



The TextOut Driver

The TextOut driver allows a North device to send simple text commands to connected third-party equipment. The connection can be made of RS232 and TCP/IP. The driver can send a range of predefined text strings, which can be triggered in a variety of ways. Available for Commander and ObSys.

This document relates to TextOut driver version 3.0

Please read the *Commander Manual* or *ObSys Manual* alongside this document, available from www.northbt.com

Contents

Purpose of TextOut Driver	3
Values	3
Detailed Operation	4
RS232 Connection.....	4
TCP/IP Connections	4
Object Reference	4
Output Text to Send	5
Example Configuration.....	6
Designing Your Objects and Configuring the Driver	7
Using the Driver	8
Making the Cable	8
Starting the Interface.....	8
Setting up the Driver.....	9
Checking Communications	9
Object Specifications.....	10
Example Object Reference	10
Device Top-Level Objects	10
TextOut Setup	11
RS232 Connection Setup	12
TCP Connection Setup.....	12
Output Setup	13
TextOut System	14
Driver Versions	15

Purpose of TextOut Driver

The TextOut driver allows a North device to send simple text strings to third-party equipment. Connections can be made over RS232 or TCP/IP. Up to 5 connections can be made at one time.

Text can be sent via all connections simultaneously, or can be sent via a single connection. The driver can send a range of predefined text strings, which can be triggered in a variety of ways.

Configure each connection for the destination equipment. The first destination always sends data via the RS232 port, the other four support connections using TCP/IP.

Configure each text output with an object reference (with optional value and attribute) to trigger it. Up to 250 text outputs are supported.

The TextOut driver could be used to trigger commands on a CCTV matrix, or any system that accepts simple ASCII commands via a serial RS232 or TCP/IP connection.

Alternative generic output drivers are available. Use the Printer driver to send information to a serial printer, or JSONNotify to send web-based messages. The AlmSet and TelDial drivers can also be used to trigger events based on specific North-format alarm messages.

Values

The TextOut driver provides a way of triggering simple text output strings from an object reference. Each output string can include:

- Printable ASCII text, i.e. 'A-Z', '0-9', etc.
- Non-printable text, i.e. carriage-return, line-feed, ASCII code
- Variable fields from the object reference, i.e. attribute or value
- Short wait time

The driver cannot receive data from the third-party device or system.

Detailed Operation

The TextOut driver contains up to 250 outputs defined by the engineer. Each output allows the engineer to define an object reference. When some other task within the North device sets this object reference to a value, the driver sends a predefined text string to the third-party device. The TextOut driver sends text over either an RS232 connection, or a TCP/IP connection.

RS232 Connection

If the RS232 COM port is specified, the driver can send text via this connection.

TCP/IP Connections

Up to 4 TCP connections can be specified.

If the IP Address and Port are configured, the driver automatically attempts to open a TCP connection with the IP address, using the specified port. The driver can be set to send a heartbeat message to test the TCP/IP link.

Object Reference

To trigger some text being sent, some other task within the device sends a write request to a particular object within the driver.

To identify a particular connection, the first part of the object's complete reference must begin 'Qx', where x is the connection number in the range 1..5. If this part is omitted, the driver will send any output string to all connections.

To identify a particular output, the engineer defines an output reference, along with an optional attribute and value. When a task writes to the TextOut driver, all three of these items must match before the text is sent.

Connection

This object part is optional. If it exists, it has the format 'Qx', where x is in the range 1..5. 'Q1' specifies the RS232 connection. 'Q2'..'Q5' specifies a TCP connection 2..5.

Reference

The engineer defines the output reference as a short four-character identifier.

Attribute

The engineer may define an attribute. This is a short three-character identifier. If defined, the task must write to the object reference and attribute to cause the sending of the text.

Value

The engineer may define a value. This is a seven-character text string. If defined, the task must write that value to the object reference to cause the sending of the text.

Output Text to Send

The engineer defines the output text to send. This is a text string of up to 63-characters. This text should contain a command to make the third-party system respond in a particular way.

The text may contain variables, which are replaced with data from the complete object reference before the text is sent. It is possible to include the attribute and value, special characters such as a carriage-return and line-feed, and a wait time to allow the third-party time to action a command.

Replacement Text

These special replacement text sequences may be used to specify a variable when configuring the text to send. They are replaced when the text is being sent.

Text	Replaced by
[AT]	The attribute of the object written
[VA]	The value written to the object
[CR]	Carriage Return character (ASCII code 13)
[LF]	Line Feed character (ASCII code 10)
[WT]	Causes a short wait time to occur while sending the text output. The duration of the wait is configured in the Wait Time driver object (WT)
[<i>ddd</i>]	ASCII code <i>ddd</i> . Where <i>ddd</i> is a 3-digit decimal number corresponding to the ASCII code, e.g. 065 = 'A'

Example Configuration

Imagine the TextOut driver set up as below:

Output Number	Reference	Attribute	Value	Send Text
1	ALM			Alarm[CR][LF]
2	DR	OP	1	Door Left Open[CR][LF]
3	CAM			MON1 CAM[VA][CR][LF]
4	Z		1	Zone [AT] Fire[CR][LF]
5	Z		0	Zone [AT] Ok[CR][LF]
6	ST			Unit [AT] = [VA][CR][LF]
..				

Notice the Sent Text field for each output contains a trailing carriage-return and line-feed – '[CR][LF]'. Most systems use these as a signal for the end of a command.

When a task elsewhere in a North device writes to the following objects in the TextOut System, it will have the following effect (carriage-return and line-feed are not shown).

Object Written	Value Written	Text sent by driver	Notes
ALM	1	ALARM	Output 1 triggered to all available connections.
ALM.X	2	ALARM	Output 1 triggered to all available connections. No Attribute or Value is defined in output, so any object will cause the sending of the text
DR.OP	1	Door Left Open	Output 2 triggered to all available connections. The Reference, Attribute and Value matched the first part of the object, the second part of the object and the value respectively
DR.OP	0		No text sent. Value did not match Output 2.
DR.CL	1		No text sent. Attribute did not match Output 2
CAM	17	MON1 CAM17	Output 3 triggered to all available connections. '[VA]' replaced with the value written
CAM.X	3	MON1 CAM3	Output 3 triggered to all available connections. No Attribute is defined in output, so any object will cause the sending of the text. '[VA]' replaced with the value written
Z.1	1	Zone 1 Fire	Output 4 triggered to all available connections. Reference and Value match. '[AT]' replaced with attribute
Z.1	0	Zone 1 Ok	Output 5 triggered to all available connections. Reference and Value match. '[AT]' replaced with attribute
Z.A	1	Zone A Fire	Output 4 triggered to all available connections. Reference and Value match. '[AT]' replaced with attribute
ST.A	12	Unit A = 12	Output 6 triggered to all available connections. '[AT]' replaced with attribute, and '[VA]' replaced with value
Q1.DR.OP	1	Door Left Open	Output 2 triggered to RS232 connection only
Q3.DR.OP	1	Door Left Open	Output 2 triggered to TCP Connection 2 only

Designing Your Objects and Configuring the Driver

The reference for an output must always match the first part of the object reference. If the attribute or the value is left blank when configuring the driver, then these fields will not be checked.

In Output 1 above for example, writing to object ALM.ABC with the value 'OK' will still trigger the text output as both the attribute and value are blank. Only the first part of the object reference, ABC, needs to match for this output.

To design your object references, look at the protocol for the serial device. What text does it expect?

Should part of the text contain a variable field such as a camera number, zone or other reference; you could use the attribute and value fields to include this information.

Example

Let us consider a simple CCTV camera matrix. It supports the following simple commands:

To move a camera to a pre-set position, send:

```
CAM<camera number> PRE<preset number><<cr>
```

To select a camera on a monitor, send:

```
MON<monitor number> CAM<camera number><<cr>
```

To activate an alarm, send:

```
ALARM<input number><<cr>
```

To clear an alarm, send:

```
CLEAR<input number><<cr>
```

So, let's configure the following outputs in the driver:

Output Number	Reference	Attribute	Value	Send Text
1	CAM			CAM[AT] PRE[VA][CR]
2	MON			MON[AT] CAM[VA][CR]
3	ALM		1	ALARM[AT][CR]
4	ALM		0	CLEAR[AT][CR]

This will make the following object references available in the TextOut system:

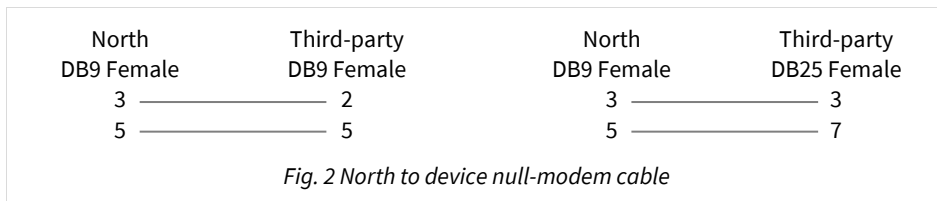
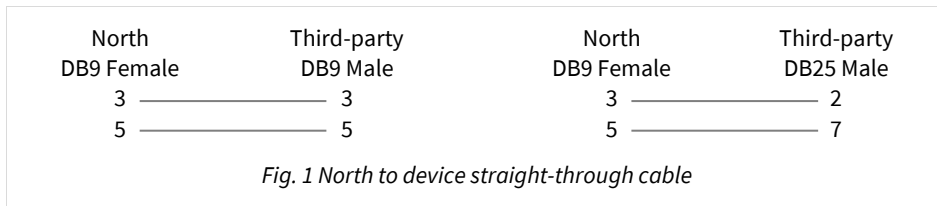
Description	Reference	Type
Camera <i>c</i> Preset Position Move camera <i>c</i> to pre-set position. Value contains pre-set number. E.g. Set CAM.5 to '1' sends 'CAM5 PRE1'	CAM. <i>c</i>	Obj\Num; Adjustable
Monitor <i>m</i> Camera Select camera on monitor <i>m</i> . Value contains camera number. E.g. Set MON.2 to '5' sends 'MON2 CAM5'	MON. <i>m</i>	Obj\Num; Adjustable
Alarm <i>i</i> Activate Activate/Clear the alarm <i>i</i> . E.g. Set ALM.16 to '1' sends 'ALARM16'	ALM. <i>i</i>	Obj\NoYes; Adjustable

Using the Driver

On ObSys and Commander, the TextOut driver is pre-installed. On all of these North devices, you can use the driver to create an interface to a third-party device or system. Once started, you will need to set up the driver before it can communicate with the device.

Making the Cable


The RS232 cable specification will depend on the device used. Check product documentation to determine the cable required. Typically, connect the North device COM port to the third-party system RS232 port using a straight-through (Fig. 1) or null-modem cable (Fig. 2). Connector types at each end of the cable are shown.



The driver outputs an RTS signal to the device (North device pin 7) to indicate its presence, although its use is optional.


The maximum RS232 cable length is 15m, and should be made as short as possible.

Starting the Interface

-  To start an interface using the TextOut driver, follow these steps:
- **Start Engineering** your North device using ObSys
 - Navigate to **Configuration, Interfaces**, and set an unused **Interface** to 'TextOut' to start the particular interface
 - Navigate to the top-level of your North device and re-scan it

The driver setup object (Mc), labelled **TextOut Setup**, should now be available. If this object is not available, check an interface licence is available and the driver is installed.

Setting up the Driver

-  To set up the driver, follow these steps:
- Navigate to the **TextOut setup** object (Mc). For example, if you started interface 1 with the driver earlier, then the object reference will be 'M1'
 - For an RS232 connection, set **RS232 Com Port** object (RS.COM) to select which serial port on the North device the third-party is connected to. Also set a **RS232 Baud Rate** (RS.BR) and **RS232 Byte Format** (RS.BF) to match the device. Set up the relevant objects within **RS232 Connection Setup** also
 - For any TCP/IP connections, set the **TCP IP Address** (LP) and **TCP Port** (LA) objects with the TCP Connection x Setup objects
 - Set up at least one **Output** object with a **Reference** (O) and **Send Text** (S), to create an object in the TextOut System.

Checking Communications

For TCP/IP connections, check the **TCP Link State**. A state of 'Opened' (2) indicates the driver has connected to the third-party system.

Write to one of the output objects to trigger the output text, then check for the desired result.

The driver does not receive data from third-party system, so there is no method of confirming whether the text was received by the device.

Object Specifications

Once an interface is started, one or more extra objects become available within the top-level object of the device. As with all North objects, each of these extra 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 in the same device: the TextOut Setup (M1) contains a Wait Time object (WT) – therefore the complete object reference is 'M1.WT'.

An example of a reference to an object in a different device: the IP network object (IP) contains the Default Commander object (CDIP), which contains the object above (M1.WT) – therefore the complete object reference is 'IP.CDIP.M1.WT'.

Device Top-Level Objects

When an interface is started using the Printer driver, the objects below become available within the top-level object of the device. For example, if Interface 1 is started, then the object with references 'M1' and 'S1' become available.

Description	Reference	Type
TextOut Setup Set up the TextOut driver, started on interface <i>c</i> (<i>c</i> is the interface number)	Mc	Fixed Container: On the Commander platform this will be [CDM v20\TextOut v30] On the ObSys platform this will be [OSM v20\TextOut v30]
TextOut System Send commands to third-party system connected to interface <i>c</i> (<i>c</i> is the interface number)	Sc	Variable Container: [TextOut v30]

TextOut Setup

Object Type: *[OSM v20\TextOut v30]*

Object Type: *[CDM v20\TextOut v30]*

The TextOut Setup contains the following objects:

Description	Reference	Type
RS232 COM Port	RS.COM	Obj\Num; Range: 0...8; Adjustable Set to '0' to disable RS232 connection
RS232 Baud Rate	RS.BR	Obj\Num; Adjustable Values: 1200,2400,4800, 9600, 19200, or 38400
RS232 Byte Format Sets the parity, data bits, and stop bits	RS.BF	Obj\ENum: 0..11; Adjustable See note 1
RS232 Connection Setup	Q1	Variable Container: On the Commander platform this will be <i>[CDM v20\TextOut v30\ComQueue]</i> On the ObSys platform this will be <i>[OSM v20\TextOut v30\ComQueue]</i>
TCP Connection x Setup Where x is in the range 2..5	Qx	Variable Container: On the Commander platform this will be <i>[CDM v20\TextOut v30\TcpQueue]</i> On the ObSys platform this will be <i>[OSM v20\TextOut v30\TcpQueue]</i>
Output x Configure text output x, where x is in the range 1..250	Ox	Fixed container: On the Commander platform this will be <i>[CDM v20\TextOut v30\Output]</i> On the ObSys platform this will be <i>[OSM v20\TextOut v30\Output]</i>

Notes

- 1 RS232 Byte format can have the following values:

Value	Parity	Data bits	Stop bits
0	None	8	1
1	None	8	2
2	None	7	1
3	None	7	2
4	Odd	8	1
5	Odd	8	2
6	Odd	7	1
7	Odd	7	2
8	Even	8	1
9	Even	8	2
10	Even	7	1
11	Even	7	2

RS232 Connection Setup

Object Type: [OSM v20\TextOut v30\ComQueue]

Object Type: [CDM v20\TextOut v30\ComQueue]

The RS232 Connection Setup contains the following objects:

Description	Reference	Type
Heartbeat Rate (s) Period of time after which the driver will send the TCP Heartbeat message. The rate time will start counting after the last message has been sent. Choose None to disable Heartbeat messages	LH	Obj\Enum; Range 0..4; Adjustable 0=None; 1=1minute, 2=5minutes, 3=15minutes, 4=1hour
Heartbeat Message	HM	Obj\Text; Max chars: 20; Adjustable Can contain [CR], [LF], and [xxx] replacement texts
Wait Time (ms) Period of time (in milliseconds) to wait after sending one text string from an Output before sending the next. Also used as the wait time to insert when using the special text '[WT]' in an output's Send Text.	WT	Obj\Num; Adjustable; Range 0..10000

TCP Connection Setup

Object Type: [OSM v20\TextOut v30\TcpQueue]

Object Type: [CDM v20\TextOut v30\TcpQueue]

The TCPConnection Setup contains the following objects:

Description	Reference	Type
IP Address Address of device to link using TCP/IP. If set to '0.0.0.0' then connection unused	LA	Obj\IP; Adjustable
Port Port number of device to link to using TCP/IP. If set to '0' then connection unused	LP	Obj\Num; Range: 0..65535; Adjustable
Link State Current state of TCP connection	LS	Obj\Enum; Range 0..2 0=Closed, 1=Opening, 2=Opened
Heartbeat Rate (s) Period of time after which the driver will send the Heartbeat message. The rate time will start counting after the last message has been sent. Choose None to disable Heartbeat messages	LH	Obj\Enum; Range 0..4; Adjustable 0=None; 1=1minute, 2=5minutes, 3=15minutes, 4=1hour
Heartbeat Message	HM	Obj\Text; Max chars: 20; Adjustable Can contain [CR], [LF], and [xxx] replacement texts
Wait Time (ms) Period of time (in milliseconds) to wait after sending one text string from an Output before sending the next. Also used as the wait time to insert when using the special text '[WT]' in an output's Send Text.	WT	Obj\Num; Adjustable; Range 0..10000

Output Setup

Object Type: [OSM v20\TextOut v30\Output]

Object Type: [CDM v20\TextOut v30\Output]

An Output Setup contains the configuration for a text output, which defines the object reference created in the System object and text to be sent. Refer to the [Detailed Operation](#) section for further information.

It contains the following objects:

Description	Reference	Type
Reference The main reference of the object, used to trigger the sending of the text	O	Obj\Text; Adjustable; Max Chars: 4
Attribute Any sub-object within the object reference (above). Optional, although if specified, must match that specified in the object reference when triggering the sending of the text	A	Obj\Text; Adjustable; Max chars: 3
Value Any value. Optional. If specified, must match that written to the object reference (and optional attribute) above to trigger the sending of the text.	V	Obj\Text; Adjustable; Max chars: 7
Send Text The text to send when triggered by the object (and optional attribute) being written. May contain variables that are replaced in the text being sent.	S	Obj\Text; Adjustable; Max chars: 63 Refer to Replacement Text section for list of variables supported

TextOut System

Object Type: [TextOut v30]

This contains example objects to ease engineering. The engineer creates objects in the module's *Output Setup* object.

Refer to the *Designing Your Objects and Configuring the Driver* section for further information.

Description	Reference	Type
Reference - Attribute Object that is written to trigger sending of text to all connections. Note: This object may require additional attributes, and therefore may not be complete.	<i>ref.att</i>	Obj\Text; Adjustable
Connection - Reference - Attribute Object that is written to trigger sending of text to a particular connection. Note: This object may require additional attributes, and therefore may not be complete.	<i>Qx.ref.att</i>	Obj\Text; Adjustable

Driver Versions

Version	Build Date	Details
1.0	18/11/2011	Initial driver released
2.0	01/03/2016	Increased object count, improved speed, added more replacement symbols, added Wait Time
2.1	10/06/2016	Support for RS232 or TCP/IP included
3.0	17/08/17	Support for simultaneous RS232 and 4 TCP connections added

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 2022 North Building Technologies Limited.

Author: TM
Checked by: JF

Document issued 08/09/2022.