last trap
UL	Unlock
WAP	View simple status information.
WIPE	Wipe all data request page (2.11+)

### POST command urls

<b>URL</b>	<b>Description</b>
_ca	Change administration data (variables as outlined above)
_cp	Change administration password pw=... un=...
_ul	Unlock command
_wipe	Wipe data request response. VC= should contain the verify code delivered by the WIPE url for this to be successful.

## Common SMTP error codes

These error codes are specified in the Internet RFCs (in STD10, and RFC-821).

<b>Cod</b>	<b>Description</b>
211	System status, or system help reply
214	Help message <i>Information on how to use the receiver or the meaning of a particular non-standard command; this reply is useful only to the human user</i>
220	<domain> Service ready
221	<domain> Service closing transmission channel
250	Requested mail action okay, completed
251	User not local; will forward to <forward-path>
354	Start mail input; end with <CRLF>.<CRLF>
421	<domain> Service not available, closing transmission channel [This may be a reply to any command if the service knows it must shut down]
450	Requested mail action not taken: mailbox unavailable <i>E.g., mailbox busy</i>
451	Requested action aborted: local error in processing
452	Requested action not taken: insufficient system storage
500	Syntax error, command unrecognised <i>This may include errors such as command line too long</i>
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command parameter not implemented
529	Indicates the NetBuffer could not even gain a TCP/IP connection with the server defined.
535	Authorisation failure
550	Requested action not taken: mailbox unavailable <i>E.g., mailbox not found, no access, requires authorisation to send email</i>
551	User not local; please try <forward-path>
552	Requested mail action aborted: exceeded storage allocation
553	Requested action not taken: mailbox name not allowed <i>E.g., mailbox syntax incorrect</i>
554	Transaction failed

Note that these codes, with the exception of "529" are generated by the SMTP server, and not the NetBuffer.

## Power supply

Specification	Part No GN03-0024 7V to 9V at 150mA. (Damage may occur above 15V) Part No GN02-0024 9V to 12V at 150mA. (Damage may occur above 15V)
Physical connector	5.5mm barrel 2.1mm hole 12mm barrel length Centre positive

## Glossary of Network Protocols

<b>Term</b>	<b>Description</b>
ARP	Address Resolution Protocol This allows a device to ask "Which Ethernet MAC address is looking after this particular IP address?"
BOOTP	Bootstrap Protocol An automatic means for setting up IP address and other information on power-up. Normally this is handled by the more up to date DHCP. BOOTP, like DHCP, uses UDP ports 67 and 68.
DHCP	Dynamic Host Configuration Protocol Allows the NetBuffer to ask "Which IP can I be?" The DHCP server(s) will give an IP address, gateway, subnet mask, and a whole host of other data when the NetBuffer requests. Normally this is performed at power-up. DHCP allows for dynamic IP address, where an address expires after a given time, and may be changed. The NetBuffer supports this. In addition, the NetBuffer asks the DHCP server to take responsibility for informing any connected DNS servers. DHCP uses UDP datagrams over ports 67 and 68.
DNS	Domain Name System Provides the means to ask "What is the IP address of this named machine or site?". DNS uses UDP with port 53.
FTP	File Transfer Protocol Allows a remote (client) machine to retrieve file details, and/or the file itself. FTP requires two TCP ports – the command and the data port. The command port provides for basic authentication (username+password), and a list of text commands. The NetBuffer normally uses TCP port 21 for the command port, although this can be changed. Port 20 is normally used for the transfer of the data, though this depends on the mode of the FTP client software.
HTTP	HyperText Transfer Protocol This is the mechanism for transferring web-pages, generally coded in HTML (HyperText Markup Language). On the NetBuffer this is always carried over TCP port 80.
ICMP	Internet Control Message Protocol The NetBuffer uses this for the "ping" operation. The "ping" is often used to find out if a device is still working as it is (almost) the lowest protocol mechanism.
IP	Internet Protocol The basis, and foundation, for UDP, TCP and others. See the Networking Overview.
NetBIOS	NetBIOS has many different functions. Local device naming is one of them, and the only one that the NetBuffer supports. NetBIOS names may be up to 16 characters long, are not case-sensitive, and are normally made from "A" to "Z", "0" to "9" and "_". Some other characters may be used.
SMTP	Simple Mail Transfer Protocol The normal mechanism for sending emails.

<b><i>Term</i></b>	<b><i>Description</i></b>
SNMP	<p>Simple Network Management Protocol</p> <p>There are SNMP queries and traps. The queries allow questions to be asked of machines, such as "How many data packets have you seen?", and "How many bytes have you transmitted?". The NetBuffer does not support such queries.</p> <p>SNMP traps, on the other hand, are supported by the NetBuffer, and they tell of important events such as powerfailures, link failures and the like. SNMP traps are transmitted from UDP port 161 to UDP port 162.</p>
TCP	<p>Transmission Control Protocol</p> <p>This is the foundation of reliable data transfer, using IP, between two devices. It is error checked, and includes timeouts for retransmission and failure.</p>
UDP	<p>User Datagram Protocol</p> <p>This is a non-guaranteed message that is encapsulated in IP. Many other protocols use UDP – such as DHCP, DNS, SNMP.</p>

## Approvals

### FCC Rules Part 15 - Computing Devices

Note: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation.

If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

CAUTION: Changes or modifications not expressly approved by the manufacturer could void the user's authority to operate the NetBuffer.

### Industry Canada Regulatory Compliance Information for Class B Equipment

This Class B digital apparatus complies with Canadian ICES-003.

AVIS: Cet appareil numérique respecte les limites de bruits radioélectriques applicables aux appareils numériques de classe B prescrites dans la norme sur le matériel brouilleur: "Appareils Numériques", NMB-003 édictée par l'Industrie Canada.

### Australia and New Zealand users

#### EMI statement

This device has been tested and found to comply with the limits for a Class B digital device, pursuant to the Australian/New Zealand standard AS/NZS 3548 set out by the Australian Communications Authority and Radio Spectrum Management Agency

### European Union (EU) Statement

This is to certify that the NetBuffer complies with the EU Directive 89/336/EEC and the amending directive 93/68/EEC, relating to Electromagnetic Compatibility, by application of CISPR 22/European Standard EN 55022 (Class B) requirements for Information Technology Equipment and EN55024.

### EMI/Safety Requirement

EN60950	
EN55022	Class B
EN55024	Class B
AS/NZS 3548	Class B
UL/CUL	UL60950/CSA C22.2 No.60950
FCC Part 15	Class B

## Frequently Asked Questions

### Is data lost when using FTP?

No. When logged in with an FTP client, the NetBuffer continues to log 'in the background'. The client will see a 'frozen' file size. That file size is refreshed to include current data whenever a "LIST" command is issued. So the correct sequence is to:

1. Log in
2. Do a LIST
3. RETRIEve the file
4. DELEte the file
5. (If you perform a LIST now, the client will see any data that arrived between steps 2 and 4)
6. Log out

### When should I disabled autobauding?

In general – never. We recommend leaving autobaud enabled. However, if the NetBuffer is being used to transmit only (and not to receive), then it is best to disable autobauding. In addition, in this mode be careful not to call the DABN link (Autobaud Now) since the NetBuffer will still enter the autobaud process, but will be unable to complete the operation with no incoming data.

### When should the serial pin out be forced?

Normally the serial pin can be left in "auto" mode. However, there are a few circumstances when it becomes necessary to force:

1. When the NetBuffer is transmitting only into a device that has no transmit itself.
2. When the NetBuffer is connected into a Y-lead, or breakout lead (in parallel with another collection device). In this state the NetBuffer sees receive levels on both pins 2 and 3 – an ambiguous situation.
3. When the device has TTL level outputs, and no negative level. In this situation the NetBuffer cannot automatically detect and needs to be forced. However, the early build NetBuffers will not be able to receive the data even when forced. Only parts GN03-0024 support receiving TTL level data (not GN02-0024).