



# The DeviceCheck Driver

---

The DeviceCheck driver monitors the health of remote devices and web servers. When the driver detects a problem reaching a device, it can notify you by sending a North-format alarm message, and trigger an automated action by setting an object value within a device. Available for Commander and ObSys.

This document relates to DeviceCheck driver version 1.0

Please read the *Commander Manual* or *ObSys Manual* alongside this document, available from [www.northbt.com](http://www.northbt.com)

# Contents

Purpose of DeviceCheck Driver .....	3
Values .....	3
Prerequisites .....	3
Detailed Operation .....	4
Checking a Device .....	4
Device Health .....	5
Checking the LAN Connection .....	5
Using the Driver .....	6
Starting the Interface .....	6
Setting up the Driver .....	6
Checking Communications .....	6
Alarms .....	7
Format .....	7
Examples .....	7
Object Specifications .....	8
Device Top-Level Objects .....	8
Device Check .....	9
Local Network .....	10
Check .....	11
Advanced Settings .....	13
Driver Versions .....	14

# Purpose of DeviceCheck Driver

The DeviceCheck driver monitors the health of remote devices and web servers. When the driver detects a problem reaching a device, it can notify you by sending a North-format alarm message, and trigger an automated action by setting an object value within a device.

The driver can monitor an object reference from a North device or connected system, request a web resource, or test a TCP connection to a server.

For each device to check, configure the resource to monitor, failure threshold, and action to perform if it is unresponsive. Up to 100 devices can be monitored.

The DeviceCheck driver could be used to provide notification of an unresponsive device; then perform an action such as remotely restart a PC, power-cycle the device, or activate back-up equipment.

## Values

DeviceCheck monitors a device and presents a state to indicate if it is healthy or not.

When a device's health changes, one of the following actions can be performed:

- Set an object to a particular value when the device is not healthy
- Set the value of an object to 'off', and then 'on', when the device is not healthy
- Set the value of an object to the current health, whenever the health state changes
- Trigger an application when the device is not healthy (ObServer platform only).

A device can be configured to send an alarm message when its health state changes.

## Prerequisites

The North device's LAN port should be configured with a DNS server address if you are checking a devices using their host name.

If you are monitoring a remote device via a firewall, then the driver will require outbound access on the configured TCP ports.

# Detailed Operation

## Checking a Device

The driver can make a request to the following types of device and check that they remain responsive:

- Web page or URL from an HTTP server
- TCP endpoint from a telnet, email, camera, or API server, etc.
- Object reference from any device connected to the North system.

A device is checked periodically, as set by the Request Interval object. If a check succeeds then the fail count is set to '0', if the check fails then the fail count is incremented.

### HTTP Server

DeviceCheck monitors a web page or URL from an HTTP server by requesting the resource (sending an HTTP HEAD request). The server can be specified using its host name or IP address. Use this to monitor a remote website, or any device with an embedded web server.

If the server responds with a HTTP success code (2xx or 3xx), then the fail count is set to '0'. If the server responds with an HTTP error code (4xx or 5xx) or fails to respond, then fail count is incremented.

The Last Response object contains the HTTP response code received along with the resolved IP address.


 To monitor the web page 'www.northbt.com/robots.txt', follow these steps:

- Navigate to an unused **Check** object
- Set **Check Method** to 'HTTP'
- Set **Host name, IP or Object** to 'www.northbt.com'
- Set **HTTP Resource Name** to '/robots.txt'.

### TCP Endpoint

DeviceCheck monitors a TCP endpoint by attempting to open a connection to the server on the specified TCP port number. Use this when a device doesn't have an embedded web server, but does has another TCP service available – such as Telnet, SMTP, API, etc. The endpoint can be specified using its host name or IP address.

If the server accepts the connection, then the fail count is set to '0'. If the server rejects the connection or fails to respond, then fail count is incremented.

 To monitor a Telnet server (port 23) at 192.168.1.56, follow these steps:


- Navigate to an unused **Check** object
- Set **Check Method** to 'TCP'
- Set **Host name, IP or Object** to '192.168.1.56'
- Set **TCP Port number** to '23'

### Object Reference

DeviceCheck monitors any object reference by reading it. Use this to monitor any system connected to a North device. Other options may be available – using Essential Data to monitor an object, or a driver may already generate offline alarm messages.

If the interface responds with a value, then the fail count is set to '0'. If the interface responds with a fail code or fails to respond, then fail count is incremented.

The Last Response object indicates if the last check was a 'Success', or if it 'Failed' along with the fail reason.

 To monitor the Modbus object 'S2.A5.D1000.A', follow these steps:

- Navigate to an unused **Check** object
- Set **Check Method** to 'Object Reference'
- Set **Host name, IP or Object** to 'S2.A5.D1000.A'

## Device Health

A device is healthy, with Healthy state set to 'Yes', when the fail count is '0'. If Fail Count exceeds the Maximum Fails, then Healthy changes to 'No'.

Changes to a device's health can trigger both an alarm event and an alarm message.

## Alarm Event

An alarm event can be used to perform an action when a device becomes unresponsive; such as power-cycle a device, activate back-up equipment, or restart a PC.

When a device's health changes, one of the following actions can be performed:

- Set object value: when Healthy changes to 'No', sets the Alarm Event Object to Alarm Event Value
- Pulse object off-on: when Healthy changes to 'No', sets the Alarm Event Object to 'off' then 'on'
- Set object to health state: sets the Alarm Event Object to the current Healthy state, whenever Healthy changes
- Trigger application: when Healthy changes to 'No', run the command line Alarm Event App with parameters in Alarm Event Params (ObServer platform only).

## Alarm Message

When the Healthy state changes, and alarms are enabled by setting Alarm Priority/Enable object to a value in the range '1' to '9', a North-format alarm message will be sent. A North device's alarm processing capabilities include routing the alarm to a destination, store in a history of alarms, send to a mobile phone or by email.

Refer to the [Alarms](#) section for details of the alarm message format.

## Checking the LAN Connection

The driver monitors the local North device's LAN connection. This can be useful to check the LAN port has been correctly configured.

DeviceCheck performs the following checks to monitor the LAN connection:


- Network available: the Ethernet link is established and a local IP address assigned
- Gateway available: can find a route to a public IP address
- DNS available: can resolve a public website hostname (www.bbc.co.uk)
- Internet available: can connect to an Internet site

# Using the Driver

On ObSys and Commander, the DeviceCheck driver is pre-installed. Once started, you will need to set up the driver before it can monitor the health of a remote device.


The DeviceCheck driver uses zero licence units.

## Starting the Interface

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

The driver setup object (Mc), labelled **Device Check**, 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 **Device Check** object (Mc). For example, if you started interface 1 with the driver earlier, then the object reference will be 'M1'
  - Configure at least one **Check** object with a **Label** (L) and **Check Method** (PR)
  - Define the HTTP, TCP or object reference resource to check by setting a **Host name** (IA), **TCP Port number** (PN), and **HTTP Resource Name** (RN)
  - Set the **Request Interval** (R), and **Maximum Fails** (MF)
  - To enable North-format alarms, set **Alarm Priority** (P)
  - To perform an action if the remote device fails to respond, set **Alarm Event** objects

## Checking Communications

Each check contains a **Last Check Time** (LT), **Last Response** (LR), and **Fails** (F) object. Use these to check the parameters have been correctly configured, and that the remote device is responding.

You can check the health of a device by reading its **Healthy** (S) object.

# Alarms

When a device's health status changes, the DeviceCheck driver sends an alarm.

## Format

North-format alarms contain six text fields. The DeviceCheck driver places the following information into these fields:

**System** – copied from System Label object (DL) within driver setup

**Point** – copied from the check's Label object (Sx.L)

**Condition** – either 'Communications Fault' or 'Communications OK'

**Priority** – copied from the check's Alarm Priority object (Sx.P)

**Date & Time** – from North device

## Examples

System	Point	Condition	Priority	Date	Time
Device Check	Web server	Communications Fault	3	12/04/16	14:22:00
Device Check	Web server	Communications OK	3	12/04/16	14:22:00

# 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.

## Device Top-Level Objects

When an interface is started using the DeviceCheck driver, the objects below become available within the top-level object of the device. For example, if interface 1 is started, then the object reference 'M1' becomes available.

Description	Reference	Type
<b>Device Check</b> Set up the DeviceCheck driver, started on interface c (c is the interface number)	Mc	Fixed Container: On the Commander platform this will be <a href="#">[CDM v20\DeviceCheck v10]</a> On the ObSys platform this will be <a href="#">[OSM v20\DeviceCheck v10]</a>



# Device Check

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

Object Type: *[OSM v20\DeviceCheck v10]*

Device Check contains the following objects for configuring the driver:

Description	Reference	Type
<b>System Label</b> Label displayed when scanning the system	DL	Obj\Text: 20 chars; Adjustable
<b>All Checks Healthy</b> Summary of all device checks and network available	DS	Obj\NoYes
<b>Total Devices Healthy</b> Number of device checks configured, with a healthy state	TDH	Obj\Num: 0...100
<b>Total Devices Alarm</b> Number of device checks configured, with a non-healthy (alarm) state	TDA	Obj\Num: 0...100
<b>Total Devices Checked</b> Number of device checks configured	TDC	Obj\Num: 0...100
<b>LAN Connection</b> Monitor or check the local network connection	N	Fixed Container: On the Commander platform this will be <i>[CDM v20\DeviceCheck v10\Local]</i> On the ObSys platform this will be <i>[OSM v20\DeviceCheck v10\Local]</i>
<b>Check x</b> Configure a device to monitor, x is in the range 1...100	Sx	Fixed Container: On the Commander platform this will be <i>[CDM v20\DeviceCheck v10\Check]</i> On the ObSys platform this will be <i>[OSM v20\DeviceCheck v10\Check]</i>
<b>Advanced Settings</b>	A	Fixed Container: On the Commander platform this will be <i>[CDM v20\DeviceCheck v10\Advanced]</i> On the ObSys platform this will be <i>[OSM v20\DeviceCheck v10\Advanced]</i>

## LAN Connection

Object Type: [OSM v20\DeviceCheck v10\Local]

Object Type: [CDM v20\DeviceCheck v10\Local]

Checks the state of the local LAN connection.

Description	Reference	Type
<b>Network available</b> Can open a network connection	NET	Obj\NoYes
<b>Gateway available</b> Valid gateway available	GW	Obj\NoYes
<b>DNS available</b> Can resolve a website host name	DNS	Obj\NoYes
<b>Internet available</b> Can connect to an Internet resource	WWW	Obj\NoYes
<b>All checks healthy</b> All the above checks have passed	S	Obj\NoYes

# Check

Object Type: [OSM v20\DeviceCheck v10\Check]

Object Type: [CDM v20\DeviceCheck v10\Check]

Use the check object to configure the remote device or server to monitor. Refer to [Detailed Operation](#) section.

The driver supports three different methods to check a device:

- HTTP protocol – requests a resource from a web server, e.g. <http://www.northbt.com/robots.txt>
- TCP protocol – attempts to open a TCP connection to the port number on an endpoint server. Monitor telnet, email, API interfaces, etc.
- Object reference – requests any object from the North system, monitoring the connection to a remote device or third-party system.

Description	Reference	Type
<b>Label</b> Description of device to monitor	L	Obj\Text: 20 chars; Adjustable
<b>Check Method</b> Protocol to connect with the device	PR	Obj\Enum; Adjustable Values: 0=HTTP, 1=TCP, 2=Object Reference
<b>Host name, IP or Object</b> Address of the device. For HTTP and TCP check methods, set this to the host name (requires a DNS server), or IP address of the server. For Object Reference checks, set this to the object reference in the remote device to monitor.	IA	Obj\Text: 60 chars; Adjustable
<b>TCP Port number</b> For HTTP and TCP check methods, set to the TCP port number on the server. By default HTTP connections use port 80.	PN	Obj\Num: 0...65535; Adjustable
<b>HTTP Resource Name</b> For HTTP checks, set to the resource to retrieve, e.g. '/robots.txt'	RN	Obj\Text: 125 chars; Adjustable
<b>Request Interval</b> How often to connect to the device. If the remote device fails to respond then a shorter retry time is used	R	Obj\Enum; Adjustable Values: 0=Default (15mins), 1=1min, 2=5mins, 3=15mins, 4=1hr, 5=4hrs, 6=12hrs
<b>Maximum Fails</b> Failure threshold. Maximum number of retries until the Healthy state changes to 'No'	MF	Obj\Num: 1...20; Adjustable
<b>Alarm Priority/Enable</b> Enables sending North-format alarm messages, with the specified priority, when the health changes	P	Obj\Num: 0...9; Adjustable
<b>Alarm Event: Action</b> Action to trigger when the health changes. Actions 1, 2, & 4 are only triggered when the device changes to not healthy. Action 3 is triggered when the device health changes	EA	Obj\Enum; Adjustable Values: 0=None, 1=Set object value, 2=Pulse object off-on, 3=Set object to health state On the ObSys platform the additional value is available: 4=Trigger application
<b>Alarm Event: Object/App</b> Object reference, or application path to use with selected action	EO	Obj\Text: 125 chars; Adjustable

Description	Reference	Type
<b>Alarm Event: Value/Params</b> Object value, or application parameters to use with selected action	EV	Obj\Text: 125 chars; Adjustable
<b>Healthy</b> Indicates if the monitored device is responding and healthy	S	Obj\NoYes
<b>Fail Count</b> Number of times the device has failed to respond	F	Obj\Num
<b>Last Check Time</b> Date and time the device was last checked	LT	Obj\DateTime
<b>Last Response</b> Information about the last response from the device	LR	Obj\Text
<b>Last Response Value</b> For Object Reference check methods, contains the last value read. To set this value when Healthy state changes to 'No', set Alarm Event Action to 'None' and Alarm Event Value to the value required.	LV	Obj\Text

# Advanced Settings

Object Type: [OSM v20\DeviceCheck v10\Advanced]

Object Type: [CDM v20\DeviceCheck v10\Advanced]

Advanced settings for the driver.

Description	Reference	Type
<b>Enable Checking</b> Enable/disable operation of all checking	E	Obj\NoYes; Adjustable
<b>Server Timeout (ms)</b> Maximum time to wait for a remote server to respond	TO	Obj\Num:500...60,000; Adjustable

# Driver Versions

Version	Build Date	Details
1.0	30/5/2018	Driver released
1.0	9/11/2018	Improved local LAN port checking. Remote server timeout default value changed to 10s HTTP response 401 (Unauthorised) now accepted as success. When a check fails, driver re-requests after Request Interval (previously this was 1 min).

## Next Steps...

If you require help, contact support on 01273 694422 or visit [www.northbt.com/support](http://www.northbt.com/support)



North Building Technologies Ltd  
+44 (0) 1273 694422  
[support@northbt.com](mailto:support@northbt.com)  
[www.northbt.com](http://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: JF  
Checked by: JP

Document issued 08/09/2022.