Commander Manual

Commander is the smaller of North's building controllers. Commander contains North's interface technology, block-based programming language, and easy-to-use information services. Commander can work as a stand-alone controller, or alongside other North controllers and display systems to create a larger control or monitoring solution.

This document relates to Commander Version 2.0

Please read the *Commander Tutorial* alongside this document, available from *www.northbt.com*

Contents

What is Commander? 6
- Interface Technology 6
- Programmable Control 6
- Information Services 6
- Typical Use 7

Quick Start 8
- Mount Commander 8
- Insert Battery and Set Switches 8
- Connect to LAN 8
- Apply Power 8
- Setup Commander using North Engineering software 8

Hardware 9
- Dimensions 9
- Mounting 9
- Connectors 10
- LEDs 10
- Power 10
- LAN Port 10
- RS232 Ports 11
- Internal Battery 11
- Internal Switches 11
- Memory 12
- Interface Licences 12

Firmware 13
- Main Firmware 13
- Interface Driver Firmware 13

Functional Organisation 14
- Functions 14
- Settings 14
- Organisation of this Manual 14

Platform Information 15
- Switch Settings 15
- Battery Level 15
- Reset/Restart Information 15
- Firmware Versions 15
- Debug Recording 15

LAN Port Setup 16
- Default IP Addressing 16
- Static IP Addressing 16
- Dynamic DHCP Addressing 16
- DNS Server Address 16
- Time Server Address 16

Real-time Clock and Time Zone 17
- Coordinated Universal Time 17
- Local Time 17
- Daylight Saving Time 17

Interfacing Commander to other Systems 18
Started Interfaces 18
Installed Drivers 18
Interface Licences 18

Transferring Values 19
Reading the Value 19
Writing the Value 19
Transfer Tasks 19

Essential Data 20
Structure 20
Pages 20
Objects 20
Value Reading and Writing 21
Adjustability 21
Alarm Monitoring 21
Value Logging 21

Controlling using Time and Date 22
Calendar 22
Timers 22
Profilers 22

ObVerse Cause-and-Effect Strategy 23
Properties 23
Modules 23
Editing ObVerse 23

Alarm Basics 24
Format 24
Alarm Delivery 24
Filtering Delivery 24
Delivery to any one of a Group of Destinations 24
History of Alarms 24
Emailing Alarms 25
Alternative Alarm Destinations 25

Communicating with other North IP Devices 26
Key Security 26
Checking Communications 26

Security Server 27
User Information 27
Groups 27
Use by other Tasks 27

Web Server 29
Defining the Page Template 29
Enhancing the Home Page 30
Security Settings 30
Support Page 30
Configuration Page 30
PROGRAM Switch 30

Telnet 31
Establishing a Session 31
IP Configuration Service 31
Query Response Service 31
PROGRAM Switch 32

Updating Commander's Firmware 33
What is TFTP? 33
Locating the CDM Files 33
Installing or Updating a CDM 33
Updating the base Commander software 34

Object Specifications 35
Example Object Reference 35
Device Top-Level Objects 35
Configuration 37
Platform Information 38
Operating System 39
Last Restart 39
Software Versions 39
Software Version Detail 40
Recording Configuration 41
LAN Port Configuration 42
UTC & Daylight Savings Configuration 43
Interfaces Configuration 44
Installed Drivers 44
Interface Licences 44
Web Server Configuration 45
Web Server Template 45
Web Server Home Page 46
Web Server Security 47
Web Server Default Access Security 47
Essential Data Configuration 48
Essential Data Page Configuration 49
Essential Data Object Configuration 51
Essential Data Task Control 53
Time Control Configuration 54
ObVerse Processor Configuration 55
ObVerse Object Modules 56
Security Server Configuration 57
Data Transfer Configuration 58
Data Transfer Task 58
Alarm Delivery Configuration 59
Alarm History Configuration 60
Alarm Email Configuration 61
Alarm Email Destination Configuration 63
Email Address Setup 64
North IP Devices Configuration 65
North IP Device Configuration 66
Telnet Setup 67
Essential Values 68
Essential Values Page 68
Time Control 69
Calendar 70
Calendar Exception Date 71
Timer 72
Profiler 73
ObVerse 74
Security Server 75
Security Server Group 75
Security Server User 76
Data Transfer 77
Transfer 77
Alarm Delivery 78
Alarm Delivery Destination 78
Alarm History 80

Alarm Emailer 81
Alarm Emailer Destination 81

Appendix A - Ethernet/IP Protocols 82

Appendix B – Pre-loaded Drivers 84

Appendix C - PROGRAM switch 85

Warranty 86

EMC Conformance 86

End of Life Recycling 87

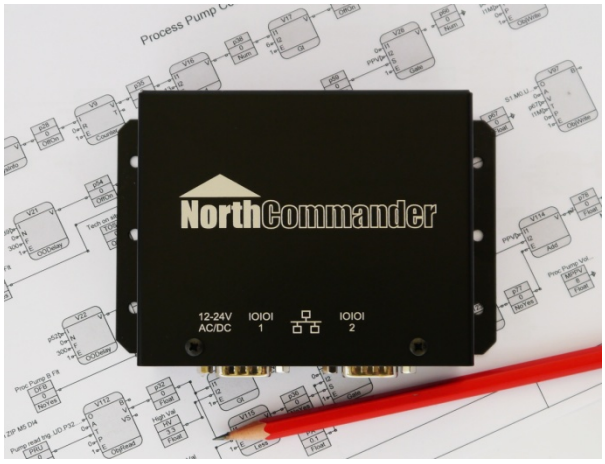
Commander Versions 88

What is Commander?

Commander is the smaller of North's building controllers, which also includes ObServer. The controllers contain North's interface technology, block-based programming language, and easy-to-use information services. Commander can work as a stand-alone controller, or alongside other North controllers and display systems to create a larger control or monitoring solution.

Interface Technology

Commander includes North's interface technology. Commander can access values from thousands of different third-party systems in a common way, using North drivers. This ability allows Commander to pass data between different systems, and enables different sub-systems within a building to be fused together to form a single, coherent system.



Programmable Control

ObVerse is North's block-based programming language. It is available in all North controllers. Although it is easy to use, it provides real flexibility during engineering, allowing the engineer to incorporate design changes with minimal effort. Date and timer functions are standard, along with feedback control and logic.

Information Services

Commander supports North's standard protocol, allowing communications with other North products, including powerful engineering tools and display software. Commander also generates and serves standard HTML web pages automatically - these provide a consistent user display on all browsers, including those on mobile devices. Commander can also monitor and inform users about alarm conditions using email or SMS, for example.

The engineer can extend information services if necessary by using North's driver technology: values from within Commander can be made available to, say, BACnet and Modbus devices.

Typical Use

Commander's powerful feature list means it can perform a wide range of tasks. The following are typical applications…

Single-device Integration of Systems

Creating an interface between two or more systems is easy with Commander (Fig. 1). Interlocking an air-conditioning system with an underfloor heating system is a common problem, although the possibilities are endless.

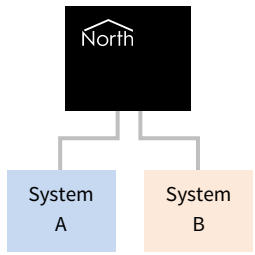


Fig. 1 Zip system

Building-wide Management System

Connecting different systems, in different locations, to a common LAN (usually an intranet or VPN) allows users to manage these systems using a common interface from anywhere on the LAN, and make adjustments as necessary.

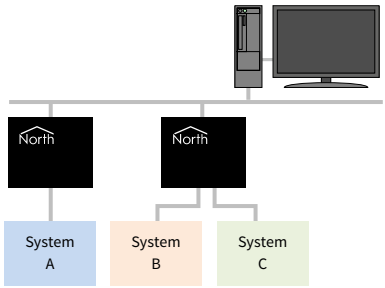


Fig. 2 Zip system

Remote Alarm Management System

Commander can be part of a remote management system that connects different systems in different buildings, and provides information back to a central display. The central display can be web-browser based, or can be North's ObSys management software.

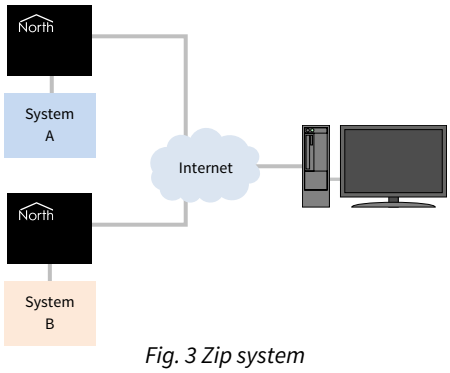


Fig. 3 Zip system

Quick Start

Mount Commander

Secure your Commander in its final position, as far away from electrical noise as possible.

If mounting Commander directly on to a wall or panel, use the chassis as a template to mark and drill holes. Screw the Commander securely to the even surface using the mounting holes.

If using the optional DIN-rail mounting kit, order code MISC\DINKIT, attach the clips to the back of the Commander using the bolts provided. Tighten the bolts as much as necessary so the clip retains its shape. Clip the Commander on to standard symmetrical 'top hat' type (TS35) DIN rail.

Leave a clearance of at least 10mm around Commander, with 90mm minimum at the cable connection edge.

Insert Battery and Set Switches

Detach Commander's lid by removing the two screws and sliding off.

Place the supplied battery into the holder, making sure to insert it the correct way round, and secure with the supplied clip. Refer to the *Internal Battery* section for more details.

Ensure FACTORY and PROGRAM switches are OFF. Set DEFAULTIP switch ON to use the default IP address of 192.168.192.167, or OFF to use a dynamic IP address from a DHCP server. Refer to the *Internal Switches* section for more details.

Re-attach Commander's lid to protect internal components.

Connect to LAN

Connect Commander to a 10Base-T compatible Ethernet switch using CAT-5 cable. Commander will detect whether a patch or cross-over cable is used, and works with either. Refer to the *LAN Port* section for more details.

Apply Power

Commander's power connector is polarity independent. Use a 12 to 24V, AC or DC, power supply rated at 3VA. We recommend using a good quality regulated DC power supply, for example 12V DC providing 500mA per Commander. Refer to the *Power* section for more details.

The Commander MODE LED will illuminate green to indicate a healthy power. Refer to the *LEDs* section for more details.

Setup Commander using North Engineering software

Install North Engineering software onto a PC on the LAN if necessary. It is available from *www.northbt.com*

Run the **Start Engineering** application.

If you are connecting to a network of devices, select **North IP Devices** then press **Scan** to rescan the LAN for the new Commander. The Commander should appear in the list – with a label that includes its serial number. If you are connecting your PC directly to Commander at its default IP address, select **Commander**.

Refer to the *Commander Tutorial* document for more details, including full walk-throughs of all the main features.

Hardware

Commander's black steel case contains a two-board device. The upper board contains the main processor, memory, LAN connection, battery and switches. The lower board contains the power regulation and the isolated RS232 ports.

Dimensions

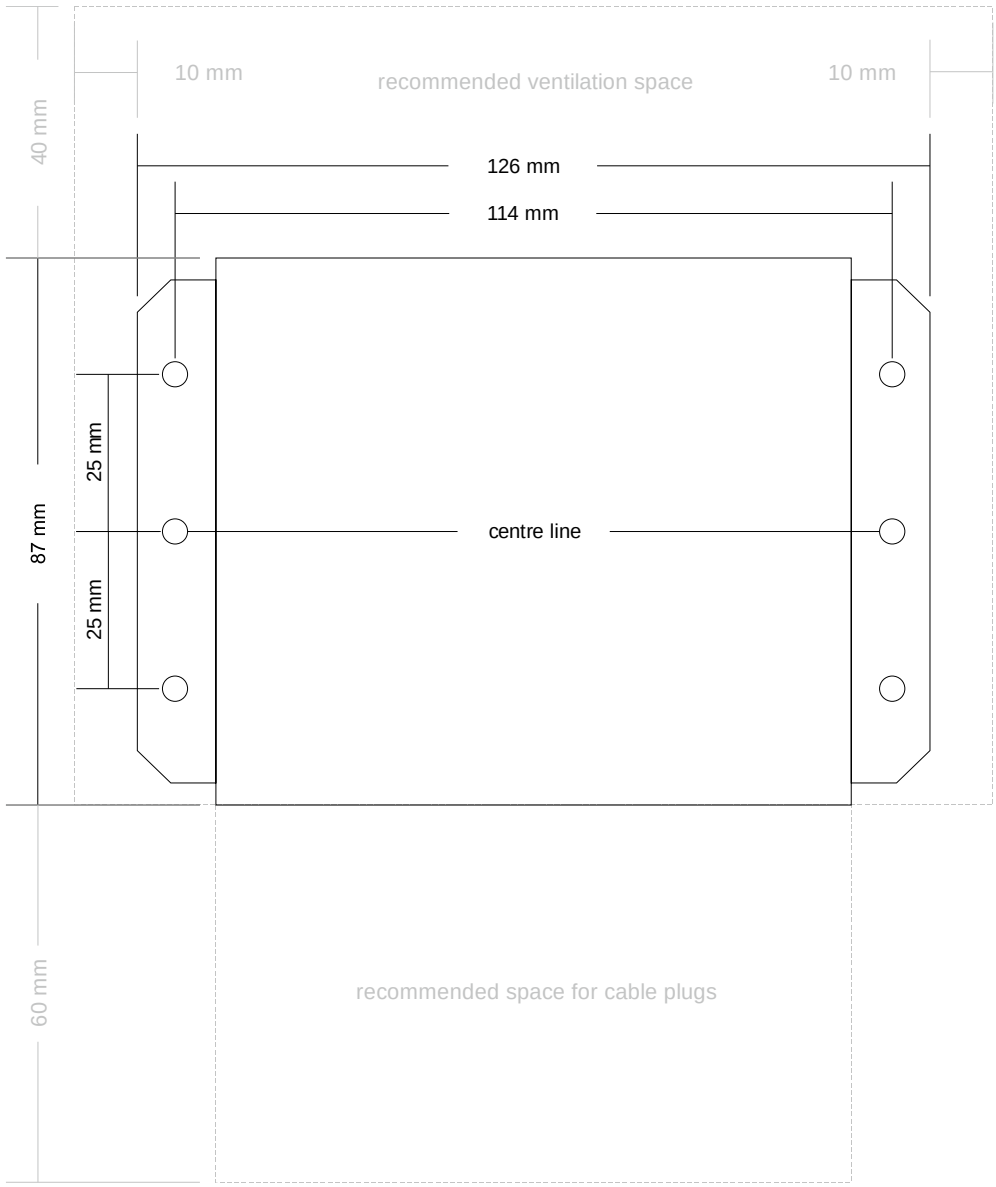


Fig. 4 Dimensions of Commander

Commander depth: 45mm

Mounting

6 x 4mm mounting holes are available (Fig. 4), suitable for M4 bolts. If mounting Commander directly on to a wall or panel, use the chassis as a template to mark and drill holes.

A DIN-rail mounting kit is available, order code MISC\DINKIT. Attach the clips to the back of Commander using the bolts provided, then clip on to standard symmetrical 'top hat' type (TS35) DIN rail.

Leave a clearance of at least 10mm around Commander, with 90mm minimum at the cable connection edge.

Connectors

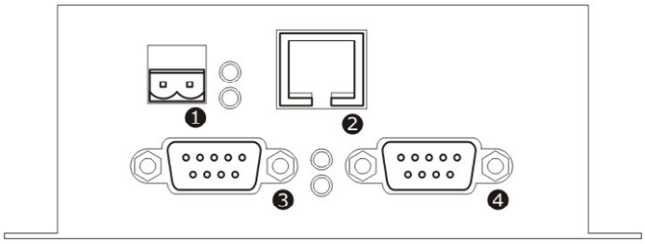


Fig. 5 Connectors on Commander

❶ Power	12-24V AC or DC
❷ Ethernet	10Base-T with auto MDI-X
❸ COM1	Isolated RS232 port
❹ COM2	Isolated RS232 port

LEDs

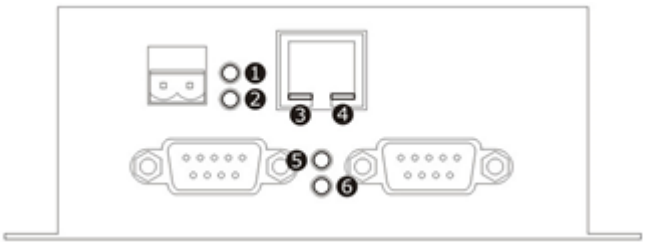


Fig. 6 LEDs on Commander

❶ MODE	On=Healthy, Flash=Check mode
❷ FLASH	On=Writing to flash
❸ LAN LNK	On=Ethernet link up
❹ LAN ACT	On=LAN port activity
❺ COM1 ACT	On=COM1 port activity
❻ COM2 ACT	On=COM2 port activity

Power

Commander requires a 12V to 24V (±10%), AC or DC, power supply rated at 3VA. We recommend using a good quality regulated 12V DC power supply, typically providing 500mA per Commander.

Commander's power connector (Fig. 5, item 1) is polarity independent.

The Commander MODE LED (Fig. 6, item 1) will illuminate green to indicate a healthy power.

LAN Port

Commander has a single Ethernet port (Fig. 5, item 2), with the following specification:

- RJ45 10Base-T
- Auto MDI-X, sensing crossover cable
- Full or half duplex with auto-negotiation
- Link-up and LAN activity LEDs

RS232 Ports

Commander has two isolated RS232 ports, COM1 and COM2 (Fig. 5, items 3 and 4). These can be used to interface either directly with RS232 devices, or with devices supporting a different hardware layer (e.g. RS485) using a converter.

Each of the 9-way DTE ports use the following pins:

1 Not connected
2 RXD – data from external system
3 TXD – data from Commander
4 DTR – held high by Commander
5 GND – the signal ground
6 Not connected
7 RTS – control from Commander, use varies depending on Interface
8 CTS – control from external system, use varies depending on Interface
9 Not connected

Internal Battery

Remove the lid to access Commander's battery (Fig. 7).

Commander requires a lithium-ion battery to retain its memory during a power-down, to preserve settings. If shipping Commander after engineering, secure the battery in position using the supplied clip.

To clear all engineered settings from within Commander, turn-off power supply and remove the battery. Wait 10 seconds, then turn-on power supply and re-insert the battery.

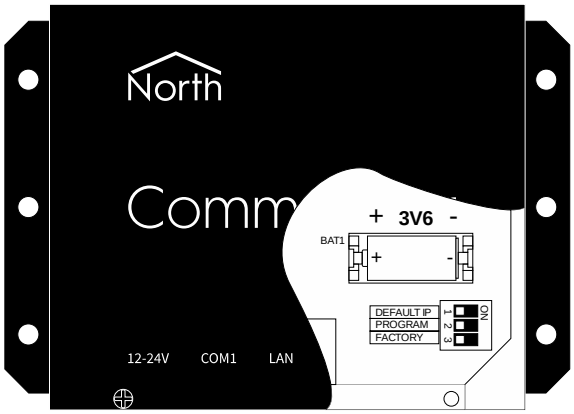


Fig. 7 Internal battery and switches

Commander measures the voltage supplied from its battery, and can send alarms when this becomes too low. You may wish to replace the battery after 10 years, regardless of whether battery has been required or not.

To replace a low battery, keep the Commander powered when you replace the battery, or you will inadvertently clear all settings.

Battery type: 1/2AA 3.6V Lithium

Caution: loose lithium batteries can cause fires, protect from short circuit.

Internal Switches

Remove the lid to access Commander's switches (Fig. 7). Commander only reads the switches at power-up, so re-power after changing.

The **DEFAULTIP** switch selects if Commander should use the default or an assigned IP address. When set ON, Commander is set to use the default IP address of 192.168.192.167. This is useful if you do not know the assigned IP address of Commander. When set OFF, Commander uses the DHCP or engineer assigned IP address. Refer to the *LAN Port Setup* section on how to assign an IP address.

When the **PROGRAM** switch is set ON, it enables the installing of new software into Commander's flash memory. It also enables other services – for example, Telnet. Be cautious therefore when deciding to leave a Commander in PROGRAM mode. Refer to *Appendix C* for a full list of services enabled.

The **FACTORY** switch should always be set OFF.

Memory

Commander has 4MB of flash memory, which holds both its main operating firmware and driver firmware. Refer to the *Firmware* section for details on how this is utilised.

The main operating and driver firmware can be updated to the latest version, and new drivers added via the LAN port. Follow the guide in the *Updating Commander's Firmware* section.

Commander also has 2MB of RAM, which it uses to hold all configurable settings – this is battery-backed to maintain the memory over a power-failure. The battery will maintain the memory contents for several years.

Interface Licences

North supply Commander with a certain number of interface licences, which control the number of interfaces that can be used at a time. Each interface usually requires one interface licence before it will start, although certain drivers require zero licences.

It is possible to upgrade the number of interface licences available within a Commander. Simply call North support while you are on site to activate a new licence.

Refer to the *Interfacing Commander to other Systems* section for more details.

Firmware

Commander's operation, and all of its power and flexibility, is controlled by firmware stored in its flash memory.

Main Firmware

The main operating features are provided in firmware. This firmware is loaded into Commander during production and can be updated if required.

The main operating features evolve over time as newer ideas are incorporated, but contain the following areas:

- General platform information, such as firmware versions
- LAN port setup, including IP addressing and time synchronisation
- Real-time clock and time zone, including daylight saving
- Interfacing to other systems
- Transferring values between systems
- Database of essential values
- Controlling using time and date
- ObVerse cause-and-effect strategy
- Alarm handling, including filtering, delivery, and audit trails
- Communicating with other North IP devices
- Security server
- Telnet server
- Web server

Interface Driver Firmware

Interface drivers are provided in individual firmware files, to simplify updating. The most popular interface drivers are loaded into Commander during production. Less popular interface driver firmware can be downloaded as required. Interface drivers can be updated as newer versions are produced.

Popular interface drivers include:

- Modbus – including Modbus over TCP/IP and serial line protocols
- BACnet/IP – as both a client and a server
- ZipMaster – to communicate and control a North Zip network
- JSON – an API for third-party software and web services
- Echelon LonWorks – for accessing LonWorks compatible and LonMark certified devices
- Kentec – one of many available fire panel interfaces
- Galaxy – one of many security panel interfaces
- Crestron – one of many user-interface connections
- APC – for UPS status

For the latest list of drivers available for Commander, visit *northbt.com/go/drivers*.

Functional Organisation

The different areas of function within Commander (or any other North product) are organised, and therefore engineered, in the same way using objects.

This object system, which North have used in products for many years, is extremely simple to understand, and yet gives a flexible, extendable system.

The object system has two simple rules:

- Each device appears as an **object**
- Each object can either **contain** other sub-objects, or have a **value** that can read (and usually set)

Commander appears on the LAN as an object; it contains sub-objects that represent the sub-functions within Commander; each of these contains sub-objects representing sub-functions; and so on until we get to sub-objects that have values, which can be viewed and possibly adjusted.

Functions

Functions within Commander are represented by container objects. A container object can either have a pre-determined list of sub-objects (called a fixed container object), or can have a variable list of sub-objects (called a variable container object).

North's object engineering software shows the contents of an object, and allows the engineer to navigate up, down and across these container objects. It also allows the engineer to scan and find the sub-objects within any variable container object.

For example, Commander is a variable container object with its contents changing as interfaces are started and stopped.

Settings

Settings within Commander are represented using value objects. The value of an object can be viewed by engineering software or by other devices; some value objects can also have their value adjusted.

North's object engineering software shows the current values of these objects, and allows the engineer to make changes to objects that are adjustable.

For example, Commander's label is a value object that can be viewed and adjusted.

Organisation of this Manual

The following chapters in this manual describe the functional areas within Commander. These areas appear as sub-objects of Commander within the object engineering software, ObView.

Refer to the *Commander Tutorial* for more help on using the object engineering software to engineer Commander, including full examples.

The final chapter, *Object Specifications*, details the contents to each of Commander's container objects. It lists each of the fixed and variable container objects, and describes the type and attributes of each value object.

Platform Information

The *Platform Information* area provides general information about Commander.

Switch Settings

It is possible to view the mode specified by Commander's switches, although objects are read-only. Because the values are available as objects, they can be used by the engineer to affect other things, such as enabling cause-and-effect strategy, or enabling security for certain users.

Battery Level

The state of Commander's battery is available as an object. Commander will also send a battery status alarm when the battery voltage reduces to a low level (3V) or very low level (2V). If necessary, the battery state can be transferred to other external devices, such as lamps and sounders.

Reset/Restart Information

Commander holds information about the last restart to occur, including the assumed cause. This information is sometimes useful when solving problems.

Firmware Versions

Commander's firmware is made up from a collection of different modules. The software version and build date for each module is available, again to help when solving problems.

Debug Recording

Commander can send information to an unused COM port to assist with debugging. The North support team may request that this information is captured.

A set of flags specify which debug information is recorded:

Information – general start-up information
Objects – request/reply messages between modules
COMs – bytes being sent and received on COM ports
Network – IP events, including ARP and DHCP
Internal Comms – fast request/reply messages between internal modules

An ASCII terminal, or PC running terminal software, is needed to receive the debug information. The terminal needs to be set to 38400 baud, no parity, 8 data-bits, 1 stop-bit, and no hardware flow control.

An RS232 null-modem cable is required to connect the terminal software to Commander.

LAN Port Setup

The LAN Port area allows the set-up of Commander's LAN port. The most important value is Commander's IP address, which allows other devices, including engineering software, to access Commander from elsewhere on the LAN.

There are three ways of assigning Commander with an IP address to use:

Default – a predetermined IP address, controlled by a hardware switch within Commander
Static – set to an engineer-assigned fixed address
Dynamic – set to an address assigned dynamically by a DHCP server [Factory setting]

Whichever way is used, Commander's current IP address is available within a read-only object.

Default IP Addressing

Setting the DEFAULTIP internal switch to ON (and rebooting), assigns Commander to work at the default IP address of 192.168.192.167 with a subnet mask of 255.255.255.0. This is useful for quick engineering, and if the Commander's IP address is unknown. Default IP mode overrides other IP address configuration. This method is not usually compatible with a real LAN, but more for connecting directly from Engineering PC to Commander

Static IP Addressing

Configuring an IP address and subnet mask assigns Commander to use that IP address – you can also configure a gateway, DNS server, and time server addresses. This is the usual way of configuring the IP address, as users and other devices usually need to know where Commander is.

Dynamic DHCP Addressing

Configuring an IP address of 0.0.0.0 instructs Commander to request its IP settings from the LAN's DHCP server – the server normally also supplies DNS and NTP server addresses. Dynamic addressing is simpler that static addressing, but the IP address given by the DHCP server to Commander may change after a period.

This is the default setting. If you wish to use Commander in this way, please ensure that the DHCP server will reserve addresses, and not re-assign them to other devices. The MAC address is listed to assist in creating a DHCP reservation.

DNS Server Address

Commander sometimes needs access to a DNS server to resolve fully qualified domain names (FQDN). It is usual for the DHCP server to provide the server's IP address, but otherwise the engineer can configure it using this object.

Time Server Address

Commander can request time information from an NTP server, but again needs to know the server's address. It is usual for the DHCP server to provide this, but otherwise the engineer can configure it.

Real-time Clock and Time Zone

Commander contains a real-time clock, time-zone information, as well as daylight saving offsets. Commander can use all of these to determine local time.

Coordinated Universal Time

Commander contains a hardware real-time clock (RTC). The internal battery retains settings when the Commander is powered-off. This clock holds Coordinated Universal Time (UTC).

If the Commander has access to an NTP time server, it will automatically correct its clock. Commander requests the time from the NTP server every 4 hours, and if the response is acceptable, the real-time clock is adjusted (if out by more than 2 seconds).

Local Time

Commander calculates local time by applying a local time offset to UTC time (as well as any daylight-saving offset). By default, Commander's local time offset is zero, which is correct for the United Kingdom and Ireland. In mainland Europe, the local time offset should be set to +1 hour.

If the local time is adjusted manually, Commander will automatically convert the value to UTC before writing to the real-time clock.

Daylight Saving Time

Commander's daylight saving table is set during production with 10 years' of dates and offsets suitable for use within the European Union. If used outside of this area, the dates and offsets within the table should be set as necessary.

Interfacing Commander to other Systems

One of Commander's strengths is its ability to interface to several systems, including those from other manufacturers. Drivers are the software convertors that allow interfaces to work.

The driver required for a particular interface is selected from those installed in Commander. Commander starts these interfaces when they are specified, and whenever Commander is restarted. Drivers can be stopped, if they are no longer needed.

Started Interfaces

Commander allows up to four interfaces to be started and used at the same time, although interface licences may limit this.

Installed Drivers

The list of drivers currently installed within the Commander can be viewed – and the names copied to the current interface list to start an interface using a particular driver. The same driver can be used on several interfaces concurrently. For example, Commander can start interfaces to two separate Zip systems, each with 16 Zip modules.

Commander has certain drivers pre-installed. For a list of these, see *Appendix B - Pre-loaded Drivers*.

Other drivers can be installed at any time. For details on how to do this, see *Updating Commander's Firmware*.

Interface Licences

Each Commander is supplied with a certain number of interface licences. It is possible to add more licences on site, although a phone call is required to check/verify settings.

Call North support on +44 (0) 1273 694422 for more help with adding more licences.

Order Codes for Commander are as follows:

North Order Code	Interface Licences
CMDR/0	0
CMDR/1	1
CMDR/2	2
CMDR/3	3
CMDR/4	4

Order Code for an additional interface licence

North Order Code	Interface Licences
UPGD/1	1

Transferring Values

Transferring values from one place to another is the simplest form of integration. Commander supports this with Data Transfers – where each transfer reads from one place and then writes to another.

Commander supports up to 500 Data Transfers.

Transfers are processed sequentially. Each transfer contains a source object reference, a source read rate, the last value read, a destination object reference, and a destination write rate.

Commander's object system means that any value available from Commander, including configuration objects and other attached systems, can be used with transfers.

Reading the Value

The source read rate should be configured to suit the type of value being transferred. For example, room temperatures are typically read every minute or so, and operating set points read every 15 minutes.

Setting the source read rate to 'as soon as possible' (ASAP) instructs Commander to read the value every time the transfer is processed – if all transfers have a read rate set to ASAP, the reading occurs sequentially.

Remember to consider the time needed to read values, because they do not occur instantaneously. If it takes 1 second to read a value from an external system, then it will take 100 seconds to read 100 values. Even if you specify the 100 values read every minute, the best that could occur would be every 100 seconds.

Writing the Value

Rather than read-then-write continuously, each transfer will usually only write the value to the destination object when the value changes – called change-of-value (COV) writing. This eases the workload within the destination system and saves time and effort within Commander.

It is also possible, to have the transfer perform a background write periodically. Occasionally it is useful to have the last value re-written in this way – for instance, if a system forgets values after a power failure, or if a local adjustment needs to be reset after a certain time.

Remember COV writes always happen, so consider if a background write is needed and how frequently it needs to occur. Writing the value takes precedence over reading, so set the write rate to a longer time than the read rate. A background write only usually needs to occur every few hours.

Transfer Tasks

By default, Commander works through the list of transfers sequentially – after finishing transfer 1, it starts transfer 2, and so on, until it reaches the last transfer, after which it starts work on transfer 1 again.

This ensures simplicity, but can be quite slow. If values are from several different systems, it is possible to allow the Data Transfer system to try to perform two or more transfers simultaneously. However, this feature should be used with care, and should be fully tested before handover, because if several tasks are reading from or writing to the same system, overloading of the system's communications can occur.

Essential Data

Several different modules within Commander need access to a database of values – called Essential Data.

Essential Data allows the engineer to configure a list of values that other Commander modules can then distribute – as web pages, as BACnet points, as Modbus points, on Zip displays…

Structure

Essential Data consists of a list of configurable pages, each of which has a list of configurable objects. In total, 640 database objects are available. The engineer may choose from a mix of page and object counts: 40 pages of 16 objects (default), 64 pages of 10 objects, 20 pages of 32 objects, or 10 pages of 64 objects. Older versions of Commander only support 30 pages of 16 objects (total 480 objects).

An Extra Data driver is also available to extend the number of database values available within Commander.

Pages

The engineer configures a label for each page. Other Commander modules use the label when displaying the value from within the database – if a page has no label, then the page does not appear in label-based hierarchical views, such as those available as web pages.

Each page has a remote object prefix. This allows objects within the page to reference remote data relative to this prefix. When in use, if any object in the page fails to read its remote object, Essential Data assumes all of the objects will fail, and therefore saves time and effort. It also allows easier copying of pages when several devices of one type are used.

Objects

For each object within a page, the engineer configures a label. The label determines whether the object appears – if an object has no label, the object does not appear in label-based hierarchical views.

The engineer configures the value type for an object. It determines whether a value is made available via a particular protocol – for example, profile objects are not made available over the Modbus protocol. This also controls the appearance of the value on hierarchical views, including when the user adjusts the value.

Type	Use	Example
Text	Text string of ASCII characters	Kingston Lane
NoYes	Binary state: 0 or 1 meaning No or Yes	1 *(means Yes)*
OffOn	Binary state: 0 or 1 meaning Off or On	0 *(means Off)*
Number	Integer, positive or negative	-12300
Float	Floating-point value, with decimal places	21.54
Times	List of on-off times	08:30-12:00,13:00-17:30
DateTime	A moment in time	25/12/15|12:30
Date	A particular day	01/01/19
ENum	A number representing something	3 *(means Automatic)*
Profile	List of time-value changes	09:00=21,12:00=20,18:00=19

Value Reading and Writing

The engineer can configure the database object to read periodically from, or write periodically to, any object accessible to Commander. This allows slow-access data to be collected and ready for speedy delivery when needed.

As a new value is read, the object also recalculates the alarms state, sends alarm messages if necessary, and logs the value.

Commander considers each object in a linear fashion, beginning with the first object on the first page. If it is time to read (or write) the value, this is started; if not, the object is passed over. When the final object has been checked, Commander starts from the first object again.

If all objects are set to read ASAP (as soon as possible), Commander performs each in turn. If it takes one second to read an object, and there are five hundred objects to be read, the whole process will take five hundred seconds. Please consider the read rate of objects, allowing time for the object to perform its remote action.

Adjustability

By enabling adjustability, along with configuring value high and low limits, the engineer can control whether the user can modify the value of an item – and if this occurs, the value will be written to the associated object.

Alarm Monitoring

When a value is updated, it is compared against the value high and low limits, and an alarm message generated as necessary if its alarm state changes.

Value Logging

If necessary, Essential Data can log the value periodically, to create a history of the value over time. Other devices can access the data log.

Essential Data uses the configured log rate to sample the value – if the log rate is set to 15 minutes, one day will use 96 readings.

A total of 40 objects can have logging enabled.

If the value has not changed from the last sample, then rather than waste sample space, the log records start-and-end sample times to save space. Using this method, 15 minute sampling will use a maximum of 96 readings per day, but could be considerably less if, for example, the value is an alarm state.

Controlling using Time and Date

Timers allow users to control when things happen in the day, and when they do not. Using timers can save energy, while still keeping the occupants happy. Commander supports time control using the calendar, timers and profilers within it's Time Control area.

Commander's single calendar determines today's day-type. Today's day-type is used by 20 timers to determine which of their on-off time-lists to use, and by 20 profilers to determine which of their time-value profiles to use.

Calendar

Commander supports 10 different day-types: one of them is an off day-type, leaving nine to configure. They are numbered 0 (off), and 1-9.

If you have a centralised calendar elsewhere on the system, Commander can request the day-type from this – in which case you need to specify the object reference of the current day-type in the central calendar.

If you are calculating the day-type in Commander, it works in the following way. The calendar determines whether today's date is an exception date – if it is, then that exception day-type is used; otherwise the day-type of the standard day-of-week is used.

The calendar re-calculates the day-type every minute, based on the day-types and the exception dates.

Any North device that can access Commander's objects can also access the calendar objects. The Calendar is also available via the Web Server.

Timers

Commander uses timers to control off/on processes. Each timer produces an off or an on state, which can be accessed by other tasks (Fig. 8).

Each timer has a time-list for each of the possible day-types, and uses that time-list on days that have that day-type. A time-list is a list of on-off time-periods. The timer re-calculates the state of the timer every minute.

Any North IP device can access the Timers. They are also available to view and edit via the Web Server.

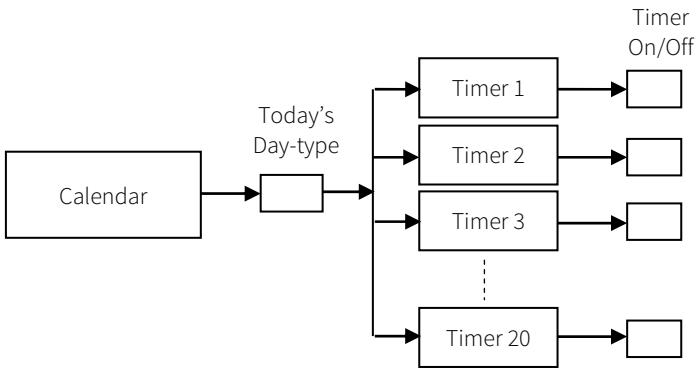


Fig. 8 Timers

Profilers

Commander uses profilers to control a variable value throughout the day.

Each profiler has a list of profiles, one for each possible day-type, and uses that profile on days that have that day-type. Each profile is a list of time-value pairs, which specify the time to change to a particular value.

ObVerse Cause-and-Effect Strategy

Sometimes you need to do more than simply transfer a value from one object to another – you need to calculate something, delay something, or perform a more complex function on a value. North provides this flexibility with ObVerse, a cause-and-effect programming language.

ObVerse consists of a range of modules. The engineer selects particular modules and links them together to perform a desired strategy.

ObVerse strategy runs in an ObVerse processor within a device.

ObVerse processors come in two types:

- Standard Processor – with logic, maths, and control modules
- Advanced Processor – with the same features found in a standard processor plus extended maths and logic, display, application execution, directory and file services, and user-defined modules.

Commander has two ObVerse standard processors.

ObVerse strategy is made up from properties, modules and comments.

Properties

ObVerse properties are containers for storing data values. They can carry a value from one module to another, or between the processor and other functions within Commander.

Properties have a data type, to define the type of value they hold – like a number or a text string.

Properties sometimes hold values passing only between modules in the same processor. In ObVerse, we call these private properties, as their value is private to the processor.

Properties sometimes hold values passing between the processor a function within Commander. In ObVerse, we call these public properties, as their value is publicly available as an object.

Modules

Modules calculate values. They take one or more inputs, and calculate one or more outputs.

Different modules are available to perform different operations. The range of modules supported depends on the processor.

In Commander's ObVerse standard processor, modules perform the following types of operation: Maths, Logic, Control, Timers, System, and Object.

Editing ObVerse

You can create and edit ObVerse strategy using North's ObvEditor application, installed as part of the ObSys software. ObvEditor provides drag-and-drop graphical editing of ObVerse, uploading and downloading of ObVerse strategy, and run-time monitoring of the strategy within the processor.

For further information on ObVerse in Commander, including properties and modules, refer to the *ObVerse Manual – Standard Processor*.

Alarm Basics

As well as responding to object requests, Commander can process alarm messages – messages that contain text information about events that have occurred.

Different users demand a wide variety of alarm processing – ranging from simple histories of alarms, to lists of critical events that require user acknowledgement, to sending messages to mobiles phones or by email.

Format

All North-format alarms are text-based, and have the following fields:

System and Point – identity of the system, device, and point that has changed

Condition – condition that the point has changed to

Priority – importance of the notification, between 1 (most) and 9 (least)

Date & Time – date and time that the condition changed

Refer to *Object Specifications* section for details of alarms sent by each module.

Alarm Delivery

All alarms arriving, or being produced by Commander are passed into the Alarm Delivery area. Its role is to distribute copies of the alarm to other alarm processing destinations, depending on the value of the alarm fields.

Commander's alarm delivery supports up to 16 destinations.

Commander has two built-in alarm processing destinations: Alarm History and Alarm Emailer. Other destinations are available within other North devices, and can appear within drivers and external systems as interfaces are started.

Filtering Delivery

Besides having a destination enable, alarms can be filtered, so that only alarms with a priority in a certain range are sent to a destination. This allows only the relevant alarms to be sent to certain destinations.

Filtering can also cover text matching and not-matching, to force alarms from particular systems to be sent only to one destination, for example, while all other alarms are sent to a different destination.

Delivery to any one of a Group of Destinations

Sometimes users may need alarms sending to any of several destinations. For example, alarms must be sent to any one of a range of on-call users (depending upon which has their PC running).

Commander supports each destination being a member of the any group, an indication that when one destination accepts the alarm, it does not send to any of the other members of the any group.

History of Alarms

Sometimes, all that is required is a list of the latest alarms. This can act as an audit trail, or can be viewed more regularly to check on new alarm conditions. However, no user action is necessary, and as new alarms occur, old alarms are lost.

Commander's alarm history holds the last 100 alarms.

Emailing Alarms

Some users want important alarms and event messages delivered directly to their inbox. Commander has alarm emailing facilities to do this.

Commander's alarm emailer supports six destination groups. Each group has its own filtering options, so only alarms from a particular system are emailed to the group's recipients, and an option to send in a text-only or friendlier HTML format. A group supports up to five email address recipients.

The Alarm Emailer can connect to an SMTP relay server that supports authentication, but not to a server that requires encryption. So you can typically connect to an SMTP relay server on your local network (e.g. Microsoft Exchange), from your Internet service provider (e.g. BT Internet, Virgin Media), or from a paid SMTP service provider (e.g. smtp2go.com, authsmtp.com, serversmtp.com, etc.).

SMTP relay servers that require TLS or SSL encryption are not currently supported (e.g. Gmail, Outlook, Office 365, etc.).

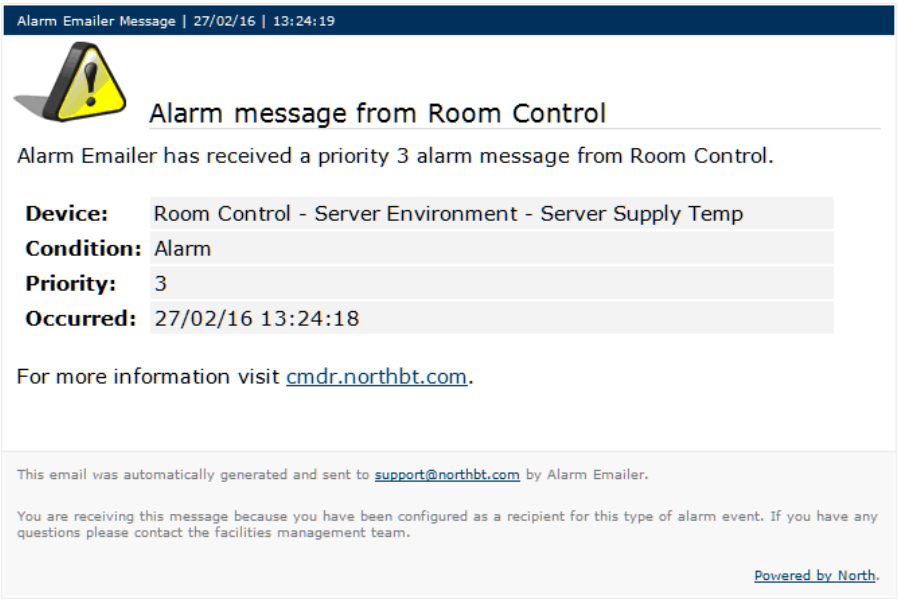


Fig. 9 Email from Commander's alarm emailer

Alternative Alarm Destinations

By adding more interfaces, it is possible to add more ways of dealing with alarms, these include the following:

Driver	Use
AlarmStore	Alarm Lists needing user acknowledgement, and archiving to CSV file (ObSys only)
AlmSet	Convert alarm text to an object write
AlmXlate	Translate alarm text
Compass	Send via a Compass network
ESPA444	Pager support
GSMSMS	SMS messages to mobile phones, via a GSM modem
Printer	Print alarms

Communicating with other North IP Devices

All the North devices that support Ethernet also support the North IP protocol – Commander and ObSys. The North IP protocol allows the devices to communicate efficiently using objects.

Key Security

It is easiest to allow all North devices on a network to communicate openly. However, it is possible to add an authentication key to a device. This will stop other devices being able to communicate with it, unless they also know the device's authentication key. Each device could have a unique key – so Device A may be able to request values from Device B, but not vice versa.

Checking Communications

It is possible to configure Commander to check that other North IP devices are still available and communicating, and generate alarm messages when they are not.

Security Server

When Commander is connected to a network, the engineer must consider security – to control who can and who cannot view and modify values, and who can engineer Commander itself.

Although it is possible to enable and disable certain engineering functions using the PROGRAM switch, this is quite restricted.

North products, including Commander, that require user authentication have access to a central user database that holds user information. Remote doors could use this, for example, to ask the central user database for authentication when a user requires access.

Commander has Security Server, its own user database – it holds information, including privilege levels, for different users.

Commander's Security Server can support up to 400 users.

It is possible for a task within Commander to use a security server elsewhere in the system – for example in a different Commander or within a PC running ObSys.

User Information

Besides a name, each user has a user ID (or card) and a password – together these form a coded token. It is the coded token that is passed around a system – the password is never seen.

Each user is given eight privilege levels, one for each of eight different areas. The privilege level is in the range 0 (no security clearance) to 7 (maximum clearance).

Individual users can also be enabled or disabled, and can be limited to access between certain dates –it is possible to see the last date that a security clearance was requested for the user.

Groups

Each user can be a member of up to three groups. Each group has a list of privilege levels, which act as a base level for users that are a member of the group. This allows a whole group of users to be controlled quickly and easily.

When groups are used, a user is enabled when any of the groups they are a member of is enabled (and the user is enabled).

The user's privilege level in a particular area is the highest level specified in that area from the individual user information and from any group they are a member of

Use by other Tasks

A task that requires security clearance will have one or more access security objects. Each access security object has a two-digit value, which controls access to a particular feature – such as adjusting a value, or viewing a page.

The two-digit value is made up of the area digit (1-8), followed by the minimum privilege level (1-7). The user must have a privilege value equal to (or greater than) that specified, in the area specified, before the task will allow access to the feature.

For example, if the minimum privilege level is '6' in area '2', then the two-digit value is '26'. If the value is set to '00', then no security checks are needed.

When the user wants to access the protected feature, the following occurs:

- The task requests identification from the user: this may be done by asking for a name and password; or by scanning a security ID card
- The task encodes the identification, and requests the privilege levels for the user with that identification from the Security Server
- The task receives the eight privilege levels from the server
- The task checks whether the user's privilege level in an area is sufficient to allow access, and if so will grant access
- The task may send an alarm message, indicating whether the user was allowed, or whether the ID was not recognised.

Web Server

Commander's web server builds pages with simple HTML, which the user can view in any browser, including those on mobile phones and tablets (Fig. 10).

The web server creates a website automatically, with information from the following functional areas:

- Essential values
- Extra values
- Calendar and timers
- Alarm history
- General support information

If necessary, it is possible to disable the web server completely, or disable the displaying of information from certain areas.

If you are browsing on to Commander with a smart phone or tablet, you may wish to choose the 'Mobile site' link, which displays pages optimised for smaller screens.

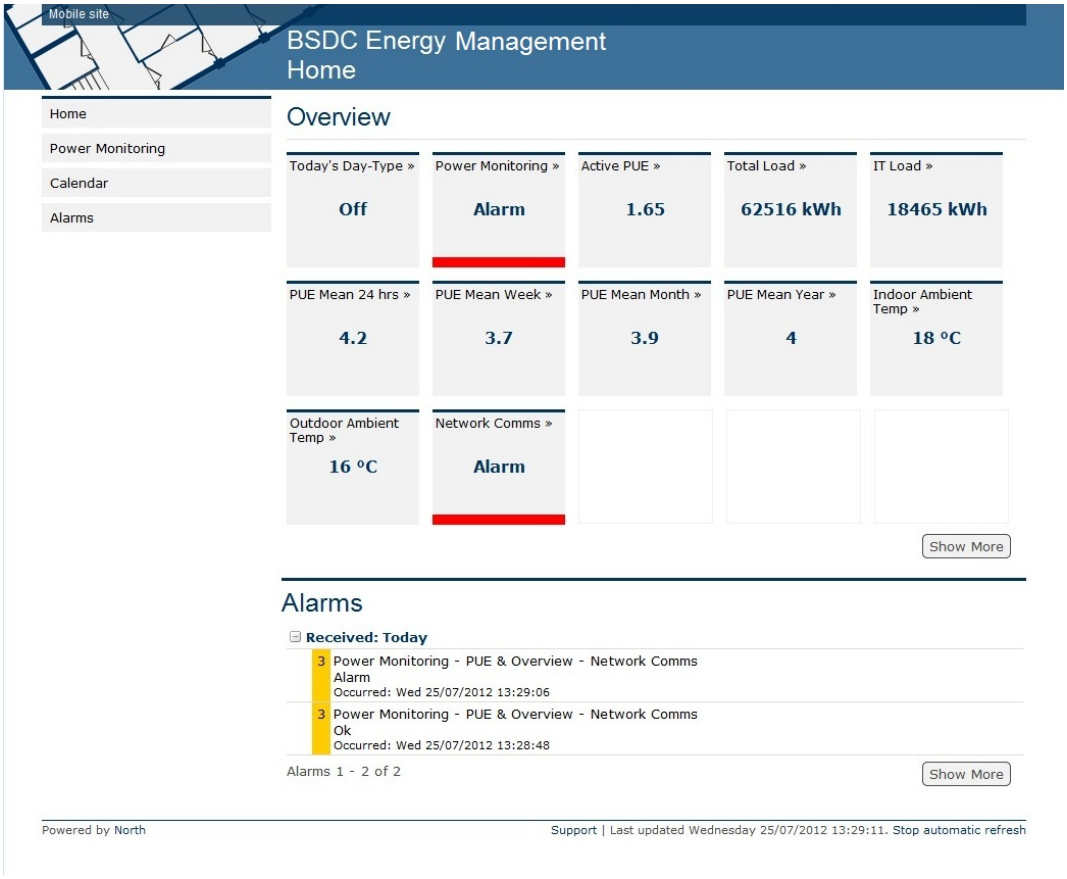


Fig. 10 Page from Commander's web server

A demonstration of Commander's web server is available: *cmdr.northbt.com*

Defining the Page Template

All web pages are built using a template. This contains links to external web sites, which could be customised to link to other Commanders or a company web site.

Enhancing the Home Page

Commander's home page acts as the gateway to values within Commander, as well as providing summary information.

Home page widgets are simple blocks that contain labels, values, and alarm state of objects within Essential Values - from either the objects within page 1, or the first objects of each page.

Recent alarms, occurring within the last 7 days, are also shown on the home page.

Security Settings

It is possible to specify whether users must sign-in before they can view or adjust values on the website.

Support Page

The support web page provides a summary report of Commander's operating state:

Platform Information – date & time, version, hardware mode, and battery state

LAN Port – current IP, subnet, gateway, DNS and time server address

Interfaces – list of interfaces started, along with version and state of connection to external system.

Configuration Page

The configuration web page replaces the support page when Commander's PROGRAM switch is set ON.

In addition to providing the same information available on the support page, the configuration page allows the Commander's label, local date & time, and IP settings to be adjusted. This is useful if the engineer needs to change the IP settings of Commander, and has no access to object-engineering software.

PROGRAM Switch

When PROGRAM switch is set ON, the web server's security is disabled, and the configuration web page becomes available.

Telnet

Sometimes it is necessary to talk to Commander without using web pages or engineering software. Commander has a Telnet server that you can enable, and can provide simple text-based access to any object values within Commander.

Commander supports two services within the Telnet session – query-response and IP-configuration.

Establishing a Session

Establish a Telnet session to Commander by opening a TCP/IP connection on port 23.

Messages are formatted as a line of ASCII text. Each line of text, or message, must end with a carriage-return (control code 0x0D) and line-feed (control code 0x0A) – represented in this manual using the symbol: ↵.

By default, Telnet is disabled, but you can enable Telnet, and configure a user-name, to act as a simple password.

Once connected to Commander, enter the user-name and then the service:

```
Telnet - North Telnet
User:*****↵
Service:
```

For a list of services available, enter '?' at the service prompt.

To close a session, send a line-feed at the service prompt. Alternatively, the session will automatically close after 5 minutes of inactivity.

IP Configuration Service

The Telnet IP-configuration service is used to discover the current network settings of Commander.

At the service prompt, enter 'ipc' and Commander will respond with the current IP configuration:

```
Service:ipc↵
IP Configuration..
Network 1:
   Address. . 192.168.192.167
   Mask . . . 255.255.255.0
   Gateway. . 0.0.0.0
```

Query Response Service

Using the Telnet query-response service, you can request and adjust the value of any object within, or connected to, Commander.

At the service prompt, enter 'qr' and the Telnet session will enter query-response mode:

```
Service:qr↵
Q:
```

Enter the query command at the Q: prompt and Commander will respond at the R: prompt.

Reading a Value

At the query prompt, enter the object reference to read. Commander responds with the object value.

```
Q:object↵
R:value
```

For example, to read the local date & time (object O.T):

```
Q:O.T↵
R:01/12/15|15:40:08
Q:
```

Writing a Value

At the query prompt, enter the object reference and value to set. Commander will respond with 'Ok' to indicate the object adjusted successfully.

```
Q:object=value↵
R:Ok
```

For example, to set the Commander label (object O.L) to 'New Commander':

```
Q:O.PL=New Commander↵
R:Ok
Q:
```

Error Response

If the service encounters an error when processing a query command, it will respond with the error prompt (E:) followed by one of the following three-character error code:

Error Code	Reason
OBJ	Object reference invalid
ACT	Action invalid – cannot read/write this object
VAL	Value invalid
FLT	General fault
DDV	Device delivery fault
PDV	Point delivery fault
NDV	Network delivery fault
???	Unknown error

For example, attempt to read the invalid object reference ABCDE:

```
Q:ABCDE↵
E:OBJ
Q:
```

PROGRAM Switch

When PROGRAM switch is set ON, Commander automatically enables Telnet with a user-name 'PROGRAM'.

Updating Commander's Firmware

When supplied, the Commander hardware contains the base Commander software, along with several popular North drivers. For a list of these drivers, see *Appendix B*.

North provide other less-used drivers in separate CDM files, but you need to install these drivers when required, before you can use them. North also update the drivers and CDM files, and you may need to install these.

For installing files, Commander supports the Trivial File Transfer Protocol (TFTP).

What is TFTP?

TFTP is one of the standard IP protocols. TFTP requires a client (your PC) and a server (Commander).

To enable TFTP client on Windows Vista and later, follow these steps:

→ From Windows **Control Panel**, select **Programs** (or **Programs and Features** in Vista) then **Turn Windows features on or off**

→ Select the check box next to **TFTP Client** to enable it, then click **OK**

The TFTP client is a DOS-style command, with the following command-line format:

```
TFTP -i <serverIPaddr> put "<filename>"
```

where *<serverIPaddr>* is the actual IP address of the Commander, and *<filename>* is the file name to transfer.

Locating the CDM Files

ObSys setup installs CDM files within the CDMs folder of the ObSys Program Files folder – for example "C:\Program Files\North Building Technologies\ObSys\CDMs".

Installing or Updating a CDM

Commander only opens its TFTP server port when PROGRAM switch is set ON. This means that without this switch enabled, Commander cannot be modified accidentally or maliciously.

North distributes drivers for Commander in CDM files – these files are made to work in certain areas of Commander's memory – called banks. You can only load one CDM file per bank, so you must choose which CDM to load in each bank.

To load ZitonZP.CDM to Commander using your TFTP client, follow these steps:

→ Power Commander OFF, and change the internal switch PROGRAM to ON

→ Re-power Commander and it will start working in Program mode – it will enable its TFTP server – the MODE LED will flash, to show Commander is in Program mode

→ On your PC, run the Command Prompt

→ At the command prompt, type the following to change to the CDMs folder:

```
CD "%PROGRAMFILES%\North Building Technologies\ObSys\CDMs"
```

→ Type the following to send the file :

```
TFTP -i 192.168.2.150 put "Bank7 ZitonZP.cdm"
```

→ Once Commander has received and checked the file, Commander will write the CDM to flash memory (showing its FLASHWRITE LED), and when complete, will restart, and the MODE LED will flash to show it is in Program mode again

→ Close your Command Prompt window

→ Power Commander OFF, and change the internal PROGRAM switch to OFF

→ Re-power Commander, and it will start working in RUN mode – it will disable its TFTP server

Updating the base Commander software

If you need to update the base Commander software itself, update two files: ‘CmdrBase.bin’ and ‘CmdrExt.bin’.

You load these files individually using TFTP in the same way as the CDM files.

Object Specifications

Once Commander has started, several objects become available within the top-level object of the device. These objects may contain sub-objects (and each of these may contain sub-objects, and so on) – the whole object structure being a multi-layer hierarchy. It is possible to navigate around the objects using the ObSys Engineering Software.

Each object is specified below, along with its sub-objects.

Example Object Reference

An example of a reference to an object within Commander: the Configuration object (O) contains Platform Information (PI), which contains Commander Version (PV) – therefore the complete object reference is 'O.PI.PV'.

An example of a reference to an object in a different device: the IP network object (IP) contains Default Commander object (CDIP), which contains the object above (O.PI.PV) – therefore the complete object reference is 'IP.CDIP.O.PI.PV'.

Device Top-Level Objects

Object Type: *[Commander v20\Device]*

When Commander is powered-on, the objects below become available within the top-level object of the device.

Description	Reference	Type
Commander Label Label used to identify this device, including scanning and within alarms. Adjust within the Configuration object.	PL	Obj\Text; Max. 30 chars
Configuration Set up Commander's main features	O	Variable Container: *[Commander v20]*
Essential Values Contains the values configured within Essential Data	UD	Variable Container: *[UserData\PageList]*
Time Control Time and date based control using calendar, timers and profilers	CT	Variable Container: *[CalTimer v20]*
ObVerse Contains public properties configured in ObVerse Processor *x*. This object is only available once ObVerse has been downloaded to the processor. The processor number, *x* is in the range 1..2. *ObVerse* is set in public property L within the ObVerse.	P*x*	Variable Container: Type is based on the filename of the ObVerse. For example if saved as the file 'TypeInfo\ObVerse\Process.obv', then the type will be: *[ObVerse\Process]*
Security Server Edit users and groups in the database for authentication	TK	Fixed Container: *[TokenMax v20\400]* *[TokenMax v20\100]*
Data Transfer Transfer values from one place to another	TX	Fixed Container: *[TransMax v14\500]* *[TransMax v14\100]*
Alarm Delivery Configure destinations to route alarm events	AR	Fixed Container: *[AlmRoute v11]*

Description	Reference	Type
Alarm History Contains a list of the latest alarms received	AH	Fixed Container: *[AlarmHistory v11\100]*
Alarm Emailer Destination for alarms to be emailed to a recipient	AE	Variable Container: *[AlmEmail v22]*
North IP Devices Available North IP-compatible devices on the network	IP	Variable Container: [IpBus Net]
Interface Setup Set up the driver, started on interface c. This object is only available once an interface has been started. The interface number, *c*, is in the range 1..4	M*c*	Fixed Container: Refer to specific driver manual for type
Interface System Access the system connected to interface *c*. This object is only available once an interface has been started. The interface number, *c*, is in the range 1..4	S*c*	Fixed or Variable Container: Refer to specific driver manual for type

Configuration

Object Type: *[Commander v20]*

The Commander Configuration object contains the following objects:

Description	Reference	Type
Label Label used to identify this device, including scanning and within alarms.	PL	Obj\Text; Max. 30 chars; Adjustable
Local Date & Time Local date and time from Commander's real-time-clock	T	Obj\DateTime; Adjustable
Platform Information General information about Commander, including hardware switches, version information, and debug recording	PI	Fixed Container: *[Commander v20\Platform v12]*
LAN Port Set up Commander's LAN port, including IP address	N	Fixed Container: *[Commander v20\DynamicLAN v10]*
UTC & Daylight Savings Set the time and date, UTC offset, and add daylight saving periods	TZ	Fixed Container: *[Commander v20\TimeZone v10]*
Interfaces Start an interface, list installed drivers, and review interface licence information	O	Fixed Container: *[Commander v20\Interfaces v10]*
Web Server Enable the web server, edit security, and select the features available from the web site	WS	Fixed Container: *[CDM v20\C2Web v20]*
Essential Data Configure the database of values used by multiple Commander modules	UD	Fixed Container: *[CDM v20\UserData v30\Format0]* *[CDM v20\UserData v30\Format1]* *[CDM v20\UserData v30\Format2]* *[CDM v20\UserData v30\Format3]*
Time Control Set the system label and access security	CT	Fixed Container: *[CDM v20\CalTimer v20]*
ObVerse Processor *x* Edit the ObVerse for this processor. Where *x* is in the range 1..2	P*x*	Fixed Container: *[CDM v20\OBVProcess v11]*
Security Server Set the label and maximum users available	TK	Fixed Container: *[CDM v20\TokenMax v20]*
Data Transfer Set the system label, maximum transfers available, and tasks	TX	Fixed Container: *[CDM v20\TransMax v14]*
Alarm Delivery Set the system label	AR	Fixed Container: *[CDM v20\AlmRoute v11]*
Alarm History Set the system label and access security	AH	Fixed Container: *[CDM v20\AlarmHistory v11]*
Alarm Email Configure the SMTP server parameters and destination recipient address	AE	Fixed Container: *[CDM v20\AlarmEmail v22]*
North IP Devices Manage the list of connected IP devices	IP	Fixed Container: *[CDM v20\IpBus v21]*
Telnet Setup Enable the Telnet service, label, and user-password	TN	Fixed Container: *[CDM v20\Telnet v10]*

Platform Information

Object Type: *[Commander v20\Platform v12]*
Object Type: *[Commander v20\Platform v11]*

The Platform Information object provides *general information about Commander*.

Description	Reference	Type
Commander Version The Commander's main firmware version and release date	PV	Obj\Text
Hardware Version Version number of Commander's physical hardware	HV	Obj\Float
Hardware Mode State of the internal switches – DEFAULTIP and PROGRAM	M	Obj\Enum; Range 0..3 Values: 0=Run, 1=Program mode, 2=Default IP, 3=Program mode & Default IP
Battery State Health of the internal battery. Replace when low.	BS	Obj\Enum; Range 0..3 Values: 0=Ok, 1=Low, 2=Very Low, 3=Unknown
Reset Platform Set to perform a soft reset of Commander. The value indicates the seconds before restart	!rst	Obj\Num; Range 0..1000
Operating System Information on disk space available	OS	Fixed Container: *[Commander v20\Platform v12\OS]*
Last Restart Information about the last restart	LR	Fixed Container: *[Commander v20\Platform v12\Restart]*
Software Versions Software versions for each of Commander's main modules and started interfaces	V	Fixed Container: *[Commander v20\Platform v12\Versions]*
Recording Enable debug recording, and select which information is recorded	R	Fixed Container: *[Commander v20\Platform v12\Record]*

Alarm Fields

Alarms are sent by Commander when the battery state reaches a low condition.

The Platform Information module places the following information into the North-format fields:

System – 'Commander'

Point – 'Battery'

Condition – either 'Very Low', 'Low', or 'OK'

Priority – '2' for Very Low, '3' for Low and OK conditions

Date & Time – from Commander

Alarm Examples

System	Point	Condition	Priority	Date	Time
Commander	Battery	Low	3	23/03/15	13:33:59
Commander	Battery	Very Low	2	15/04/15	07:30:16
Commander	Battery	OK	3	18/04/15	10:15:24

Operating System

Object Type: *[Commander v20\Platform v12\OS]*

An Operating System object contains the following objects:

Description	Reference	Type
Disk Space Available (MB)	DA	Obj\Num; Range 0..128

Last Restart

Object Type: *[Commander v20\Platform v12\Restart]*
Object Type: *[Commander v20\Platform v11\Restart]*

A Last Restart object contains information about the last time Commander was powered-up or reset.

Description	Reference	Type
Last Start Date & Time	ST	Obj\DateTime
Reset Count Number of times Commander has been restarted	RC	Obj\Num
Reset Reason The reason for the last reset	RR	Obj\ENum; Range 0..2 Values: 0=Power failure, 1=User reset, 2=Watchdog

Software Versions

Object Type: *[Commander v20\Platform v12\Versions]*
Object Type: *[Commander v20\Platform v11\Versions]*

A Software Versions object holds the version and release date of the sub-components within Commander, and contains the following objects:

Description	Reference	Type
Web Server	WS	Fixed Container: *[Commander v20\Platform v12\Detail]*
Essential Data	UD	Fixed Container: *[Commander v20\Platform v12\Detail]*
Time Control	CT	Fixed Container: *[Commander v20\Platform v12\Detail]*
ObVerse Processor	P1	Fixed Container: *[Commander v20\Platform v12\Detail]*
Security Server	TK	Fixed Container: *[Commander v20\Platform v12\Detail]*
Data Transfer	TX	Fixed Container: *[Commander v20\Platform v12\Detail]*
Alarm Delivery	AR	Fixed Container: *[Commander v20\Platform v12\Detail]*
Alarm History	AH	Fixed Container: *[Commander v20\Platform v12\Detail]*
Alarm Email	AE	Fixed Container: *[Commander v20\Platform v12\Detail]*
North IP Devices	IP	Fixed Container: *[Commander v20\Platform v12\Detail]*
Telnet	TN	Fixed Container: *[Commander v20\Platform v12\Detail]*
Interface *x* Where *x* is in the range 1..4	M*x*	Fixed Container: *[Commander v20\Platform v12\Detail]*

Software Version Detail

Object Type: *[Commander v20\Platform v12\Detail]*
Object Type: *[Commander v20\Platform v11\Detail]*

A Software Version Detail object contains the following objects:

Description	Reference	Type
Driver Version Software component name, version and release date	DV	Obj\Text; Max chars 32
CDM Surround Version Common library version and release date	PV	Obj\Text; Max chars 32

Recording Configuration

Object Type: *[Commander v20\Platform v12\Record]*
Object Type: *[Commander v20\Platform v11\Record]*

The Recording Configuration object enables *debug recording* for Commander.

An ASCII terminal, or PC running terminal software, is needed to receive the debug information. The terminal needs to be set to 38400 baud, no parity, 8 data-bits, 1 stop-bit, and no hardware flow control.

An RS232 null-modem cable is required to connect the terminal software to Commander.

Description	Reference	Type
Record to Port COM port to send debug text to. Do not select a COM port already in use by an interface	P	Obj\ENum; Range 0..2; Adjustable Values: 0=Off, 1=COM1, 2=COM2
Record Information Record general information such as Commander starting	I	Obj\NoYes; Adjustable
Record Objects Record object request and reply messages between modules, plus any other additional debug messages from a driver	O	Obj\NoYes; Adjustable
Record COMs Record bytes sent and received on COM ports	C	Obj\NoYes; Adjustable
Record Network Record Ethernet information and IP events such as ARP, DNS, etc.	N	Obj\NoYes; Adjustable
Record Internal Comms Record fast request/reply messages between internal modules	T	Obj\NoYes; Adjustable

LAN Port Configuration

Object Type: *[Commander v20\DynamicLAN v10]*

A LAN Port Configuration object allows the *set-up of Commander's LAN port*, including selecting how the IP address is assigned.

After changing the LAN Port objects, reset Commander for the new values to take effect.

Description	Reference	Type
Label Fixed label 'LAN Port' – for compatibility with ObSys	L	Obj\Text; Max chars 20
Network Available Whether the network port has been initialised, link is up with remote end, and has IP address assigned	S	Obj\NoYes
Current IP Address IP address currently in use	CIA	Obj\IP
IP Address Assign a static IP address, or '0.0.0.0' to request a dynamic IP address from a DHCP server	IA	Obj\IP; Adjustable
Subnet Mask Subnet mask to use with a static IP address, e.g. '255.255.255.0' for class C networks	IM	Obj\IP; Adjustable
Gateway Address Gateway IP address to use in static IP mode	IG	Obj\IP; Adjustable
DNS Server Address Domain name server IP address to use in static IP mode, or if not supplied by a DHCP server	ID	Obj\IP; Adjustable
Time Server Address NTP server IP address to use in static IP mode, or if not supplied by a DHCP server. If an NTP is not available on your LAN, visit *www.pool.ntp.org* to find your nearest server	IT	Obj\IP; Adjustable
MAC Address MAC address of Commander's Ethernet port	MAC	Obj\Text; 17 chars in the format: XX:XX:XX:XX:XX:XX

UTC & Daylight Savings Configuration

Object Type: *[Commander v20\TimeZone v10]*

A UTC & Daylight Savings object configures *Commander's real-time clock and time zone*, including Coordinated Universal Time (UTC), local time offset, local time, and daylight saving offsets.

Description	Reference	Type
UTC Commander's built-in clock date & time in Universal Coordinated Time	UTC	Obj\DateTime; Adjustable
Local Time Offset from UTC Offset time in hours and minutes to apply to UTC in order to calculate local time	LTO	Obj\Text; Adjustable Value in the format: '+*hh*:*mm*' or '–*hh*:*mm*'
Daylight Saving Offset Current daylight saving offset applied to local time	DSO	Obj\Text Value in the format: '+*hh*:*mm*' or '–*hh*:*mm*'
Local Time Commander's local time based on UTC, local time offset, and daylight saving offset	LT	Obj\DateTime
Daylight Saving *x* Time Date and time to apply the daylight saving offset. Daylight saving index, *x*, is in the range 1…20	D*x*.T	Obj\DateTime; Adjustable
Daylight Saving *x* Offset Daylight saving offset to apply. Daylight saving index, *x*, is in the range 1…20	D*x*.O	Obj\Text; Adjustable; in the format +HH:MM or –HH:MM
Last Time Sync Source Method last used to update Commander's time and date. If Commander has access to an NTP time server, it will automatically correct its clock	TSS	Obj\ENum Value: 0=Never, 1=Object, 2=NTP server, 3=Other
Days Since Time Adjusted Period since Commander's date and time was last updated	TSD	Obj\Num

Interfaces Configuration

Object Type: *[Commander v20\Interfaces v10]*

The Interfaces Configuration object allows Commander to *start interfaces to other systems*, list the installed drivers, and review licencing information.

Description	Reference	Type
Interface *x* Name of driver to start at interface *x*, where *x* is in the range 1..4	O*x*	Obj\Text; Max. 16 char; Adjustable
Installed Drivers List of installed drivers	A	Fixed Container: *[Commander v20\Interfaces v10\Avail]*
Interface Licences Licencing information and upgrade	IL	Fixed Container: *[Commander v20\Interfaces v10\IntLic]*

Installed Drivers

Object Type: *[Commander v20\Interfaces v10\Avail]*

An Installed Drivers object contains the following objects:

Description	Reference	Type
Installed Driver List of drivers available. The list index, *x,* is in the range 1..250	A*x*	Obj\Text; Max chars 16; Adjustable

Interface Licences

Object Type: *[Commander v20\Interfaces v10\IntLic]*

Each Commander is supplied with a certain number of *interface licences*. It is possible to add more licences on site, although a phone call is required to check/verify settings.

Call North support on +44 (0) 1273 694422 for more help with adding more licences.

Description	Reference	Type
Serial Number Commander's unique serial number	S	Obj\Text; Max chars 16
Total Licences Total number of interface licences installed	T	Obj\Num; in the range 0..4
Available Licences Number of interface licences available. Each started interface will typically use one licence	A	Obj\Num; in the range 0..4
Upgrade Licences Key used to add more interface licences	U	Obj\Text; Max chars 32; Adjustable

Web Server Configuration

Object Type: *[CDM v20\C2Web v20]*
Object Type: *[CDM v20\C2Web v11]*

The Web Server Configuration object enables *Commander's web server* and selects what information is accessible from it.

Description	Reference	Type
Enable Web Server	E	Obj\NoYes; Adjustable
Show Essential Values Enables access to values from Essential Data	UD.EV	Obj\NoYes; Adjustable
Show Extra Values Enables access to values from Extra Data. Available in C2Web v2.0 and later	UD.XV	Obj\NoYes; Adjustable
Show Values in Sub-menu Places pages from Essential Values and Extra Values in their own sub-menus on the website. Without this option enabled, pages are displayed in the main menu	UD.PL	Obj\NoYes; Adjustable
Show Calendar Enables access to the calendar, timers and profilers from the website	C.E	Obj\NoYes; Adjustable
Show Alarms Enables access to alarm history from the website	AM.E	Obj\NoYes; Adjustable
Total Pages Served A count of the total web pages requested	PC	Obj\Num;
Template Configure options on the web template	T	Fixed Container: *[CDM v20\C2Web v20\Template]*
Home Page Configure options to customise the home page	H	Fixed Container: *[CDM v20\C2Web v20\HomePage]*
Security Configure sign-in and security options	S	Fixed Container: *[CDM v20\C2Web v20\Security]*
Debug Enable Store additional debug information in the record file. Use this option only when instructed by North Support	DE	Obj\NoYes; Adjustable

Web Server Template

Object Type: *[CDM v20\C2Web v20\Template]*
Object Type: *[CDM v20\C2Web v11\Template]*

A Web Server Template object contains the following objects to *define the page template*.

Description	Reference	Type
Show Value Reliability Enables the display of status information for values, e.g. Ok, Alarm states	SS	Obj\NoYes; Adjustable
Link x Label Label for external link *x*. The link number, *x*, is in the range 1..3	L*x*.L	Obj\Text; Max. 20 chars; Adjustable
Link x URL Web page address for external link *x*. E.g. 'www.northbt.com'. The link number, *x*, is in the range 1..3	L*x*.U	Obj\Text; Max. 125 chars; Adjustable

Web Server Home Page

Object Type: *[CDM v20\C2Web v20\HomePage]*
Object Type: *[CDM v20\C2Web v11\HomePage]*

The Web Server Home Page object contains configuration options to *enhance the home page*, including a title, optional text paragraphs, informative widgets from Essential Data, and an alarm list summary.

Description	Reference	Type
Title Headline to display on home page. If not specified, then the date is shown	T	Obj\Text; Max chars 60; Adjustable
Info Para x Set with a line of text to display as an informational paragraph on the home page. The paragraph number, *x*, is in the range 1..4	P*x*	Obj\Text; Max chars 125; Adjustable
Widget Style Select what information from Essential Data is displayed in the widget area	WS	Obj\ENum; Range 0..2; Adjustable Values: 0=Off, 1=1st object of each page, 2=Objects in page 1
Max Widgets Maximum number of widgets to display	WM	Obj\Num; Range 0..32; Adjustable
Max Alarms Maximum number of alarms to display	WA	Obj\Num; Range 0..127; Adjustable

Web Server Security

Object Type: *[CDM v20\C2Web v20\Security]*
Object Type: *[CDM v20\C2Web v11\Security]*

The Web Server Security object contains *security settings* to enable user sign-in and control access to the server.

Description	Reference	Type
Require User Sign-in Enable website authentication using the Security Server	M	Obj\NoYes; Adjustable
Allow Password Change Enables change password option when user is signed-in	UPC	Obj\NoYes; Adjustable
Security Server Object Object reference of a Security Server. Typically this is set to 'TK' for local Commander, but could be set to the object of a central server	R	Obj\Obj; Adjustable
Default Access Security Set privilege levels for when user sign-in is not enabled. Available in C2Web v2.0 and later	D	Fixed Container: *[CDM v20\C2Web v20\DefaultAccess]*

Web Server Default Access Security

Object Type: *[CDM v20\C2Web v20\DefaultAccess]*

A Web Server Default Access Security object contains the privilege levels to use when the user sign-in option is not enabled.

Description	Reference	Type
Privilege Level in Area *x* The area, x, is in the range 1…8	P*x*	Obj\Num: 0…7; Adjustable

Essential Data Configuration

Object Type: *[CDM v20\UserData v30\Format0]*
Object Type: *[CDM v20\UserData v30\Format1]*
Object Type: *[CDM v20\UserData v30\Format2]*
Object Type: *[CDM v20\UserData v30\Format3]*

The Essential Data object contains the configuration for *Commander's database of values*.

Essential Data consists of a list of configurable pages, each of which has a list of configurable objects. In total 640 database objects are available.

Description	Reference	Type
Essential Data Label	DL	Obj\Text; Max chars 20; Adjustable
Page x Object Layout Arrangement of pages and objects within the database. Choose from 40 pages of 16 objects (default), 64 pages of 10 objects, 20 pages of 32 objects, or 10 pages of 64 objects. Rescan the Commander Configuration and this object after changing the value	PO	Obj\ENum; Adjustable Values: 0=40x16, 1=64x10, 2=20x32, 3=10x64
Alarm State Overall alarm state of all pages within the database – out-of-range alarm, communications fault	S	Obj\ENum; Range 0..3 Values: 0=Ok, 1=Alarm, 2=Comms, 3=Alarm & Comms
Log Channels Free Count of available object logging channels	FL	Obj\Num; Range 0..40
Page *x* Configure database page *x*. Where *x* is in the range 1..64, depending on Page x Object Layout object (PO) above	P*x*	Fixed Container: 40 page x16 object layout *[CDM v20\UserData v30\PageDef0]* 64 page x10 object layout *[CDM v20\UserData v30\PageDef1]* 20 page x32 object layout *[CDM v20\UserData v30\PageDef2]* 10 page x64 object layout *[CDM v20\UserData v30\PageDef3]*
Task Control Allows control and monitoring of the module's operation	TI	Fixed Container: *[CDM v20\UserData v30\TaskInfo]*

Essential Data Page Configuration

Object Type: *[CDM v20\UserData v30\PageDef0]*
Object Type: *[CDM v20\UserData v30\PageDef1]*
Object Type: *[CDM v20\UserData v30\PageDef2]*
Object Type: *[CDM v20\UserData v30\PageDef3]*

The Essential Data Page object contains the configuration for a page within *Commander's database of values*. Each page has a label, access security, remote object prefix, and up to 64 objects.

The Remote Object Prefix (RP), if set, applies a root object to all remote objects within the page. It allows the page to deal with a particular container object, for example an energy meter or fan coil, without the need to enter the full object reference. A benefit of this is that the same page can be copied and used like a template for similar devices. Then by only changing the Remote Object Prefix the same information from an energy meter or fan coil at a different address can be accessed.

When the Remote Object Prefix (RP) is set, then Essential Data uses this to optimize communications in a fault condition. When Object 1 (O1) is in a communications fault alarm state, then only this object on the page performs the remote action – the other objects on the page will automatically fail. Once this first object is communicating again, then normal operation will resume.

Description	Reference	Type
Label Required to enable the page to other modules	L	Obj\Text; Max chars 20; Adjustable
Remote Object prefix Optional prefix to add in front of each database object's Remote Object	RP	Obj\Obj; Adjustable
Access Security Area and minimum privilege level required to read all objects within the page from other modules. See *Security Server*	AS	Obj\Num; Range 0, 10..87; Adjustable
Page Alarm State Overall alarm state of all objects within the page – out-of-range alarm, communications fault	S	Obj\ENum; Range 0..3 Values: 0=Ok, 1=Alarm, 2=Comms, 3=Alarm & Comms
Comms Alarm Enable/Priority Enables communications fault alarms for the page, see below	P	Obj\Num; Range 0..9; Adjustable Values: 0=no alarms are sent, 1…9=alarm priority (1 is the highest alarm priority, and 9 is the lowest)
Object *x* Configure database object *x*. Where *x* is in the range 1..64, depending on Page x Object Layout object (PO)	O*x*	Fixed Container: *[CDM v20\UserData v30\ObjDef]*

Alarm Fields

Alarms can be sent by the page to indicate the communication state of objects within the page.

The Essential Data module places the following information into the North-format fields:

System – from the Essential Data Label object (DL)

Point – from the page's Label object (L)

Condition – either 'Communications Fault' or 'Communications Ok'

Priority – set using Comms Alarm Enable/Priority object (P)

Date & Time – from Commander

Alarm Examples

System	Point	Condition	Priority	Date	Time
Essential Values	Air conditioning	Communications Fault	3	23/03/15	13:33:59
Essential Values	Air conditioning	Communications Ok	3	24/03/15	07:30:16

Essential Data Object Configuration

Object Type: *[CDM v20\UserData v30\ObjDef]*

Essential Data Object contains the configuration for an object within *Commander's database of values*.

Essential Data allows the engineer to configure a list of values that other Commander modules can then distribute – as web pages, as BACnet points, as Modbus points, on Zip displays…

Description	Reference	Type
Label Required to enable the object to other modules	L	Obj\Text; Max. 20 chars; Adjustable
Type Specifies the type of value that the database object should hold	T	Obj\ENum; Range 0..9; Adjustable Values: 0=Text, 1=NoYes, 2=OffOn, 3=Number, 4=Float, 5=Times, 6=DateTime, 7=Date, 8=ENum, 9=Profile
Adjustable Specifies whether the user can adjust the value from other modules	A	Obj\NoYes; Adjustable
Units Optional unit of measurement	U	Obj\Text; Max. 8 chars
Access Security Area and minimum privilege level required to adjust the object from other modules. See *Security Server*	AS	Obj\Num; Range 0, 10..87; Adjustable
Type ENum Alternatives When Type is 'ENum', set this to a comma-separated list of value labels. i.e. 'value-0,value-1,value-2', etc.	EA	Obj\Text; Max. 30 chars; Adjustable
Type Float Dps/Time Periods When Type is 'Float', set this to the number of decimal places to display the current value. When Type is 'Times' or 'Profile', set this to the number of on-off or time-value periods.	D	Obj\Num; Adjustable Type Float: Range 0…4 Type Times: Range 0…4 Type Profile: Range 0…8
Value High Limit Used as alarm limits, if alarms are enabled, and used as value limits when adjustments are made	VH	Obj\Float; Decimal places=4; Adjustable
Value Low Limit Used as alarm limits, if alarms are enabled, and used as value limits when adjustments are made	VL	Obj\Float; Decimal places=4; Adjustable
Current Value Database object's value	V	Obj\Text; Max chars 32; Adjustable
Value Alarm State Object's alarm state – out-of-range alarm, communications fault	S	Obj\ENum; Range 0..3 Values: 0=Ok, 1=Alarm, 2=Comms, 3=Alarm & Comms
Value Last Updated Time and date the current value was last set, either by a user or remote action	VT	Obj\DateTime
Value Alarm Enable/Priority Enables out-of-range value alarms, see below	P	Obj\Num; Range 0..9; Adjustable Values: 0=no alarms are sent, 1…9=alarm priority (1 is the highest alarm priority, and 9 is the lowest)

Description	Reference	Type
Remote Action Selects that the current value is periodically read from or written to the Remote Object. When set to 'read', and Adjustable to 'yes', then the current value will also be written to the Remote Object when adjusted by a user.	RA	Obj\ENum; Range 0..2; Adjustable Values: 0=None, 1=Read, 2=Write
Remote Object Object reference to read or write. If the Page has a remote object prefix set, then this will be inserted before the Remote Object	RO	Obj\Obj; Adjustable
Remote Rate Frequency to read or write the Current Value. If Remote Action is 'write', then the value is always written on a change-of-value. Set a rate to perform an additional background write periodically.	RR	Obj\ENum; Range 0..9; Adjustable Values: 0=ASAP/COV, 1=1 sec, 2=5 secs, 3=15 secs, 4=1 min, 5=5 mins, 6=15 mins, 7=1 hr, 8=4 hrs, 9=12 hrs
Remote Fails Count of times the remote object has continuously failed to read or write	RF	Obj\Num; Range 0..9
Data Log Enable/Rate Enables logging the current value periodically. There are a limited number of logging channels available, refer to Log Channels Free (FL)	LR	Obj\ENum; Range ; Adjustable Values: 0=Disable, 1=1 min, 2=5 mins, 3=15 mins, 4=1 hr, 5=4 hrs, 6=12 hrs, 7=24 hrs
Data Log Access historical log of the value, if enabled	LOG	Obj\Log

Alarm Fields

Alarms can be sent by an object to indicate an out-of-range value alarm state.

The Essential Data module places the following information into the North-format fields:

System – from the Essential Data Label object (DL)

Point – from page Label (L), then ' - ', and then the object Label (L)

Condition – either 'Alarm' or 'Ok'

Priority – set using Value Alarm Enable/Priority object (P)

Date & Time – from Commander

Alarm Examples

System	Point	Condition	Priority	Date	Time
Essential Values	Air conditioning - Room temp	Alarm	3	24/03/15	10:16:02
Essential Values	Air conditioning - Room temp	Ok	3	24/03/15	10:52:16

Essential Data Task Control

Object Type: *[CDM v20\UserData v30\TaskInfo]*

The Essential Data Task Control object is used to control and monitor the operation of *Commander's database of values*.

Description	Reference	Type
Enable Option to enable or disable the operation of Essential Data, or to only allow read remote actions. 'Enable reading' will stop all writing to the remote objects and only permit remote actions set to read.	E	Obj\ENum: Range 0..2 ; Adjustable Values: 0=Disable, 1=Enable all, 2=Enable reading
Value being read Page and object index that Essential Data is currently processing with a remote action of read	RT	Obj\Text
Value being written Page and object index that Essential Data is currently processing with a remote action of write	WT	Obj\Text

Time Control Configuration

Object Type: *[CDM v20\CalTimer v20]*

A Time Control Configuration object contains a label and access security objects for *Commander's calendar, timers and profilers*.

Description	Reference	Type
Time Control Label	DL	Obj\Text; Max chars 20; Adjustable
Security: Read Access Area and minimum privilege level required to read the calendar and timers. See *Security Server*	AS	Obj\Num; Range 0, 10..87; Adjustable
Security: Edit Timer Area and minimum privilege level required to adjust a timer or profiler	AS.ET	Obj\Num; Range 0, 10..87; Adjustable
Security: Edit Calendar Area and minimum privilege level required to adjust the calendar	AS.EC	Obj\Num; Range 0, 10..87; Adjustable

ObVerse Processor Configuration

Object Type: *[CDM v20\OBVProcess v11]*
Object Type: *[CDM v20\OBVProcess v10]*

An ObVerse Processor Configuration contains an object to engineer the strategy within the *ObVerse Processor*. Use the North ObvEditor application to create and edit a cause-and-effect strategy.

Description	Reference	Type
ObVerse Edits ObVerse strategy within the processor. Connect using ObvEditor software	P	Obj\EProcess; Adjustable
Idles per second Number of times the complete ObVerse strategy is processed per second. Not available in driver version 1.0.	IR	Obj\Num
Object Modules Monitor the operation of object module types within the processor. These include Object-Read and Object-Write modules. Not available in driver version 1.0.	T	Fixed container: *[OSM v20\OBVProcess v11\ObjMods]*

Alarm Fields

Alarms can be sent by any alarm module within the ObVerse strategy. Refer to *ObVerse Manual: Standard Processor* for more information on the alarm module.

The ObVerse Processor places the following information into the North-format fields:

System – from label of the process

Point – from alarm module's point object

Condition – from alarm module's condition field

Priority – from alarm module's priority field

Date & Time – from Commander

Alarm Examples

System	Point	Condition	Priority	Date	Time
Main Plant	Temperature	Too High	3	23/03/16	13:33:59
Main Plant	Temperature	OK	3	23/03/16	15:21:16

ObVerse Object Modules

Object Type: *[CDM v20\OBVProcess v11\ObjMods]*

ObVerse Object Modules contains objects to enable and monitor the operation of object modules types within the processor. These modules perform a remote object operation and include the Object-Read, Object-Write, and Alarm modules.

Description	Reference	Type
Operation Temporarily disable the remote object operation of all object module types in the processor	D	Obj\ENum; Adjustable Values: 0=Enabled, 1=Disabled
Active Module Indicates which object module the processor is performing the remote object operation	M	Obj\Text Format: *item=module-type object-reference* Examples: V6=!Obv\ObjRead S1.M3.UI1.V V21=!Obv\ObjWrite UD.P2.O6 V36=!Obv\Alarm ALARM
Last Failed Module Indicates the last object module the processor failed to perform the remote object operation	FM	Obj\Text Format: *item=module-type object-reference*
Last Failed Time Indicates the date and time the processor failed to perform the remote object operation	FDT	Obj\DateTime

Security Server Configuration

Object Type: *[CDM v20\TokenMax v20]*

A Security Server Configuration contains objects to set a label, and the number of users stored by *Commander's Security Server*.

Description	Reference	Type
Security Server Label	DL	Obj\Text; Max chars 20; Adjustable
Maximum Users Sets the maximum number of users available within the Security Server. Set this to 100 or 400 only.	MU	Obj\Num; Range 100..400; Adjustable

Data Transfer Configuration

Object Type: *[CDM v20\TransMax v14]*

A Data Transfer Configuration object enables the *transferring of values by Commander*, and contains objects to set the maximum transfers available and to monitor its operation.

Description	Reference	Type
Transfer Enable Option to enable or disable the operation of transfers, or to only allow reading. 'Enable reading' will stop all writing to destination objects and only read source objects.	E	Obj\ENum; Range 0..2; Adjustable Values: 0=Disable, 1=Enable All, 2=Enable Reading
Maximum Transfers Sets the maximum number of transfers available. Set this to 100 or 500 only.	MT	Obj\Num; Range 100..500; Adjustable
Maximum Tasks Sets the number of simultaneous transfer tasks performed. Typically set this to '1'.	XT.M	Obj\Num; Range 1..4; Adjustable
Task *x* Allows monitoring of the module's four transfer tasks. The task number, *x*, is in the range 1..4	XT*x*	Fixed Container: *[CDM v20\TransMax v14\Task]*

Data Transfer Task

Object Type: *[CDM v20\TransMax v14\Task]*

A Data Transfer Task object contains information about a transfer task.

Description	Reference	Type
Process Current action the transfer task is performing	P	Obj\ENum; Range 0..2 Values: 0= Idle, 1=Reading, 2=Writing
Transfer Number Transfer the task is currently performing an action on	X	Obj\Num; Range 0..500

Alarm Delivery Configuration

Object Type: *[CDM v20\AlmRoute v11]*

An Alarm Delivery Configuration object contains a label for the *alarm delivery system*.

Description	Reference	Type
Alarm Delivery Label	DL	Obj\Text; Max chars 20; Adjustable

Alarm History Configuration

Object Type: *[CDM v20\AlarmHistory v11]*

An Alarm History Configuration object contains a system label and access security objects for *Commander's alarm history list*.

Description	Reference	Type
Alarm History Label	DL	Obj\Text; Max chars 20; Adjustable
Read Access Security Area and minimum privilege level required to read the alarms. See *Security Server*	AS	Obj\Num; Range 0, 10..87; Adjustable
Delete Access Security Area and minimum privilege level required to delete the alarms	AS.D	Obj\Num; Range 0, 10..87; Adjustable

Alarm Email Configuration

Object Type: *[CDM v20\AlmEmail v22]*

An Alarm Email Configuration contains objects that set up how *Commander's alarm emailer* connects to an SMTP relay server, and set up the six destinations to provide email addresses for recipients.

Description	Reference	Type
Alarm Email label	DL	Obj\Text; Max chars 20; Adjustable
Email server IP/FQDN IP address or host name of SMTP relay server, e.g. '102.13.14.15' or 'mail.domain.com'. If a host name is used, ensure a DNS server address is also configured in the LAN port settings	IA	Obj\Text; Max chars 127; Adjustable
Email server port TCP/IP port number of SMTP service, default 25. If a firewall is in use, ensure it permits outbound traffic on this port number. SMTP servers supporting encryption are not supported, so check compatibility of servers using ports 465 or 587.	PN	Obj\Num; Adjustable
Email server reachable Indicates that the email server IP and port have been configured correctly	DS	Obj\NoYes
Server timeout (s) Maximum time to wait for a response from the SMTP server	TO	Obj\Num; Range 5..120; Adjustable
Authentication name User name to use when sign-in is required. A sign-in may be required on some public SMTP relay servers, such as from BT Internet. The authentication supported by the Alarm Email module is AUTH LOGIN.	A.ID	Obj\Text; Max chars 40; Adjustable
Authentication password Password to use when sign-in is required	A.PW	Obj\Text; Max chars 40; Adjustable If a password is set, it will read as '****'
From name Label describing the email author, Commander. If not specified, Alarm Email Label (DL) will be used	FT	Obj\Text; Max chars 40; Adjustable
From email address Email address to use for the email author. Usually required by the SMTP server	DFA	Obj\Text; Max chars 40; Adjustable
Client Domain Name Optional hostname used to identify Commander when connecting to the SMTP server. E.g. 'cmdr.domain.com' If not specified, then Commander's IP address will be used	FQDN	Obj\Text; Max chars 127; Adjustable
Web page address Optional url to include in the email message. Configure the link with the url of Commander's website, e.g. 'cmdr.domain.com'	WV	Obj\Text; Max chars 120; Adjustable
Sent email count Total number of emails send since Commander last started	SC	Obj\Num

Description	Reference	Type
Failed email count Count of times email messages have continuously failed to send	FC	Obj\Num
Sent test to destination Triggers a test email message to the destination number specified	TST	Obj\Num; Range 0..6; Adjustable
Debug enable Enables the recording of communication with the SMTP server. Use this when instructed by North support to assist with fault finding sending emails to an SMTP server. See *Debug recording*	DE	Obj\NoYes; Adjustable
Destination *x* Configure a destination with filtering options and email address. The destination, *x*, is in the range 1..6	D*x*	Fixed Container: *[CDM v20\AlmEmail v22\Dest]*

Alarm Email Destination Configuration

Object Type: *[CDM v20\AlmEmail v22\Dest]*

An Alarm Email Destination Configuration contains up to five addresses of where an alarm message is sent by *Commander's alarm emailer*. An optional comparison method can be specified to filter alarms received from the system's ALARM object.

Description	Reference	Type
Label Description of destination group	L	Obj\Text; Max chars 20; Adjustable
Send as text only Send emails as text-only format rather than HTML	EF	Obj\NoYes; Adjustable
Comparison Method Optional filter, so that only alarms containing certain text are emailed	C	Obj\ENum; Range ; Adjustable Values: 0=Begins with, 1=Contains, 2=Does not begin, 3=Does not contain, 4=Always send
Comparison String Optional text used with the comparison method to filter incoming alarms. String comparisons are case insensitive	S	Obj\Text; Max chars 20; Adjustable
Address *x* Configure an email recipient. The address, *x*, is in the range 1..5	TO*x*	Fixed Container: *[CDM v20\AlmEmail v22\Addr]*

Email Address Setup

Object Type: *[CDM v20\AlmEmail v22\Addr]*

An Email Address Setup object contains the following objects.

When the Fails (F) count reaches 3 for the address, then Enable (E) is set to 'No' and a destination failed alarm is sent.

Description	Reference	Type
Email address Single email address to email alarms, e.g. 'name@domain.com'	A	Obj\Text; Max chars 63; Adjustable
Enable Enables emailing alarms to the address	E	Obj\NoYes; Adjustable
Fails Count of times email messages have continuously failed to send to this address	F	Obj\Num; Range 0..9; Adjustable

Alarm Fields

Alarms are sent by an address to indicate a fail state.

The Alarm Email module places the following information into the North-format fields:

System – from the Alarm Email Label object (DL)

Point – from the email destination's Label object (L), or 'Destination *x*' if no label is set, then ' - ', then the Email address (A)

Condition – set to 'Destination Failed'

Priority – set to '3'

Date & Time – from Commander

Alarm Examples

System	Point	Condition	Priority	Date	Time
Alarm Emailer	Destination 1 - name@domain.com	Destination Failed	3	23/03/15	13:33:59

North IP Devices Configuration

Object Type: *[CDM v20\IPBus v21]*

A North IP Devices Configuration object manages a list of other Commanders, ObSys PCs, etc. that are available on the LAN or WAN. Refer to *Communicating with other North IP Devices* for an introduction on this module.

North IP devices on the LAN can be discovered automatically. By default, new devices on the local network are found when the North IP Devices system object is scanned from engineering software. In addition to this, the list can be scanned immediately using the Scan device list object (AF).

Devices that are outside the local network, or use an encryption key, must be added to the device list manually.

Description	Reference	Type
Scan device list Discover North IP devices on the LAN. Select 'scan now' to perform an immediate scan for new devices. Use 'wipe then scan', to erase the list and then scan for devices. After performing an action, refresh the object view.	AF	Obj\ENum; Range 0..2; Adjustable Values: 0=Auto, 1=Scan now, 2=Wipe then scan
Local encryption key If specified, all other devices that communicate with this Commander must be set to use this encryption key. Once set, reads as '****'	K	Obj\Text; Max chars 8; Adjustable
Device *x* North IP device *x* configuration, where *x* is in the range 1..200	A*x*	Fixed Container: *[CDM v20\IPBus v21\Alias]*

North IP Device Configuration

Object Type: *[CDM v20\IPBus v21\Alias]*

A North IP Device Configuration object contains an object reference, IP address, and encryption key for a Commander, ObSys PC, etc.

The module can also periodically check communications with the other device, and generate an alarm message when communications fail.

Description	Reference	Type
Reference A short object reference for the device	N	Obj\Text; Max chars 5; Adjustable
IP address The IP address of the device	A	Obj\IP; Adjustable
Encryption key If the device has IP encryption enabled (with the Local Encryption Key object), this key must match that in the device. Always reads as '****'	K	Obj\Text; Max chars 8
Comms Check Rate If required, Commander will periodically check it can request objects from the device, and will send an alarm message if comms cannot be established	R	Obj\ENum; Range ; Adjustable Values: 0=Off, 1=1m, 2=5m, 3=15m, 4=1h, 5=4h
Comms Fails Count of consecutive fails, or 0 is comms were successful	F	Obj\Num; Range 0..9

Alarm Fields

Alarms can be sent by the module to indicate a communications failure when value of Comms Fail reaches 3.

The North IP Devices module places the following information into the North-format fields:

System – 'North IP Devices'

Point – from Reference (N), then ' at ', then the IP Address (A)

Condition – either 'Comms Fail' or 'Comms Ok'

Priority – '2'

Date & Time – from Commander

Alarm Examples

System	Point	Condition	Priority	Date	Time
North IP Devices	A1 at 192.168.192.167	Comms Fail	2	24/03/15	10:16:02
North IP Devices	A1 at 192.168.192.167	Comms Ok	2	24/03/15	10:52:16

Telnet Setup

Object Type: *[CDM v20\Telnet v10]*

A Telnet Setup object controls access to Commander from Telnet clients elsewhere on the LAN.

When enabled, the session label is returned when a Telnet client opens the session. The client is then prompted for a user. If successfully authenticated, the session then asks which service the telnet client wants.

For information on establishing a Telnet session and the commands supported, refer to the *Telnet* section.

Description	Reference	Type
Session Label Provided when a Telnet session is opened	DL	Obj\Text; Max chars 20; Adjustable
Telnet Enabled Enables the Telnet service	TE	Obj\NoYes; Adjustable
User/Password User name required to access the Telnet service	PSW	Obj\Text; Max chars 7; Adjustable
Connections Number of Telnet sessions currently established	CC	Obj\Num: 0..2

Essential Values

Object Type: *[UserData\PageList]*

The Essential Values object contains pages of values from *Commander's database of values*.

Configure the database using the *Essential Data Configuration* object.

Description	Reference	Type
Page Label Page of values with label *Page Label.* The page number, *x*, is in the range 1..64	P*x*	Variable Container: *[UserData\Page]*
Alarm State Overall alarm state of all pages within the database – out-of-range alarm, communications fault	S	Obj\ENum; Range 0..3 Values: 0=Ok, 1=Alarm, 2=Comms, 3=Alarm & Comms

Essential Values Page

Object Type: *[UserData\Page]*

An Essential Values Page object contains a value for each of the objects configured in the database.

Description	Reference	Type
Label Object's value with label *Label*. The object number, *x*, is in the range 1..64	O*x*	Type as configured for the object in the database, could be: Obj\Text Obj\NoYes Obj\OffOn Obj\Num Obj\Float Obj\Times Obj\DateTime Obj\Date Obj\ENum Obj\Profile If the object has been set to Adjustable, then the value will be adjustable here.
Alarm State Alarm state for all objects within the page – out-of-range alarm, communications fault	S	Obj\ENum; Range 0..3 Values: 0=Ok, 1=Alarm, 2=Comms, 3=Alarm & Comms

Time Control

Object Type: *[CalTimer v20]*

A Time Control object is used to provide time-based control. The control can be simple timer on-off control, for things such as lighting. The control can also be profile-based control, for things such as temperature set points.

For an overview, see the section on *Controlling using Time and Date*.

Description	Reference	Type
Calendar Set day-types for standard days of the week and exception days	C	Fixed Container: *[CalTimer v20\Calendar]*
Timer *x* Set a time-list for each day-type. Timer number, *x*, is in the range 1..20	T*x*	Fixed Container: *[CalTimer v20\Timer]*
Profiler *x* Set a profile for each day-type. Profiler number, *x*, is in the range 1..20	P*x*	Fixed Container: *[CalTimer v20\Profiler]*

Calendar

Object Type: *[CalTimer v20\Calendar]*

A Calendar object is used to determine today's day-type. This is calculated from a set of day-types based on the day-of-week, which can be overridden by a list of exception dates. Each day-type can be assigned a label. If necessary, rather than calculate today's day-type, a Calendar can read a day-type calculated by a Calendar in a different device.

Description	Reference	Type
Source Object Optional current day-type object of the calendar in a different device, e.g. 'IP.A1.CT.C.T'	SO	Obj\Obj; Adjustable
Source Read Rate The rate to read the calendar source object in a different device	SR	Obj\ENum; Range 0..9; Adjustable Values: 0=ASAP, 1=1s, 2=5s, 3=15s, 4=1m, 5=5m, 6=15m, 7=1h, 8=4h, 9=12h
Source Fail Count Count of consecutive times the source object has failed to read successfully	SF	Obj\Num; Range 0..9
Day-type *x* Label Label for day-type *x*, where *x* is in the range 0..9. The label for day-type 0 is fixed to 'Off'.	T*x*.L	Obj\Text; Max chars 20; Adjustable
Current Day-type Calculated day-type for today	T	Obj\Num; Range 0..9
Current Day-type label Calculated day-type for today	T.L	Obj\Text; Max chars 20
Monday Day-type Set to the day-type required for each standard Monday	D1.T	Obj\Num; Range 0..9; Adjustable
Tuesday Day-type Set to the day-type required for each standard Tuesday	D2.T	Obj\Num; Range 0..9; Adjustable
Wednesday Day-type Set to the day-type required for each standard Wednesday	D3.T	Obj\Num; Range 0..9; Adjustable
Thursday Day-type Set to the day-type required for each standard Thursday	D4.T	Obj\Num; Range 0..9; Adjustable
Friday Day-type Set to the day-type required for each standard Friday	D5.T	Obj\Num; Range 0..9; Adjustable
Saturday Day-type Set to the day-type required for each standard Saturday	D6.T	Obj\Num; Range 0..9; Adjustable
Sunday Day-type Set to the day-type required for each standard Sunday	D7.T	Obj\Num; Range 0..9; Adjustable
Exceptions Used Count of exceptions currently in use	EC	Obj\Num; Range 0..40
Exception Date *x* List of non-standard exception days. Where *x* is in the range 1..40	E*x*	Fixed Container: *[CalTimer v20\Calendar\Except]*
***dd/mm/yy* day-type** Alternative object to set the exception day-type for a specific date, *dd/mm/yy*	ED*dd.mm.yy*	Obj\Num; Range 0..9; Adjustable

Calendar Exception Date

Object Type: *[CalTimer v20\Calendar\Except]*

A Calendar Exception Date contains the date and day-type to use on a specific exception day.

Description	Reference	Type
Date The date to apply the exception	D	Obj\Date; Adjustable
Day-type Set to the day-type to use on the exception date	T	Obj\Num; Range 0..9

Timer

Object Type: *[CalTimer v20\Timer]*

A Timer object uses today's day-type, from the Calendar, to select one of several on-off times to be used today. It uses the current time to determine from those on-off times to determine whether it should be on or off at this time. It can calculate a profile, based on simple on and off values.

A Timer object contains the following objects:

Description	Reference	Type
Label	L	Obj\Text; Max chars 20; Adjustable
Day-type *x* Times Times for day-type *x*, where *x* is in the range 1..9	T*x*	Obj\Times; Max periods 5; Adjustable
State Current state of timer, based on today's times, and the current time	S	Obj\OffOn
Destination Object Optional object to write State to when it changes	DO	Obj\Obj; Adjustable
Destination Fails Count of consecutive fails when writing to destination object, or 0 if successfully written	DF	Obj\Num; Range 0..9
Today's Times A copy of the times for today's day-type. Adjusting this object will adjust the times for today's day-type	TT	Obj\Times; Adjustable; Max periods 5
Profile Off Value Used for converting Today's Times to Today's Profile, the value in the profile when the time is off	V0	Obj\Float; Adjustable; Range 0.0..100.0
Profile On Value Used for converting Today's Times to Today's Profile, the value in the profile when the time is on	V1	Obj\Float; Adjustable; Range 0.0..100.0
Today's Profile A calculated profile based on Today's Times	TP	Obj\Profile; Max points 8

Profiler

Object Type: [CalTimer v20\Profiler]

A Profiler uses todays's day-type, from a Calendar, to select one of several profiles to be used today. It then uses the current time to determine the value at this moment in time. It can also calculate a set of on-off times, based on a switch level.

A Profiler object contains the following objects:

Description	Reference	Type
Label	L	Obj\Text; Max chars 20; Adjustable
Day-type *x* Profile Profile for day-type *x*, where *x* is in the range 1..9	P*x*	Obj\Profile; Max points 8; Adjustable
Value Current value of the profiler, based on today's profile and current time	V	Obj\Float; Range 0.00..100.0
Destination Object Optional object to write State to when it changes	DO	Obj\Obj; Adjustable
Destination Fails Count of consecutive fails when writing to destination object, or 0 if successfully written	DF	Obj\Num; Range 0..9
Today's Profile A copy of the profile for today's day-type. Adjusting this object will adjust the profile for today's day-type	TP	Obj\Profile; Adjustable; Max points 8
Time Switch Value Used for converting Today's Profile to Today's Times. The threshold value in today's profile below which the time is off, and above which the time is on	TV	Obj\Float; Adjustable; Range 0.00..100.0
Today's Times Calculated on-off times based on Today's Profile	TT	Obj\Times; Max period 5

ObVerse

Object Type: *[ObVerse\Process]*

The ObVerse object contains a list of the public properties available within the ObVerse processor.

The public properties are defined by the engineer of the ObVerse, so cannot be listed here. Instead, we list an object for each property type available.

Objects may be adjustable depending on the ObVerse. If the property is linked to the output of a module, then the module controls the value of the property, and so the property is not adjustable here. If the property has no module output linked to it, then the property is adjustable here.

For more information, refer to the *ObVerse Manual – Standard Processor* document.

Description	Reference	Type
ENum Label ENum property, holding an enumerated value, with reference *r*	*r*	Obj\ENum Text alternatives for each number are defined in the property
Float Label Float property, holding a floating-point value, with reference *r*	*r*	Obj\Float Value high and low limits, and decimal places are defined in the property
NoYes Label NoYes property, holding a binary state, with reference *r*	*r*	Obj\NoYes
Num Label Num property, holding an integer value, with reference *r*	*r*	Obj\Num: -2147483648… 2147483647 Value high, and low limits are defined in the property
Obj Label Obj property, holding an object reference, with reference *r*	*r*	Obj\Obj
OffOn Label OffOn property, holding a binary state, with reference *r*	*r*	Obj\OffOn
Text Label Text property, holding a text string value, with reference *r*	*r*	Obj\Text: max 31 chars. Maximum length is defined in the property

Security Server

Object Type: *[TokenMax v20\100]*
Object Type: *[TokenMax v20\400]*

The Security Server object provides user authentication to Commander and connected devices. For an overview, refer to the *Security Server* section.

Use the Editor Sign-in object to sign-in before configuring users and groups in the server.

Set the maximum users available in the server using the *Security Server Configuration* object.

Description	Reference	Type
Editor Access Allowed Indicates if the editor has signed-in	MP.S	Obj\NoYes
Editor Sign-in Write the password to sign-in, or any other value to sign-out. A password recovery key is returned when reading this object	MP.L	Obj\Text; Adjustable
Change Editor password Once signed-in, set a new editor password	ML.P	Obj\Text; Max 20 chars; Adjustable
Next available user Indicates the next unused user record available	FT	Obj\Num; Range 0..400
Area x Label Label of Security Area x, where x is in the range 1..8	Ax	Obj\Text; Max chars 20; Adjustable
Privilege Level x Label Label of Security Privilege Level x, where x is in the range 0..7	Lx	Obj\Text; Max Chars 20; Adjustable
Group x Set privilege levels for a group of users. The group number, *x*, is in the range 1..31	G*x*	Fixed Container: *[TokenMax v20\Group]*
User *x* Configure a user's authentication credentials. The user number, *y*, is in the range 1..400, depending on the maximum users available set in the configuration object	U*y*	Fixed Container: *[TokenMax v20\User]*

Security Server Group

Object Type: *[TokenMax v20\Group]*

A Security Server Group object contains privilege levels for a *group of users*.

Description	Reference	Type
Name Group name or description	N	Obj\Text; Max chars 20; Adjustable
Enabled Enable or disable all users that are a member of this group	E	Obj\NoYes; Adjustable
Privilege Level in Area *x* Area *x* privilege level, where *x* is in the range 1..8	P*x*	Obj\Num; Range 0..7; Adjustable

Security Server User

Object Type: *[TokenMax v20\User]*

A Security Server User object contains *user information* and credentials to provide authentication.

Description	Reference	Type
Name User name or description	N	Obj\Text; Max chars 20; Adjustable
Enabled If set to yes, the user is enabled if all groups that they belong to are also enabled	E	Obj\NoYes; Adjustable
User ID/Card Sign-in user name or access card number	ID	Obj\Text; Max Chars 20; Adjustable
Password This object always has a value of '****' when read	PW	Obj\Text; Max chars 20; Adjustable
Token This is a combination of the User ID and coded password	T	Obj\Text; Max chars 30
Group x Group x membership information, where x is in the range 1..3 When a user is a member of a group, then the group's privilege levels are combined with the user's own.	Gx	Obj\Num; Range 0..31; Adjustable
Privilege Level in Area *x* User's privilege level in Area *x*, where *x* is in the range 1..8	P*x*	Obj\Num; Range 0..7; Adjustable
Valid Start Date Optionally enables a temporary user between the start and end dates specified. For permanent users, set the start and end dates to the same value	SD	Obj\Date; Adjustable
Valid End Date Optionally enables a temporary user between the start and end dates specified. For permanent users, set the start and end dates to the same value	ED	Obj\Date; Adjustable
Last Validation Date Date the user was last validated successfully	LD	Obj\Date

Data Transfer

Object Type: *[TransMax v14\100]*
Object Type: *[TransMax v14\500]*

A Data Transfer object contains up to 500 transfers, where each transfer reads from one place and then writes to another. For an overview, see the section on *Transferring Values*.

Set the maximum transfers available using the *Data Transfer Configuration* object.

Description	Reference	Type
Transfer *x* Configure a transfer's information. The transfer number, *x*, is in the range 1..500, depending on the maximum transfers available set in the configuration object	T*x*	Fixed Container: *[TransMax v14\Transfer]*

Transfer

Object Type: *[TransMax v14\Transfer]*

A Transfer object contains the following objects:

Description	Reference	Type
Source Object Object reference to read	SO	Obj\Obj; Adjustable
Source Read Rate Frequency to read the value. See *Reading the Value*	SR	Obj\ENum; Range 0..9; Adjustable Values: 0=ASAP, 1=Never, (2 and 3 not allowed,) 4=1m, 5=5m, 6=15m, 7=1h, 8=4h, 9=12h
Source Read Fails Count of times the remote object has continuously failed to read	SF	Obj\Num; Range 0..9
Value Transfer's current value	V	Obj\Text; Max chars 32; Adjustable
Dest Write Object Object reference to write	DO	Obj\Obj; Adjustable
Dest Write Rate Frequency to write the value. The value is always written on a change-of-value. Set a rate to perform an additional background write periodically. See *Writing the Value*	DR	Obj\ENum; Range 0..10; Adjustable Values: 0=COV, 1=Never, (2 and 3 not allowed,) 4=1m, 5=5m, 6=15m, 7=1h, 8=4h, 9=12h, 10=Every 00:00
Dest Write Fails Count of times the remote object has continuously failed to write	DF	Obj\Num; Range 0..9

Alarm Delivery

Object Type: *[AlmRoute v11]*

The Alarm Delivery object contains 16 destinations to route North-format alarms received.

All alarms arriving, or being produced by Commander are passed to the ALARM object for distribution. For an overview, see the section on *Alarm Basics*.

Description	Reference	Type
Route New Alarm	ALARM	Obj\Alarm; Adjustable
Destination *x* Destination *x* information, where x is in the range 1..16	D*x*	Fixed Container: *[AlmRoute v11\Dest]*

Alarm Delivery Destination

Object Type: *[AlmRoute v11\Dest]*

An Alarm Delivery Destination object contains the following objects:

Description	Reference	Type
Label Description of this destination	L	Obj\Text; Max 20 chars; Adjustable
Enable If set to yes, alarms can be delivered to this destination	E	Obj\NoYes; Adjustable
Destination Object Object reference to route alarms. Typically this object reference ends '.ALARM'	R	Obj\Obj; Adjustable
Add Device Label If set to yes, prefixes the alarm system field with Commander's label	CL	Obj\NoYes; Adjustable
Fail State Indicates the Destination Object is failing to accept alarms	F	Obj\NoYes
High Priority Only alarms with a priority number equal or greater than this value are delivered. To disable priority filtering, set the high and low priority to the same value. See *Filtering Delivery*	HP	Obj\Num; Range 0..9; Adjustable
Low Priority Only alarms with a priority number equal or less than this value are delivered. To disable priority filtering, set the high and low priority to the same value.	LP	Obj\Num; Range 0..9; Adjustable
Comparison Method Optional text filter. Specify the method to use along with up to three comparison strings. See *Filtering Delivery*	C	Obj\ENum; Range 0..; Adjustable Values: 0=Begins with, 1=Contains, 2=Not begin with, 3=Not contain
Comparison String *x* Text string to use with the comparison method. The string number, *x*, is in the range 1..3	S*x*	Obj\Text; Max 20 chars; Adjustable
Any Group Join this destination to the any group. When one destination accepts the alarm, it will not be sent to any other destinations in the group. See *Delivery to a Group*	A	Obj\NoYes; Adjustable

Alarm Fields

Alarms are sent by the delivery destination to indicate a fail state.

The Alarm Delivery module places the following information into the North-format fields:

System – from the Alarm Delivery Label object (O.AR.DL)

Point – from the delivery destination's Label object (L), or 'Destination *x*' if no label is set

Condition – either 'Comms Fail' or 'Comms Ok'

Priority – set to '1'

Date & Time – from Commander

Alarm Examples

System	Point	Condition	Priority	Date	Time
Alarm Delivery	Destination 1	Comms Fail	1	23/03/15	13:33:59
Alarm Delivery	Destination 1	Comms Ok	1	24/03/15	07:30:16

Alarm History

Object Type: *[AlarmHistory v11\100]*

An Alarm History object contains a *list of the last 100 alarms* received.

Deliver new alarms, using *Alarm Delivery*, to the ALARM object.

Description	Reference	Type
Alarm count	C	Obj\Num; Range 0..100; Adjustable
Store new alarm	ALARM	Obj\Alarm; Adjustable
Alarm *x* List of alarms, where Alarm 1 is the last alarm received. The alarm number, *x*, is in the range 1..100	H*x*	Obj\Alarm

Alarm Emailer

Object Type: *[AlmEmail v22]*

An Alarm Emailer contains six destinations to send alarms. The email destinations are set in the *Alarm Email Configuration* object.

Description	Reference	Type
Destination Label The label of the destination, where *x* is in the range 1..6, depending whether each has been set up	D*x*	Fixed Container: *[AlmEmail v22\Dest]*

Alarm Emailer Destination

Object Type: *[AlmEmail v22\Dest]*

An Alarm Emailer Destination object contains a queue of alarms to send via email to a destination.

Deliver new alarms, using *Alarm Delivery*, to the ALARM object.

Description	Reference	Type
Deliver New Alarm	ALARM	Obj\Alarm; Adjustable
Alarms in Queue	C	Obj\Num; Range 0..31

Appendix A - Ethernet/IP Protocols

The RJ45 Ethernet port on Commander provides an IEEE 802.3 compliant 10Base-T full-duplex or half-duplex link, with auto-negotiation. The link supports auto MDI-X. Commander has an IEEE 802.3 compliant Medium Access Controller (MAC).

For information on a particular driver's IP port usage, refer to the driver manual.

Address Resolution Protocol (ARP)

Commander supports ARP to determine IP-to-Ethernet mapping.

Table Size: 32. Request Timeout: 100ms. Entry Timeout: 1200s

Internet Control Message Protocol (ICMP)

Commander supports ICMP, and responds to the PING request.

Dynamic Host Control Protocol (DHCP)

The DHCP client, if enabled by setting the Fixed IP address to '0.0.0.0', requests IP information from a DHCP server.

Ports used: UDP 67 (remote), UDP 68 (local).
Requests: IP address, network mask, gateway address, DNS server address, NTP server address

Domain Name System Protocol (DNS)

If needed, the DNS client resolves fully qualified domain names by requesting from a DNS server.

Ports used: UDP 53 (remote), UDP ephemeral (local)

Network Time Protocol (NTP)

If Commander has an NTP server address, it will try to synchronise its real-time clock.

Ports: UDP 123 (local and remote)

Telnet

If enabled, Commander supports an incoming Telnet session.

Ports used: TCP 23 (local)

Simple Mail Transfer Protocol (SMTP)

Once Alarm Emailer is configured, this SMTP client sends emails to the specified server.

Ports used: TCP 25 (configurable, remote), TCP ephemeral (local)

Hypertext Transfer Protocol (HTTP)

The Web Server, if enabled, serves simple HTML pages when requested.

Ports used: TCP 80 (local)

North/IP

The North/IP Protocol allows North products that support IP networking to communicate.

Ports used: UDP 37926

Trivial File Transfer Protocol (TFTP)

The TFTP protocol allows new CDMs and basecode to be downloaded into Commander. It is enabled only when the PROGRAM switch is on.

Ports used: UDP 69 (local)

Appendix B – Pre-loaded Drivers

Commander contains several pre-installed drivers, and you can install others as necessary. Each driver comes in a CDM file, designed to work in one of the memory banks within Commander.

Driver Name	Use	Interface Licences
BACnetIP	Interface with BACnet/ IP devices	0
Compass	Connect to North's Compass System	0
GSMSMS	SMS gateway	0
JSONData	Provides a web-based API to Commander's Essential Data	0
Modbus	Connect to Modbus over serial devices	0
ModbusSlave	Provide values to a Modbus over serial front-end	0
ModbusTCP	Interface with Modbus over IP devices	0
SG	Provides a simple text API	0
SNMPTrap	Sends SNMP traps to a receiver	0
ZipMaster	Connect to North's Zip System	0
Advanced4000	Connect to an Advanced Electronics fire system	1
APC	Connects to an APC UPS	1
Carel	Connect to a Carel air-conditioning system	1
Daikin	Connect to a Daikin air-conditioning system	1
DALI	Connect to a DALI compatible lighting system	1
DataSync	Provides a simple text-based API to Commander's Essential Data	1
ExtraData	Extends the Essential Data capabilities of Commander	1
Galaxy	Connect to Galaxy security system	1
GeistPDU	Connects to Geist PDUs	1
Helvar	Connect to a Helvar lighting system	1
JSONNotify	Provides a web-based method to push data from Commander Essential Data	1
Kentec	Connect to a Kentec Syncro fire system	1
KNXIP	Interface with equipment supporting the KNX standard	1
LegrandMyHome	Connect to Legrand and BTicino My Home range	1
LonSLTA	Connect to Echelon LonWorks devices	1
LutronQS	Connect to Lutron HomeWorks QS lighting	1
MBus	Connect to a network of energy meters supporting the M-Bus standard	1
MitsubishiG50	Connect to Mitsubishi air-conditioning system	1
Morley	Connect to a Morley-IAS ZX fire system	1
MQTT	Provides a method to push data from Commander Essential Data	1
PanasonicVRF	Connect to Panasonic or Sanyo air-conditioning	1
PhilipsLM	Connects to Philips ECS LightMaster lighting	1
PowerOne	Connects to Power-One DC power system	1
Protec	Connect to a Protec fire detection system	1
TrendIQ	Connect to a Trend IQ series BMS	1

Appendix C - PROGRAM switch

When the internal switch labelled PROGRAM is set ON, and Commander re-powered, the Commander enters ‘program’ mode, where the following happens:

The TFTP service is enabled, allowing new/updated CDMs and base code to be downloaded, and written to flash memory.

The TELNET service is enabled, with the user ‘PROGRAM’, allows access to both ‘qr’ service and ‘ipc’ service.

The HTML web service is enabled, no sign-in is required (regardless of security), allowing access to all web-pages, and adjustment of any adjustable values.

The HTML web service also enables the ‘Configure’ page, which allows the browser-user to specify the IP address information, along with other data.

Warranty

North warrants goods of its manufacture as being free of defective materials and faulty workmanship for one year from the date of purchase. If this product should become defective please contact our support team. This warranty becomes invalid if the product has been tampered with or used in an environment that is unsuitable.

If you require further help, please contact our support team on +44 (0) 1273 694422, or visit *www.northbt.com/support*

EMC Conformance

The design and manufacture of North products conforms to the appropriate Electromagnetic Compatibility (EMC) standards, but correct installation is required to ensure that performance is not compromised.

End of Life Recycling

WEEE Directive

The WEEE Directive (2002/96/EC) requires the recycling of waste electrical and electronic equipment.

The crossed out wheelie bin symbol found on our products signifies that it should not be disposed of in general waste or landfill.

At the end of its life, return this product to your local supplier upon the purchase of equivalent new equipment, or dispose of it at designated collection points.

Battery Directive

The Battery Directive (2006/66/EC) aims to reduce the environmental impact of the manufacture and use of batteries.

The removable lithium battery contained within this product is marked with the crossed out wheelie bin symbol. This signifies that it should not be disposed of as unsorted municipal waste at the end of its working life.

For further information on end of life recycling visit *www.northbt.com/go/weee*, or contact our support team on +44 (0) 1273 694422.

Commander Versions

Build Date	Details
21/01/2012	Released
10/05/2012	Low-level TCP idle timeout increased
09/09/2012	Recording to COM port now closed when attempting to use driver on same COM
08/04/2013	Reset Reason object now provides indication of reset caused by power and user Resolved issue with defrag eating messages Resolved issue with PING messages larger than 128 bytes Resolved issue when DNS server not reachable Resolved DHCP renewal issue Modified RS232 flow control pins
01/07/2013	Telnet driver rework
19/09/2013	Now uses fixed values provided for gateway and NTP if not provided by DHCP server
01/03/2014	Resolved issue when receiving intensive messages from IP network Added Last Time Sync information
01/04/2014	Added support for version 2.0 hardware Resolved issue when terminating TCP connection
20/08/2014	Resolved issue when Ethernet link established, but no IP address Add new hardware RTC support, including adjustments Updated NTP code NOTE: Commander will clear it's memory on installing this update
02/06/2015	Battery hysteresis added to stop repeat battery alarms
07/03/2016	Added support for new internal requests (FastComms). Increased number of pre-installed drivers. Essential Data upgraded to 640 objects. ObvProcess: new Raise-Lower and Usage modules. New objects to monitor state on remote-object actions. Minor changes some module categories to tie in with ObVerse Manual.
01/11/2016	Add support for 300 and 600 baud on serial ports. Resolved issue on receiving short TCP/UDP messages with additional padding Update to DHCP
24/02/2017	Improve operation when receiving a broadcast storm on the Ethernet network

Next Steps…

If you require help, contact support on 01273 694422 or visit *www.northbt.com/support*

North Building Technologies Ltd
+44 (0) 1273 694422
support@northbt.com
www.northbt.com

This document is subject to change without notice and does not represent any commitment by North Building Technologies Ltd.

ObSys and Commander are trademarks of North Building Technologies Ltd. All other trademarks are property of their respective owners.

© Copyright 2017 North Building Technologies Limited.

Author: TM
Checked by: JF

Document issued 13/03/2017.