|  |  |
| --- | --- |
|  | |
|  | |
|  |  |

|  |  |
| --- | --- |
| Publication type: |  |
| Publication number: | ACC |
| Title: |  |
| Subject: |  |
| Issue: | A2.1.10 |
| Publication date: | 9 October 2014 |
| Total number of pages: |  |
| Author: |  |
| Quality Control: |  |

Table of contents

Page #

Figures 5

Tables 7

References 9

Abbreviations list 10

Safety instructions 11

Revision history 11

1. What is the Sensorlist 12

1.1 Introduction 12

1.2 Excel 12

1.3 Saving and naming 13

1.4 Saving as sensorlist for import 14

2. Devicelist 15

2.1 Introduction 15

2.2 Columns 16

2.3 IP-addresses 21

2.3.1 Introduction 21

2.3.2 IPAddressUp- IPAddressDown 21

2.4 Mac addresses 22

2.4.1 Introduction 22

2.5 How to implement this in the devicelist 22

2.5.1 Introduction 22

2.5.2 The devices 23

2.5.3 Comment 26

2.5.4 Location 26

2.5.5 Protocol 26

2.5.6 Interface 29

2.5.7 Port and Source 30

2.5.8 Type 31

2.5.9 Speed, Datalink and Hardware 33

2.5.10 Options 34

2.5.11 IP addresses and MAC addresses 34

2.5.11.1 Other devices 37

3. Sensorlist 41

3.1 Introduction 41

3.2 Columns 41

3.3 Implementation in the sensorlist 49

3.3.1 Introduction 49

3.3.2 Import Result 50

3.3.3 ID, CableLabel, GroupLabel 50

3.3.4 Item 50

3.3.4.1 Conjunction with SensorType 51

3.3.5 SensorType 51

3.3.5.1 Fields 52

3.3.5.2 Back to SensorType 52

3.3.6 Connection 53

3.3.7 Device 53

3.3.8 Location 54

3.3.9 Interface 54

3.3.10 Module 55

3.3.11 Pin 55

3.3.11.1 Wago 56

3.3.11.2 Serial Protocols 58

3.3.12 Type 58

3.3.12.1 Wago 58

3.3.12.2 Serial Protocols 59

3.3.13 Min-Max 59

3.3.14 DefaultUnit 59

3.3.15 Manufacturer 60

3.3.16 Supplier 60

3.3.17 Comment 60

3.3.18 Revision 60

3.3.19 Field 60

3.3.19.1 How to work with fieldlist.txt 61

3.3.19.2 Back to the Field column 63

3.3.20 Label 64

3.3.21 Rate 64

3.3.22 Index 64

3.3.23 Datatype 64

3.3.24 Enum 64

3.3.25 Count 64

3.3.26 Multiplier 65

3.3.27 Offset 65

3.3.28 Unit 65

3.3.29 Other columns 65

3.4 Filter sensorlist 65

3.5 Special issues 68

4. Importing in FT NavVision 71

4.1 Introduction 71

4.2 How to import 71

4.3 Check the import 72

4.4 Devicelist generated 73

4.5 Sensorlist generated 74

4.6 Sensorlist generated diff 75

4.6.1 Making an export 76

5. Keeping up to date 77

5.1 Introduction 77

5.2 Direct changing 77

5.2.1 Insert 78

5.3 Bigger changes 78

5.4 Keep the sensorlist up to date afterwards 81

5.4.1 Introduction 81

5.4.2 What do you need 81

5.4.3 Cleaning up after a day on board 82

5.4.3.1 Copy devicelist.dat and sensorlist.dat 82

5.4.3.2 The old sensorlist 82

5.4.3.3 Startup your project folder 84

5.4.3.4 Inspecting the generated files 84

5.4.3.5 Comment 85

5.4.3.6 Changed 86

5.4.3.7 Failed 86

5.4.3.8 Missing 87

5.4.3.9 New 87

5.4.3.10 Keep importing 87

6. Special notes 88

6.1 Introduction 88

6.2 PLC 88

6.3 Search 90

6.4 Setting NMEA in the sensorlist 91

6.4.1 Interface 91

6.4.2 Module 91

6.4.3 Pin 92

6.4.4 Type 92

6.4.5 Count 93

6.4.6 Count 93

6.5 WatchIO in the sensorlist 93

6.6 CANopen in the sensorlist 95

6.6.1 By hand 96

6.6.2 By import 97

Figures

Figure 1‑1: devicelist 12

Figure 1‑2: Sensorlist 13

Figure 1‑3: Excel saving 14

Figure 1‑4: saving as sensorlist.xls 15

Figure 2‑1: single line drawing 16

Figure 2‑2: numbering the drawing 22

Figure 2‑3: Filling deviceId column 1 23

Figure 2‑4: Filling deviceId column 2 23

Figure 2‑5: Filling deviceId column 3 24

Figure 2‑6: Filling deviceiD column 4 25

Figure 2‑7: Filling location column 26

Figure 2‑8: Filling protocol column 1 27

Figure 2‑9: Filling protocol column 2 28

Figure 2‑10: Filling protocol column 3 29

Figure 2‑11: Filling interface column 30

Figure 2‑12: Port and Source 1 30

Figure 2‑13 Port and Source 2 31

Figure 2‑14: Network Serial colors 32

Figure 2‑15: Filling Type column 1 32

Figure 2‑16: Filling type column 2 33

Figure 2‑17: speed, datalink and hardware 34

Figure 2‑18: Addresses and connection 1 35

Figure 2‑19: Addresses and connection 2 35

Figure 2‑20: Addresses and connection 3 35

Figure 2‑21: Addresses and connection 4 35

Figure 2‑22: Addresses and connection 5 36

Figure 2‑23: Addresses and connection 6 36

Figure 2‑24: Addresses and connection 7 36

Figure 2‑25: Addresses and connection 8 37

Figure 2‑26: Wago Addresses 1 37

Figure 2‑27: Wago Addresses 2 37

Figure 2‑28: Wago Addresses 3 38

Figure 2‑29: Network Serial addresses 1 39

Figure 2‑30: Network Serial addresses 2 39

Figure 2‑31: Network Serial addresses 3 39

Figure 2‑32: Client addresses 40

Figure 3‑1: ID, CableLabel, GroupLabel Example 50

Figure 3‑2: Item example 51

Figure 3‑3: Double fault 53

Figure 3‑4: Sensorlist device column 54

Figure 3‑5: Sensorlist Interface column 55

Figure 3‑6: Sensorlist Module column 1 55

Figure 3‑7: Sensorlist Module column 2 55

Figure 3‑8: Wago Numbering 1 56

Figure 3‑9: Wago Numbering 2 57

Figure 3‑10: Pin column 1 57

Figure 3‑11: pin column 2 58

Figure 3‑12: Pin column 3 58

Figure 3‑13: Type column 1 59

Figure 3‑14: Type column 2 59

Figure 3‑15: Default Unit column 60

Figure 3‑16: open with Excel 61

Figure 3‑17: Excel 1 61

Figure 3‑18: Excel 2 62

Figure 3‑19: Excel 3 62

Figure 3‑20 : Excel 4 62

Figure 3‑21: Excel 5 63

Figure 3‑22: Field column 1 63

Figure 3‑23: Field column 2 63

Figure 3‑24: Custom sort 66

Figure 3‑25: Custom sort window 66

Figure 3‑26: Sorting device and interface 66

Figure 3‑27: Sorted sensorlist 67

Figure 3‑28: Cut and paste 1 68

Figure 3‑29: Cut and paste 2 68

Figure 3‑30: Duty names 69

Figure 3‑31: Call names 70

Figure 3‑32: Special issues 1 70

Figure 3‑33: Special issues 2 71

Figure 4‑1: Root folder 71

Figure 4‑2: Import devicelist 72

Figure 4‑3: Import sensorlist 72

Figure 4‑4: Checking system layout 73

Figure 4‑5: devicelist\_generated good 74

Figure 4‑6: Fault codes 74

Figure 4‑7: sensorlist\_generated 75

Figure 4‑8: Diff example 1 75

Figure 4‑9: Diff example 2 75

Figure 4‑10: Diff example 3 75

Figure 4‑11: Export shortcut 76

Figure 5‑1: changing baudrate old 77

Figure 5‑2: Changing baudrate new 77

Figure 5‑3: Changing Wago original 78

Figure 5‑4: Changing Wago lines 78

Figure 5‑5: Changing Wago numbers 78

Figure 5‑6: Inserting a Wago slice 1 79

Figure 5‑7: Inserting a Wago slice 2 79

Figure 5‑8: Inserting a Wago slice 3 80

Figure 5‑9: Excel trick 1 80

Figure 5‑10: Excel trick 2 81

Figure 5‑11: clean NavVision folder 82

Figure 5‑12: Devicelist.dat and sensorlist.dat in network folder 83

Figure 5‑13: sensorlist.xls in root of project folder 83

Figure 5‑14: root folder after import sensorlist 84

Figure 5‑15: Comment example 1 85

Figure 5‑16: Comment example 2 85

Figure 5‑17: Comment example 3 85

Figure 5‑18: Changed example 1 86

Figure 5‑19: Changed example 2 86

Figure 5‑20: Changed example 3 86

Figure 5‑21: Failed example 1 86

Figure 5‑22: Missing example 1 87

Figure 5‑23: New example 1 87

Figure 6‑1: PLC added 89

Figure 6‑2: Wago overview PLC 90

Figure 6‑3: NMEA sensorlist example 91

Figure 6‑4: NMEA example 1 91

Figure 6‑5: NMEA example 2 92

Figure 6‑6: Devicelist WatchIO 93

Figure 6‑7: Sensorlist WatchIO 94

Figure 14‑9: Naiad manual 96

Figure 14‑10: Example CanOpen 96

# Tables

Table 2‑1: Devicelist Columns 17

Table 2‑2: Protocol Options 19

Table 2‑3: Interface Options 19

Table 2‑4: Type Options 20

Table 2‑5: Device options 20

Table 2‑6: IP Ranges 21

Table 3‑1: Sensorlist columns 45

Table 3‑2: Sensor Type mode Read 46

Table 3‑3: Sensor Type mode Write 47

Table 3‑4: Unit Type 49

Table 3‑5: (Default) Unit options 60

Table 5‑1: Import Result fields 85

Table 6‑1: Devicelist WatchIO 94

Table 6‑2: Sensorlist WatchIO 95

**NOTICE**

**This document contains proprietary information.**

**No part of this document may be photocopied, reproduced or translated into another language without the prior written consent of   
Imtech B.V.**

References

Not applicable.

Abbreviations list

AM(C)S Alarm Monitoring (and Control) System

COM Communication

CPU Central Processing Unit

DAP Duty Alarm panel

DM Dead Man’s

ECR Engine Control Room

GEA General Engineers Alarm

GPS Global Positioning System

GRP Group

ID Identification

I/O Input/Output

LAN Local Area Network

LED Light Emitting Diode

LPU Local Processing Unit

MAC Media Access Control

NMEA National Marine Electronics Association

OWS Operator Work Station

SMS Short Message Service

TCP/IP Transmission Control Protocol/ Internet Protocol

TFT Thin Film Transistor

USB Universal Serial Bus

Safety instructions

* This section provides only a summary of the safety requirements and notes in the following sections. To protect your health and prevent damage to the AM(C)S equipment or vessel, it is essential to read and carefully follow the safety instructions.*

The indications NOTE, CAUTION and WARNING have the following significance:

* NOTE:  
An operating procedure, practice or condition etc., which it is important to emphasize.*

* CAUTION:*

*An operating procedure, practise or condition etc., which, if not strictly observed, may damage AM(C)S equipment or crash NavVision software.*

* WARNING:*

*An operating procedure, practise or condition etc., which, if not carefully observed may result in personal injury or damage to the vessel.*

Revision history

Revisions issued since publication.

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Revision** | **Reason** |
| 2.1.1 | September 09, 2014 | New version | Decimus update |

# What is the Sensorlist

## Introduction

The sensorlist is a validated description of the total FT system. Everything that is connected, whether it is a sensor, a serial connection, an engine etc. is represented in the sensorlist. The sensorlist is the start of where we build the topology, the network and all connected devices. Once imported, the sensorlist will be updated into a “sensorlist\_generated” and a “devicelist\_generated” file that is pretty much automated. Missing files, wrong connections, etc. will be highlighted or even changed. New connections will be highlighted and fails will be highlighted as well.

If the sensorlist is kept well up to date all changes to the system can be made from within the sensorlist. It will be your ultimate tool to easily change your setup, change names or even add new sensors or complete devices. Also the sensorlist is a nice tool to troubleshoot the system. It makes it easy to find double connections, wrong terminations, strange values etc.

Learn how to work with the sensorlist and you’ve got half the job done.

## Excel

The sensorlist is composed in Microsoft Excel. Some knowledge on working with Excel is absolutely necessary. That part of training lies beyond the scope of this manual. We refer to books and courses for Microsoft excel to learn the basics.

It is enough to have basic knowledge because the sensorlist itself is merely a form you have to fill in with the appropriate data. The sensorlist exists of two parts (tabs) which are the “devicelist” and the “sensorlist”. The devicelist (see Figure 1‑1) contains all the devices where he system consists of and is namely an enumeration of the topology of the system. The sensorlist (see Figure 1‑2) on the other hand is a list of all the I/O attached to the system together with the necessary information for the connection.

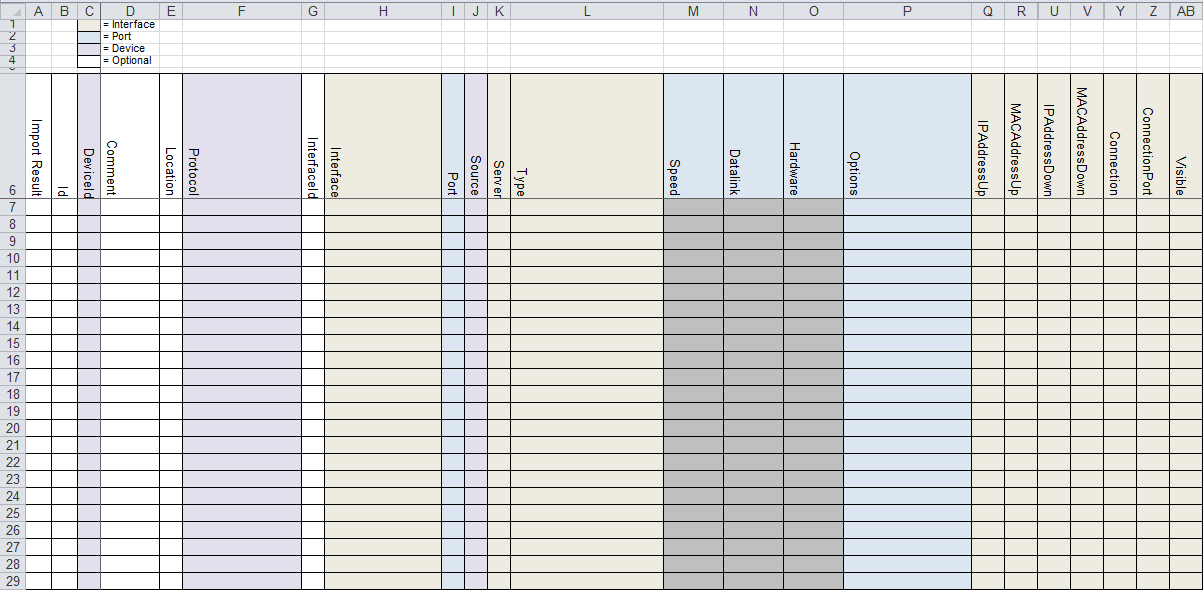


Figure 1‑1: devicelist

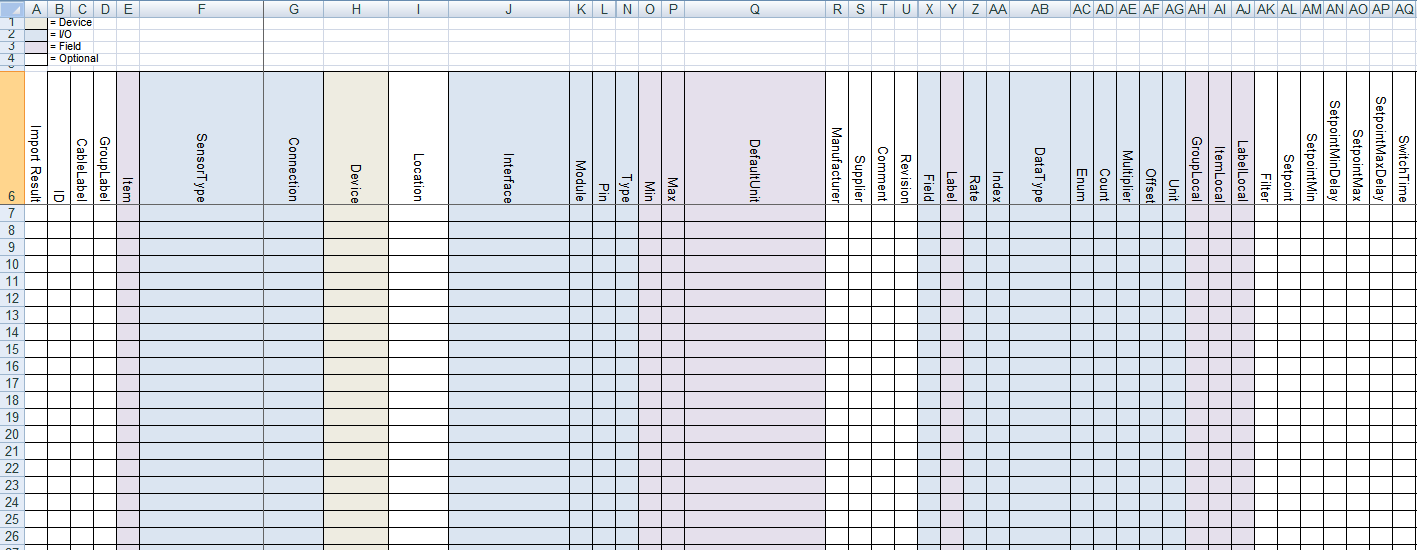


Figure 1‑2: Sensorlist

It goes unsaid that for filling in the sensorlist properly, you need to make sure that you have all the appropriate data form the shipyard available.

*: Always ask Imtech for the latest sensorlist. The sensorlist changes with upg**rades when new features or protocols are added.*

## Saving and naming

For working with the sensorlist always make sure that you use the latest version of Microsoft Excel. At this moment this is Microsoft Excel 2010. Although it is also possible to work with an earlier version, we will use this version as an example in this manual.

While working on the sensorlist, make sure that you save your work regularly to prevent loss of data. Goto startbutton>save as>Excel workfolder (See Figure 1‑3).

Make sure you choose the right folder to save to and the right format (in this case .xlsx) and save the sensorlist with a distinctive name. When working on ship A you can use for example “sensorlist\_shipA\_v1.1.xlsx”. When renewing or changing the sensorlist you can add a new version number to distinguish the different versions (i.e. “sensorlist\_shipA\_v1.2”).

When working on the ship or on the original configuration, make sure that, together with the newest sensorlist, you take a backup of the complete NavVision folder.

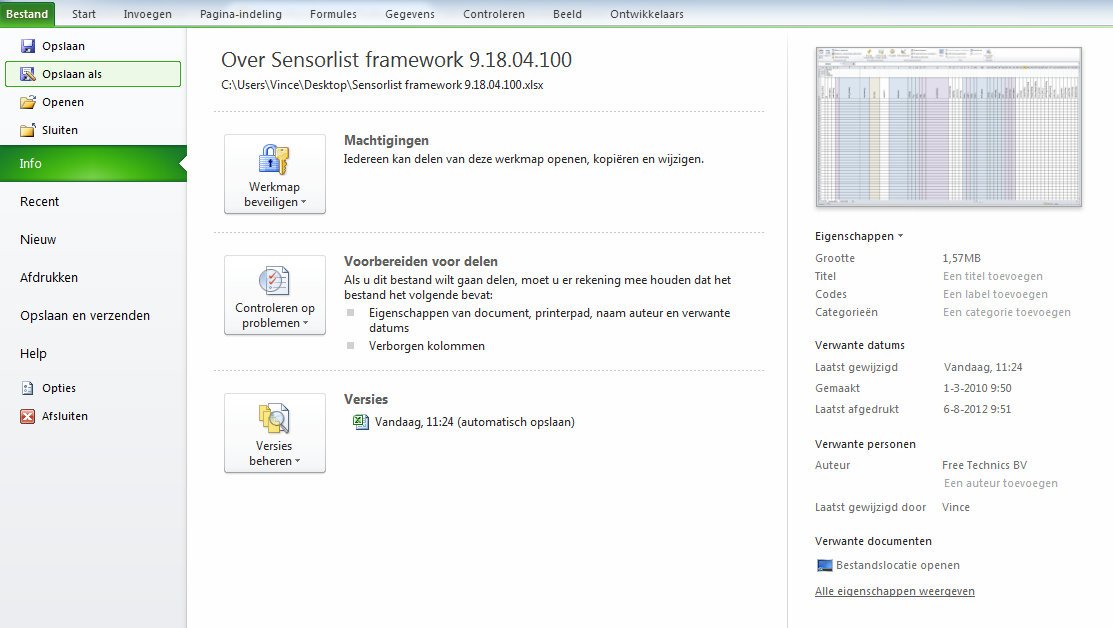


Figure 1‑3: Excel saving

## Saving as sensorlist for import

When saving the sensorlist to be used as import-file you need to do two things:

After choosing “save as” you go to the drop-down menu for the file type and choose

“Excel 97-2003-workfolder (\*.xls)” (see Figure 1‑4) while this is the supported format

for importing a sensorlist.

Save the sensorlist as “sensorlist.xls” no capitals.

Save this “sensorlist.xls” in the same folder as the sensorlist.xlsx you derived it from, so you can always check what has been changed and/or you can get back to previous versions.

For importing the sensorlist into NavVision we refer you to Chapter 4.

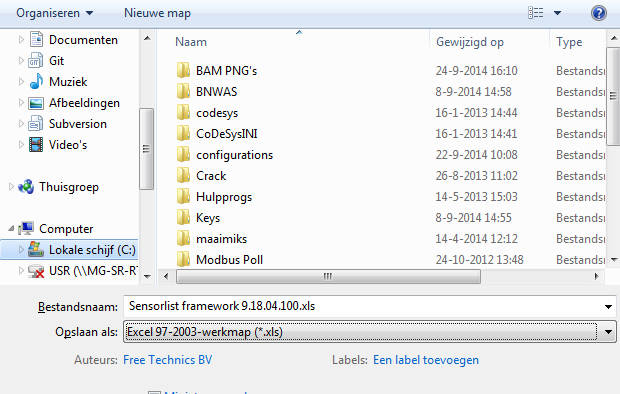


Figure 1‑4: saving as sensorlist.xls

# Devicelist

The devicelist is the part of the sensorlist that contains all the devices that are connected to NavVision together with all the specific data concerning that connection. When opening a sensorlist framework you will see 2 tabs from which you will have to choose the tab named “devicelist”. (see Figure 1‑1).

## Introduction

The devicelist is separated in different columns which need to be filled with the right data. A few of the columns are optional and merely there for you to put your own comment. These columns are white. The other columns are almost all necessary for the proper working of the system and are colored differently. These colors belong to the different groups which can be divided into interface, port and device. Columns with the same color belong to the same group.

By defining all the devices the right way in the devicelist you will get a properly closed network once you import the sensorlist into the system. To do so you need to make a plan on how you need the network to be applied, a list of all the devices and a list of how everything will be connected. To make it visual it is best to make a single-line drawing of the topology for reference (see Figure 2‑1).

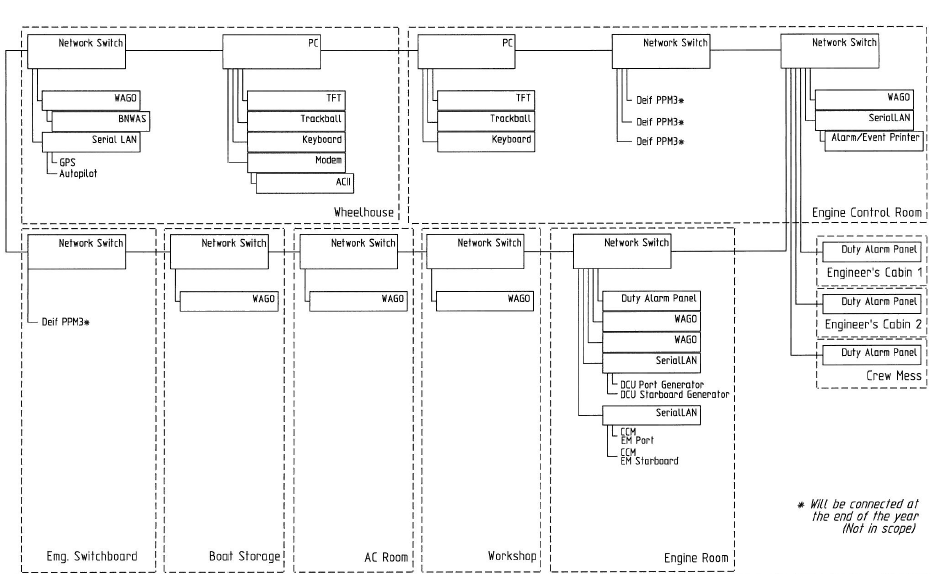


Figure 2‑1: single line drawing

## Columns

The columns in the devicelist are labelled in the first row. The fields underneath can be filled with free text or have a drop-down menu where you can choose a tag. These tags are mandatory and the devicelist won’t except tags that are not in the list for these columns.

The following columns are in the devicelist:

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| Import Result | Text | “Checking” value generated by NavVision |
| ID | Text | Any given ID you want or need. |
| DeviceId | Text | Identification of the device where the sensor/control or serial device is connected to. This text should be unique for each NavVision device. The text is case sensitive |
| Comment | Text | Freely to add comment |
| Location | Text | Identification of the substation where the sensor/control is connected to in the FT NavVision® system. (i.e. ER or WH) |
| Protocol | Select  (Index) | The protocol used for serial connections. (for options see Table 2‑2) |
| InterfaceId | Text | Free interface Id description for use in the “System Layout” |
| Interface | Text  (Index) | Choose the appropriate interface to distinguish the different interfaces in the system (for options see Table 2‑3) |
| Port | Value  (Index) | Port number on the FT NavVision® interface. For MOXA serial servers it’s 1 or 2. On a WAGO it’s always 1. |
| Source | Value  (Index) | Identification of multiple devices on a bus protocol. Used for example for Mod bus (ID byte) and CAN bus (SA byte). Default address is 1. |
| Server | Text | In some cases (like with OPC and WatchIO), you need to specify a server name. |
| Type | Text  (Index) | defines the type of module used to read/control the I/O. (for options see Table 2‑4) |
| Speed | Value  (Index) | The Baudrate the device is communicating with. See devices manual for the appropriate speed. |
| Datalink | Value  (Index) | Defines the parity, databits and stopbit. See devices manual for appropriate settings |
| Hardware | Value  (Index) | Serial communication protocol |
| Options | Text  (comma separated) | Divers special settings for various devices. See devices manual for need of these special demands. (for options see Table 2‑5). |
| IPAddressUp | IP-address | IP address of the FT NavVision® interface that’s connected to the device or sensor/control. Up-side (for explanation see Chapter 2.3). |
| MACAddressUp | MAC-address | MAC address of the FT NavVision® interface that’s connected to the device or sensor/control. Up-side (for explanation see Chapter 2.4). |
| IPAddressDown | IP-address | IP address of the FT NavVision® interface that’s connected to the device or sensor/control. Down-side (for explanation see Chapter 2.3). |
| MACAddressDown | MAC-address | MAC address of the FT NavVision® interface that’s connected to the device or sensor/control. Down-side (for explanation see Chapter 2.4). |
| Connection | Text | Specify the device (see first column) to which this device is connected |
| Connection Port | Value | Specify the port on the device where this device is connected to |
| Visible | Yes/No | Non mandatory field to tell NavVision if the node needs to be visible in the network topology. |

Table 2‑1: Devicelist Columns

|  |  |  |
| --- | --- | --- |
| Option | Connection | Description |
| Adam | Serial | Advantech 4500/5000 series |
| AIS | Serial | AIS Data over Nmea |
| Algodue | Serial | Algodue AC monitoring module |
| Asea | Serial | Asea Shore converters |
| AutoAnchor601 | Serial | Chaincounter |
| BMV501 | Serial | Victron battery monitoring modules |
| BMV602 | Serial | Victron battery monitoring modules |
| BTM1 | Serial | Mastervolt battery monitoring modules |
| Camera | IP | IP Cameras |
| Can | I7540D | CAN bus |
| CanOpen | IP | CanOpen protocol |
| Cat | Serial | Caterpillar CAT-Link protocol. Link via CCM |
| CF Smartview | Serial | Broadband |
| Crompton | Serial | Crompton AC monitoring module |
| DssKeypad | Serial | CAN-based keypad |
| EM4000 | Serial | ELEQ AC monitoring module |
| EmpirBus | Serial | power supply systems |
| Frigomar\_626C | Serial | Airconditioning |
| FSI\_2DACM | Serial | Current measurement sensors from Falmouth Scientific Instruments |
| Generic | Serial | Gen-set |
| Gensys | Serial | GenSYS power management system (PMS) monitoring |
| J1708 | Autotap | SAE J1708 |
| J1939 | I7540D | SAE J1939 |
| KiloPakIguard | Serial | Kilopak I-Guard Generators |
| Littau Anchor | Serial | Anchoring |
| Lutron | Serial | Lutron Light system |
| MalinDraught | Serial | Draft System |
| Masterbus Modbus | Serial | Mastervolt charger/inverter modules through Modbus |
| Mastervolt | Serial | Mastervolt charger/inverter modules |
| Mitsubishi\_DMS\_II | Serial |  |
| ModBus | Serial/IP | Modbus ASCII/RTU Serial or TCP/IP |
| ModBus Slave | Serial/IP | Modbus ASCII/RTU Serial or TCP/IP |
| MPC30 | Serial | Inkjet printer |
| MTU | Serial | MTU MCS-5 system. Connections to be made through LOP, PIM or PCS. |
| MVECP | Serial | PaxMAN Engine Control Unit |
| Nke | Serial | NKE Navigation Instruments and Autopilots |
| Nmea | Serial | NMEA 183 |
| Nmea Mecmar | Serial | Proprietary NMEA |
| Nmea Nacos | Serial | Proprietary NMEA |
| Nmea Quantum | Serial | Proprietary NMEA |
| Nmea2000 | I7540D | NMEA 2000 over CAN. |
| PC | IP | Server or Client PC failsafe Client or DAP |
| PPM3 | Serial | Deif power management system (PMS) monitoring |
| Printer | Serial | Printer |
| Sae | I7540D | SAE |
| SD41 | Serial |  |
| SMS | Serial | SMS Module (Tango blackbox modem) |
| Sounder | Serial | Black box video sounder |
| Switch | IP | Switch |
| SygoDraft | Serial | Sygo Draft systems |
| TMA4S | Serial | Tank Gauging System |
| Vaisala\_CL31 | Serial | Vaisala cloud detection sensor |
| Vaisala\_LT31 | Serial | Vaisala LT series visibility sensor |
| Vaisala\_PTB330 | Serial | Vaisala Digital Barometer |
| Vaisala\_PW | Serial | Vaisala PW series visibility sensor |
| VDR | Serial | VDR output connection (NMEA 183 based) |
| Victron | Serial | Victron charger/inverter modules |
| VictronVEBus | Serial | Victron BUS |
| VisiplexPaging | Serial | Alarm paging system |
| Wago | IP | Wago |
| WatchIO | Internal | WatchIO |

Table 2‑2: Protocol Options

|  |  |
| --- | --- |
| Interface | Description |
| Camera 01, Camera 02, etc. | Define the different IP cameras on the network. Do not use the same Camera twice. |
| CAN 01, CAN 02, etc. | Use a separate interface-ID for each Canbus device. If you, for example, have two I7540D devices, you choose CAN 01 for the first and CAN 02 for the second |
| Local Serial | Choose this interface setting for a serial connection that is directly connected to the server. |
| Network Serial 01, Network Serial 02, etc. | Network Serial devices are devices like the MOXA that are used as an interface between serial to LAN. Each interface needs a distinctive interface. More ports on the same device will get the same interface |
| OPC Client | OPC client |
| PC 01, PC 02, etc. | The main workstations will act as server. Each server gets its own interface |
| Printer | When a printer is connected |
| Remote Monitoring | Connection NavVision to an external PC |
| Settings | Use if the line contains a setting for NavVision |
| Switch 01, Switch 02, etc. | Interface for network switches. Although the switches have multiple ports, you only use one interface for each switch. |
| Wago 01, Wago 02, etc. | When a Wago is connected, choose Wago as interface. Each Wago gets its own interface |
| WatchIO | Special connection type for WatchIO |

Table 2‑3: Interface Options

|  |  |
| --- | --- |
| Type | Description |
| Axis 241Q | Axis IP camera interface |
| Client Fail Safe | Definition of PC type |
| Dual LAN Switch | Switch to connect different devices, Dual LAN type |
| GW003 | CANbus to serial interface |
| ICPdas i7540D | CANbus to serial interface |
| ModBus TCP/IP | Modbus over TCP/IP |
| ModBus TCP/IP Slave | Modbus over TCP/IP slave |
| Moxa UC-711X | Serial to Ethernet interface |
| Printer | Printer |
| Serial TCP/IP Client | TCP/IP client over serial connection |
| Serial TCP/IP Server | TCP/IP server over serial connection |
| Serial UDP/IP Client | UDP/IP client over serial connection |
| Serial UDP/IP Broadcast | Typical broadcast over UDP/IP |
| Server | Definition of PC type |
| Switch | Switch to connect different devices |
| Telnet | Telnet |
| V-Linx ESR-904 | Serial to Ethernet interface |
| Wago | PLC |
| Wago 750-352 | PLC type specific |
| Wago 750-881 | PLC type specific |
| Wago 750-882 | PLC type specific |

Table 2‑4: Type Options

|  |  |
| --- | --- |
| Device | Description |
| AlarmDataLoss | Gives an alarm on loss of data on the specific port. Works only when the interface have had a connection before. |
| DTR | When Data Terminal Ready needs to be set High |
| dtr | When Data Terminal Ready needs to be set Low |
| RTS | When Request to send needs to be set High |
| rts | When Request to send needs to be set Low |
| RTU | Sets the port to RTU |
| ASCII | Sets the port to ASCII |
| Interval=”125” | Interval between messages in milliseconds |
| Timeout=”500” | Timeout-time of message in milliseconds |
| MaxBitCount=”2000” | Same as MaxWordCount but then in Bits |
| MaxWordCount= | Some Modbus protocols can read only an x-amount of registers at one time. While FT works with the Modbus standard of 123 registers, you need to limit the max value of words that FT is questioning. For Heinen Hopman for example it is “MaxWordCount=10” |
| Nodiff | Sent data even if no changes detected |
| NoHoles | Some Modbus protocols can’t handle it when there are a lot of unused registers between the different calls. With the option “NoHoles” all the registers that are not used will be ignored. |
| IgnoreSource | Ignore source-ID in NMEA |
| KeepAlive | Especially for H&H interfaces, but can be used in other Modbus protocols. When a Modbus call doesn’t get an answer in the predefined time, it will keep the question alive until answered. |
| OutputFirst | Especially for H&H interfaces, but can be used in other Modbus protocols. If a request is send (Modbus function 6) it will be handled before other questions |
| MSBFirst | Set reading of Most Significant Bit First |
| LSBFirst | Set reading of Least Significant Bit First |
| MSWFirst | Set reading of Most Significant Word First |
| LSWFirst | Set reading of Least Significant Word First |

Table 2‑5: Device options

## IP-addresses

### Introduction

At Imtech we use a specific set of IP-addresses for our connections. We use the 172.16.x.x range for the IO-side of our system and the 172.17.x.x range for the next ring. If there are more rings connected than these two we go on with 172.18.x.x etc. As you can find in the “installation and commissioning manual” we use also specific ranges for the different devices and interfaces (see Table 2‑6).

|  |  |
| --- | --- |
| **Detail** | **IP-Address** |
| PC I/O | 172.16.x.x (172.16.24.35 for key number 2435) |
| PC I/O next ring | 172.17.x.x (172.17.24.35 for key number 2435) |
| Duty Alarm Panels  (DAP) | Using range x.x.1.8y  Depending on the network connected, this will result in:  DAP 1: 172.16.1.81  DAP 2: 172.16.1.82  DAP 3: 172.16.1.83 |
| Serial LAN servers | Using range 172.16.1.4x (attached to I/O subnet 172.16) INT 1: 172.16.1.41 INT 2: 172.16.1.42 INT 3: 172.16.1.43 |
| Wago | Using range 172.16.1.9x (attached to I/O subnet 172.16) Wago substation 1: 172.16.1.91 Wago substation 2: 172.16.1.92 Wago substation 3: 172.16.1.93 |
| CAN-Interface | Using range 172.16.1.3x (attached to I/O subnet 172.16) CAN interface 1: 172.16.1.31 CAN interface 2: 172.16.1.32 CAN interface 3: 172.16.1.33 |
| Axis or IP camera | Using range 172.16.1.24x (attached to I/O subnet 172.16) Axis cam server 1: 172.16.1.241 Axis cam server 2: 172.16.1.242 Axis cam server 3: 172.16.1.243 |

Table 2‑6: IP Ranges

We work from the single line drawing to make it possible to get all the IP-addresses to the right line in the devicelist. Also it is wise to start with building the topology of the single line drawing in to the devicelist. This way you will get closed rings.

### IPAddressUp- IPAddressDown

In the single line drawing you best number all the connections upfront so you minimise the mistakes. Say that the IPAddressUp is number 1 and the IPAddressDown is number 2 (try to make the Up-address to go to the i/o side of the system). The drawing will look like the following:

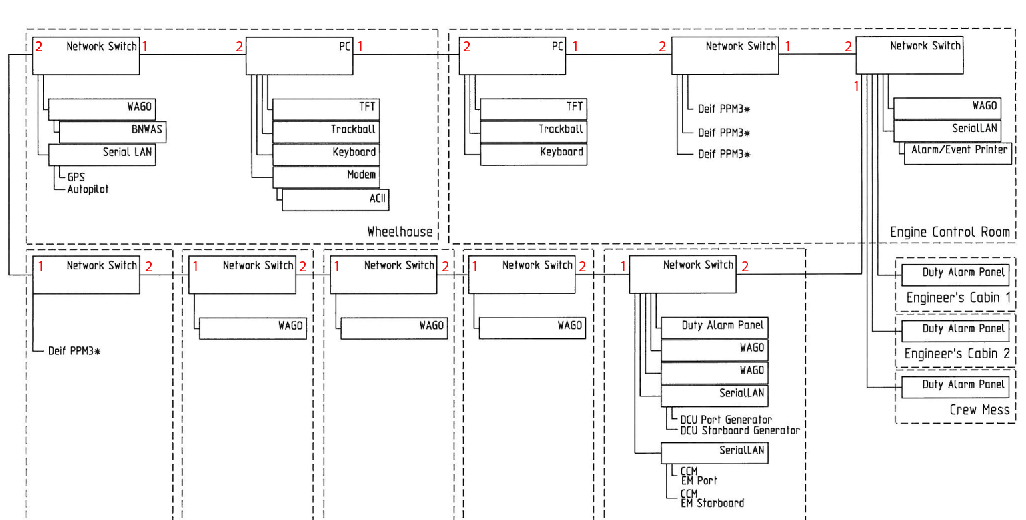


Figure 2‑2: numbering the drawing

As you can see we have numbered all the devices with the numbers 1 and 2. Now number 1 is the “IPAddressUp” and number 2 is the “IPAddressDown”. So, for example. for the PC (let’s assume it has key number 3035) the number 1 side , in the sensorlist IPAddressUp, will be 172.16.30.35. the number 2 side, in the devicelist the IPAddressDown, will be (in this case) 172.17.30.35. You’ll notice that de down-side is considered as another ring and will get another IP-range.

While the Switches do not have an IP-address they need not have one of the above mentioned IP addresses assigned. More on how to build that in to the devicelist in chapter

2.5.11

The interfaces such as the Wago, the SerialLan etc. will get their own IP address as well as a port connection (also see Chapter 2.5.11)

## Mac addresses

### Introduction

To be identified in a network it is sometimes needed that the MAC address is available to distinguish two or more of the same devices. The MAC address is a unique number that is only conjugated to one device. So if there are two or more Moxa’s on a network, we need to separate them with their distinct MAC number. For these devices you need to fill in the MAC address in the devicelist.

## How to implement this in the devicelist

### Introduction

To implement all the devices in the devicelist we will now give an example on the basis of the single-line drawing we presented earlier. We will try to give you a method that is easy to use, yet with the least possibilities to make any mistakes. Once you get familiar with the devicelist, you can derive your own method of working. The ultimate goal of the devicelist is to make the topology connected flawlessly which can be checked in NavVision.

### The devices

As seen earlier a device name is “Identification of the device where the sensor/control or serial device is connected to. This text should be unique for each NavVision device. The text is case sensitive.

So to use it later on in the sensorlist it is mandatory that you give each device a name that is unique. So use names as “GPS” or “Port Engine” or any other name as long as it is descriptive for your device.

As the example single line drawing shows we have two PC’s that are in the topology. For now let’s call them OWS WH and OWS ER. So the first devices that we put into the devicelist are these two. Please remember which one you call OWS WH and which OWS ER (easiest way to do is to write it down in the drawing). Now let’s put them in the devicelist (see Figure 2‑3).

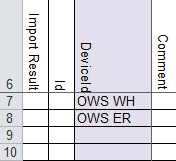


Figure 2‑3: Filling deviceId column 1

Next thing we find in the drawing are a lot of switches. We can number the switches or give them the location as a tag, or even both. That way we know about which switch we are talking. In this case we work with the descriptive name and we get the following:

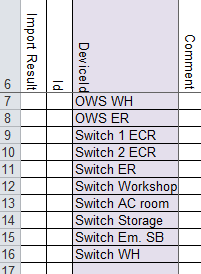


Figure 2‑4: Filling deviceId column 2

Next we take the Wago’s. Let’s also take the descriptive name for that and we get the following:

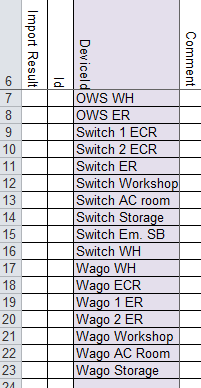


Figure 2‑5: Filling deviceId column 3

Last items are some serial LAN interfaces and some DAP’s (Duty Alarm Panels). Adding these makes the device column complete and gives the following result:

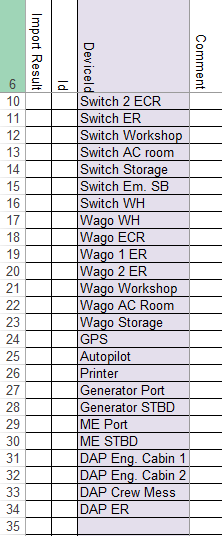


Figure 2‑6: Filling deviceiD column 4

*: Serial Lan has multiple ports as well as some other interfaces. Make sure you put both ports (if in use) in the device column. You can give it a name like “serial 1 ER-1”wich is the first port of the first serialLan interface in the engine room, or you can name it after the sensor or device that is connected to that port (in this case “Port Generator”).*

This completes the filling of the device column. If you compare it to your drawing, you can see that all the devices in the topology are now in the devicelist. Time to move to the next column.

### Comment

We leave this up to you. If you have something you need to remember with a certain device you can keep it here. NavVision doesn’t use this information.

### Location

The location is the Identification of the substation where the sensor/control is connected to in the NavVision system. You can easily get that from the drawing and put it in the Location column. In our case we will get the following (not mandatory):

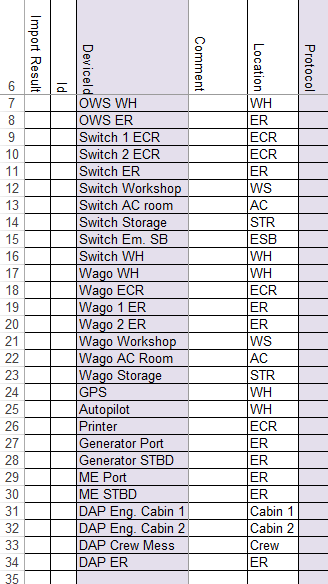
`

Figure 2‑7: Filling location column

### Protocol

To choose the protocol you have a lot of options. To make it easy there is a drop-down box. Just click the appropriate field and look in de drop-down menu if you can find the right protocol (for explanation on the options see Table 2‑2).

The first ones are easy. We have two server and a couple of switches and Wago’s. The protocols for these are quite clear. Fill them in and you get the following:

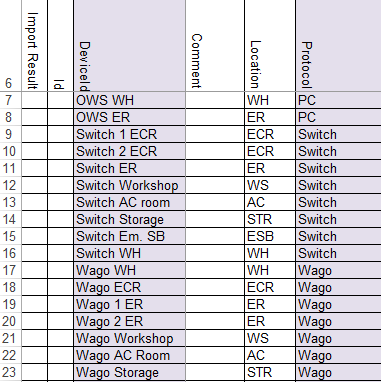


Figure 2‑8: Filling protocol column 1

For the serialLan we need to know what is connected. Look at the drawing and find the right protocol for the connections you see there. For example the GPS and Autopilot are NMEA, the generators and engines are Caterpillar and the printer is a printer. Find these protocols in the drop-down menu and you’ll get the following:

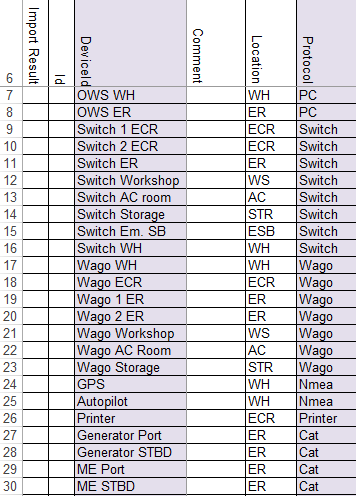


Figure 2‑9: Filling protocol column 2

The DAP’s are PC’s so that ends the filling of the protocol column as follows:



Figure 2‑10: Filling protocol column 3

### Interface

The interface is the name of the actual sort of interface that is used to get the data into the system. This is used to distinguish the same sort of interfaces by a separate number.

It speaks for itself for most of the interfaces. Only notice that for an interface that has multiple ports you need to add the same interface for each port (for options see Table 2‑3).

The result will be as follows:

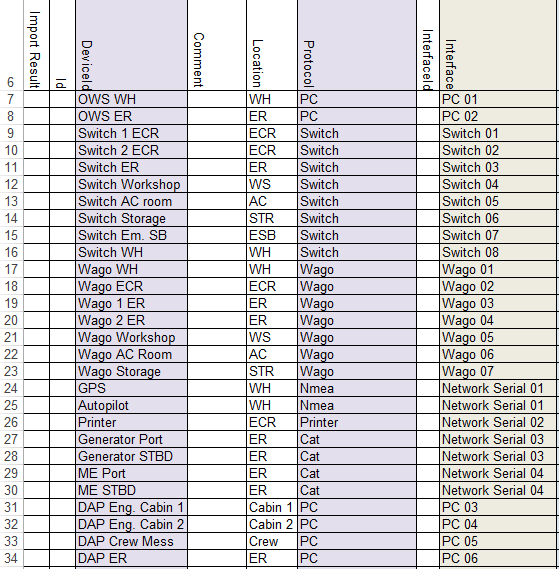


Figure 2‑11: Filling interface column

### Port and Source

The port defines the port on the device that the sensor or IO is connected. So in our case for example we have a port and a stbd engine that are both connected to the same SerialLan. While they are separately connected one will be on port 1 and one will be on port 2. In this “Port” column you can specify this as follows:



Figure 2‑12: Port and Source 1

As you can see, the first port on the Serial Lan gets number 1 and the second port gets number 2. In this case the “source” will stay at number 1.

Default the Port and Source will be “1”

This will result in the following list:

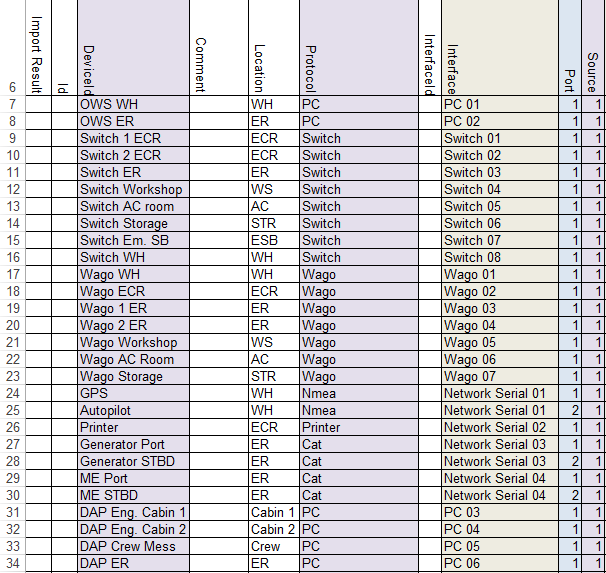


Figure 2‑13 Port and Source 2

*: The source can be as high as 256. When, for instance, you have Modbus/TCP connected through a serialLan it is possible that there are multiple devices connected through the same bus. The ID’s of these devices can also be put in the “source” column. In this case Port will be “1” (there is only one port, the Modbus TCP/IP connection) that the signal is coming in. The source will be all the ID’s on the bus.*

### Type

Defines the type of interface used to read (control) the I/O. (for options see Table 2‑4). As you can tell from the options table it is mostly used when the I/O source is connected to NavVision through some type of interface. This can be SerialLan, TCP/IP (Modbus or Serial) and a few more options.

Also you specify things further in this column. For instance a PC can be a Server, a Failsafe -Client or a Client.

A Wago PLC can be a 750-881, 750-882 or another coupler station.

The interface for SerialLan can be a Moxa or a Vlinx, and so on.

If the interface needs some extra specification, the fields “type, speed, datalink and hardware” will change color (see Figure 2‑14). This way you will know that you have to fill in these fields.

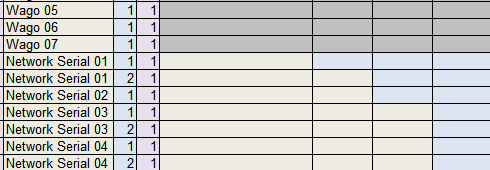


Figure 2‑14: Network Serial colors

In our example, while we do not have any special interfaces, it is quite obvious what to choose. Only some extra information will be needed for the Network Serial. The rest will look like the following:

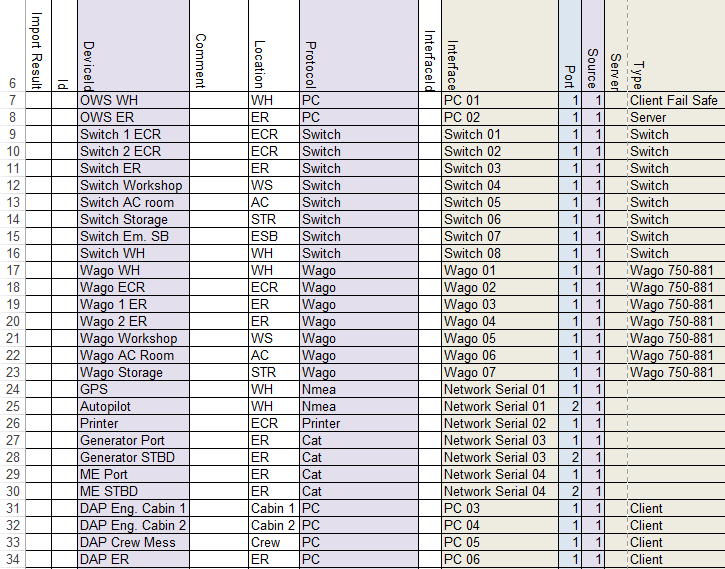


Figure 2‑15: Filling Type column 1

For the Network Serial you will have to look at the Interface in the same way you do as in the rest of the “Type” column. Only in this case it will probably be a serial to Ethernet interface.

You can’t get that directly from the single line drawing, so you need to find out up front. In this example all the network serial interfaces are Moxa’s so we will fill the column with that.

The Type column will be finished as follows:

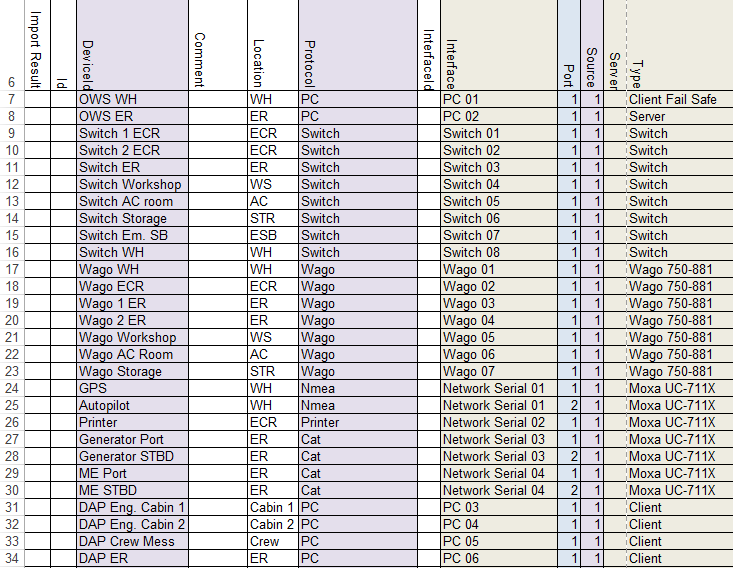


Figure 2‑16: Filling type column 2

### Speed, Datalink and Hardware

The speed, datalink and hardware are figures that you will find in the manuals of the attached devices. If a GPS is connected to the Serial interface, you probably will find a paragraph describing that it is NMEA, at a speed (baudrate) of 4800. None parity, eight data bits and 1 stop bit and that the serial connection is RS232 (see FT Port Connections and Protocols manual for more information).

This is the data that you need for these columns. NavVision will set the ports on the devices, accordingly to what you put here. On the basis of our example drawing we know the following:

* GPS 9600 None 8 1 RS232
* Autopilot 115200 None 8 1 RS232
* Printer 9600 None 8 1 RS232
* Generator 115200 None 8 1 RS485
* Engines 115200 None 8 1 RS485

So with this we can finish these columns and we get the following:

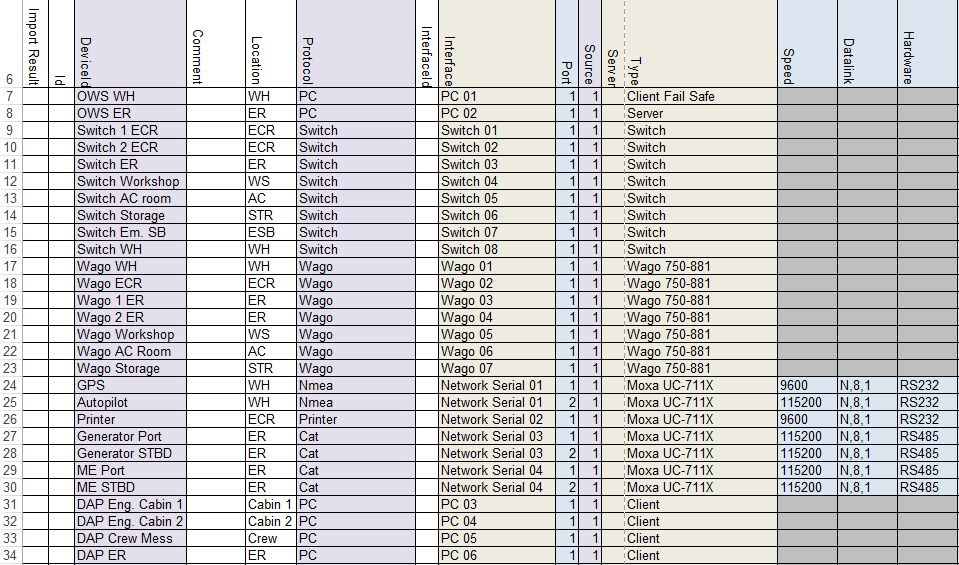


Figure 2‑17: speed, datalink and hardware

### Options

Some devices need some special attention. Mostly because they have another interpretation on protocols, or just that their interpretation deviates from the one that NavVision uses. To make it easier we have made a separate column where we can devine those differences. (for options see Table 2‑5). You can use more options on one device. Just put them in the same cell “comma separated”.

### IP addresses and MAC addresses

As described earlier in Chapters 2.3 and 2.4, we now need to put in the diverse IP addresses and MAC addresses to let the system know how everything is connected. This is essential because the system needs to know where to transfer requests and to make sure that the system is connected the right way. Also the alarms on lost connections depends on these figures.

Let’s look back at the single line drawing that we made (see Figure 2‑1). We already gave the Up-link the number 1 and the Down-link the number 2. These are two separate rings so they will get a separate IP-range. For the Up-link we start with 172.16.x.x.

Based on our drawing we state that the WH-pc has the key 2637 and the ER-pc has the key 2636. In this case we can fill in the x.x with the key number. While we always start in the direction of the I/O we will start at the ER-pc. The ER-pc port 1 (Up-link) will get the IP address from the first IP range ending with the key number. In this case that will be 172.16.26.36. This address we put in the IPAddressUp behind the Server 2 (see Figure 2‑18).

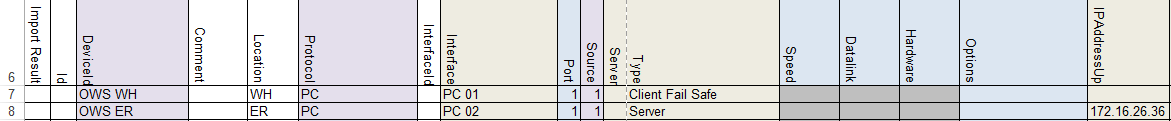


Figure 2‑18: Addresses and connection 1

From port 1 at the ER pc we come at a switch in the ECR. As we use the single line drawing as our reference, it is easier that we fill in the devicelist as we follow the main lead of this drawing instead of filling the IP addresses one by one following the order in the devicelist. This will also give you a good indication on any mistakes that you might have made in the devicelist.

So the next column we now will look at is the “connection” column. Here you need to put the device that the server 2 is connected to. In this case that will be the “Switch 1 ECR” as we named that switch in the “device” column (see Figure 2‑19).

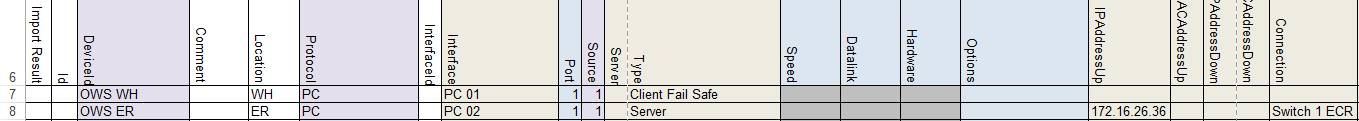


Figure 2‑19: Addresses and connection 2

As we connect from the Up-link from one device to the Down-link of the other device, we now that we connect Server 2 Port 1 to the Switch 1 ECR Port 2. So in column “ConnectionPort” We type “2” (see Figure 2‑20).

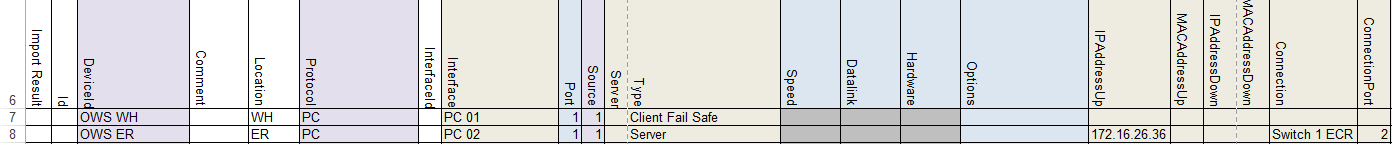


Figure 2‑20: Addresses and connection 3

As we mentioned, we will follow the single line drawing. So the next row that we will process is the row of the “Switch 1 ECR”. A switch doesn’t have an IP address nor a MAC address is needed. This only leaves us to fill in where a switch is connected at. Following the drawing we see that the “Switch 1 ECR” is connected to the “Switch 2 ECR”. So in this case we need to fill in that it is connected to “Switch 2 ECR” at Port 2 (the Down-link of that switch). This way we come to the following:

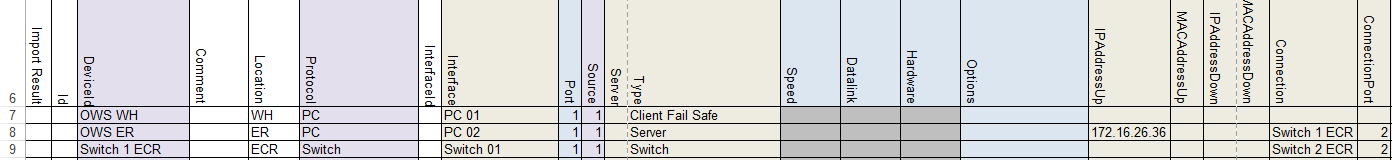


Figure 2‑21: Addresses and connection 4

As you can see in the single line drawing, the main ring is connected through a bunch of switches until you come to the WH server. So the rest of the rows are quite the same. After filling in all the switches you will get to the following:

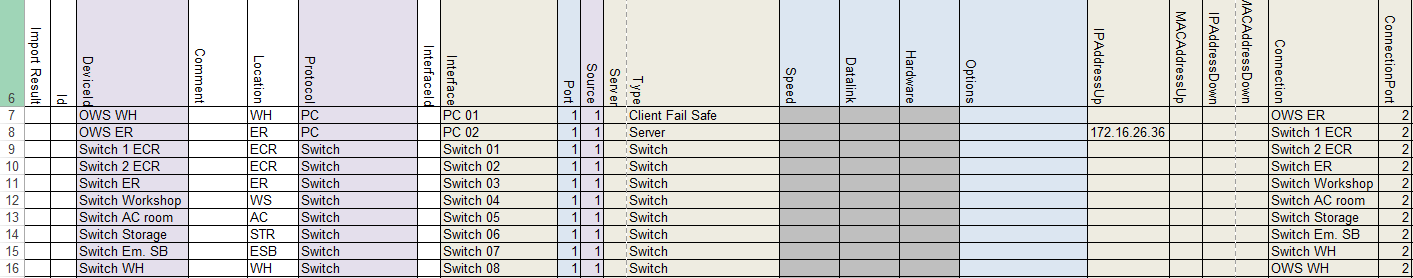


Figure 2‑22: Addresses and connection 5

From the last switch we come to the WH PC or as in the devicelist “OWS WH”. This is the tricky part. As you find in the single line drawing, that switch is connected to Port 2 of the Server. As we mentioned earlier, the Up- and Down-link are two separate rings. These rings need to go round all the way. So the Port 2 of the WH PC has to be in the same IP-range. With the key number of the WH sever being 2637 the IP address of that port will have to be 172.16.26.37. Now while this is Port 2 on the WH PC (the Down-link) you will have to put that IP address in the “IPAddressDown” column. See the following figure:

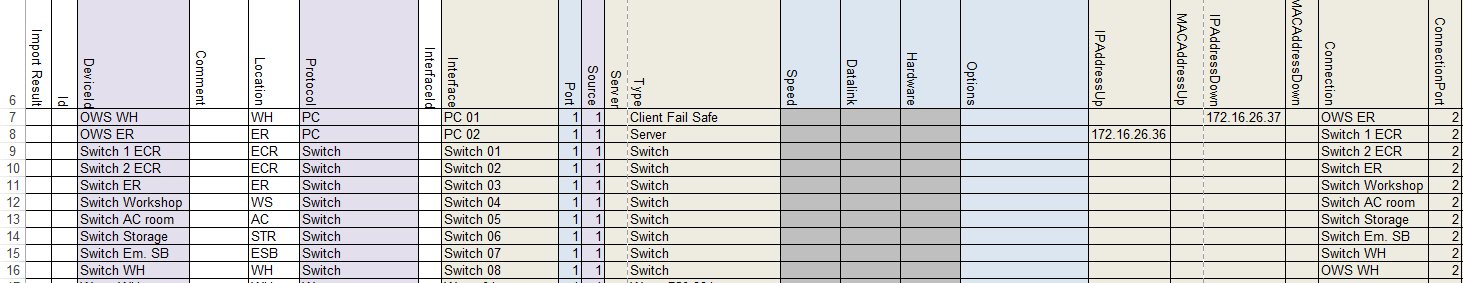


Figure 2‑23: Addresses and connection 6

So now the ring is completed. You can use the devicelist to check if the lines are correct.

Next thing we do is close the ring in the opposite direction. This will be the next IP-range, so 172.17.x.x.

Starting again with the OWS ER the ECR PC We are going to address the Down-link port or Port 2. While it has the key 2636 the IP address for that will be 172.17.26.36 and has to be filled in at the “IPAddressDown” column at the Server 2 row. See following:

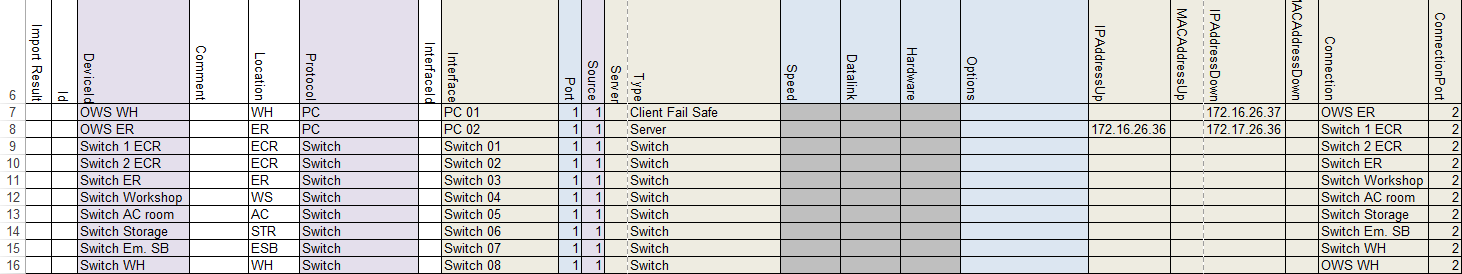


Figure 2‑24: Addresses and connection 7

Concluding that it is connected to Port 1 on the WH PC we can now say that the IP address in the “IPAddressUp” column at the Server 1 row must be 172.17.26.37. See following:

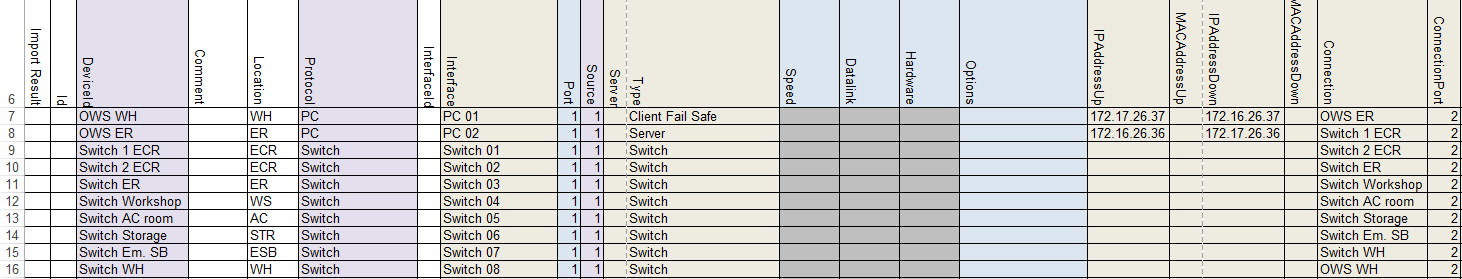


Figure 2‑25: Addresses and connection 8

Now the circle is really connected properly and NavVision can calculate all the connections and make the necessary arrangements for network monitoring and alarming.

#### Other devices

The other devices such as Wago, Network Serial and Clients will not have a Down-link (unless they are in a double-wired systems which goes beyond the scope of this manual), but they do need an IP address, a connection port and sometimes a MAC address.

Let’s start at the Wago. As we can find in Table 2‑6 the Wago use the IP range x.x.1.9y. While the connection lies in the 172.16.x.x range the first Wago will get the address 172.16.1.91, the second Wago will get the address 172.16.1.92 etc. Resulting for our example in the following:

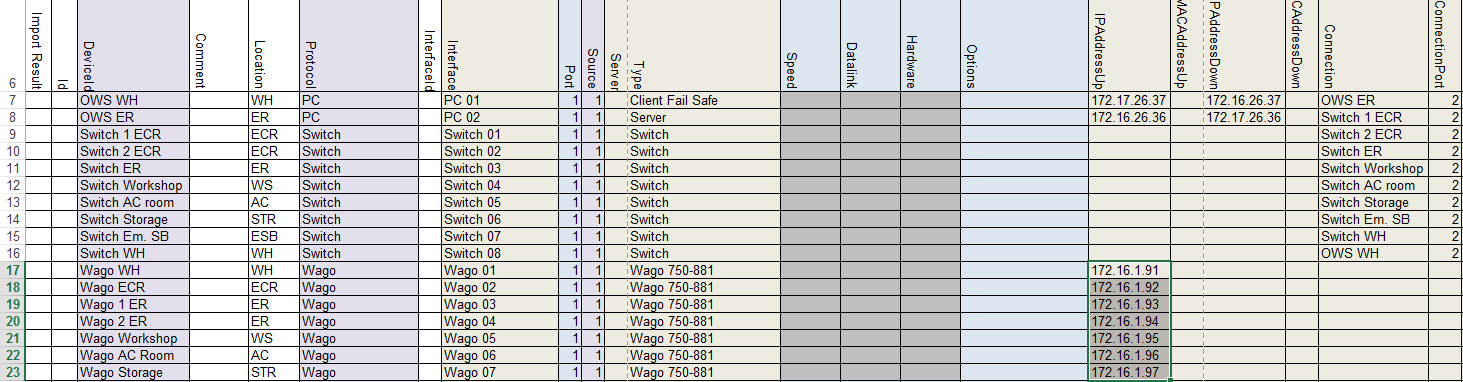


Figure 2‑26: Wago Addresses 1

Wago does need a MAC address but is doesn’t have a Down-link. The MAC address can be found on the Wago PLC itself and will probably lie in the range of 0030DE. Fill in the MAC address in the appropriate row. Also we do need to fill in where they are connected at. For that we again use the single line drawing. As we started earlier at the ECR server we now start again in the ECR and go clockwise to find all the Wago’s. There is one Wago in the ECR (that is why it gets the address 172.16.1.91) and it is connected at the “Switch 2 ECR”. The first free port at the switch is port 3. This results in the following:

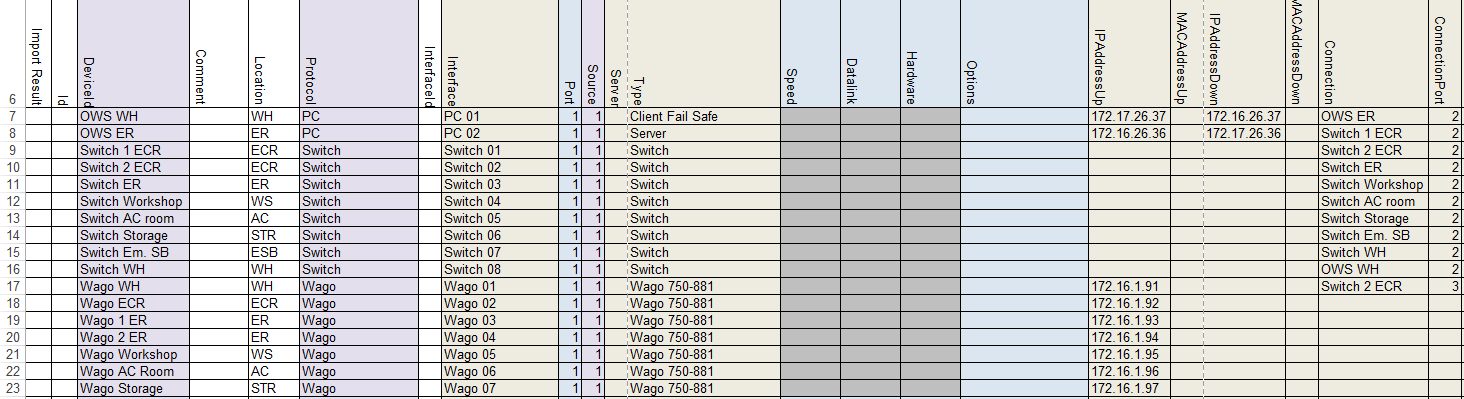


Figure 2‑27: Wago Addresses 2

*: NavVision calculates all the port 1 connections itself. So it is not possible that you find a “1” in the “ConnectionPort” column.*

You can follow this for all the Wago’s. If, like in this example, there are two Wago’s on one switch, than you need to give them separate connection ports. In this case the firs adjacent free ports will be port 3 and port 4.

*: The ports you assign in the devicelist, must be connected exactly the same in the installation. Because NavVision works with multicast, it would be impossible to troubleshoot the system if you mix up the ports.*

The devicelist will be like the next figure after filling in the information (including the MAC addresses) for the Wago’s:

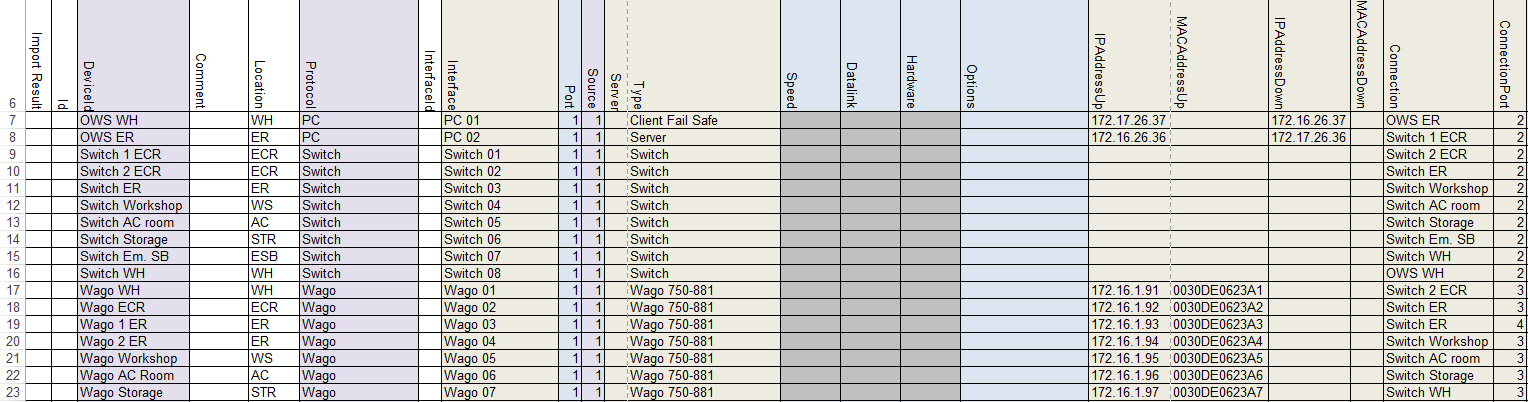


Figure 2‑28: Wago Addresses 3

The Network Serial connections need some special attention. Network Serial Connections can be a variety of interfaces with different approaches in the devicelist. In our example we have Moxa’s as interface. These Moxa’s have an Up-link and you need to specify the MAC address. Also you must specify the connection and the connection port.

According to Table 2‑6 the Moxa falls in the range of x.x.1.4y. so in this case, starting again in the ECR the first Moxa (Serial Network 01) will get the address 172.16.1.41 (as it exists in the 172.16.x.x. range.

*: if you use multiple ports on a Serial Network interface, make sure that you give the same IP address and MAC address to these ports as they are on the same interface.*

The MAC address range of a Moxa will probably be within the 0090E8 range. You’ll find it on the backside of the interface. Put it in the appropriate row.

The first Moxa we find in the ECR with the printer connected to it. This will get the address 172.16.1.41. While only one port is in use, we only need to fill in one row. See the following:



Figure 2‑29: Network Serial addresses 1

Now we can do that for the rest of the Network Serial connections. Be sure that you fill in the same addresses at multiple port connections.

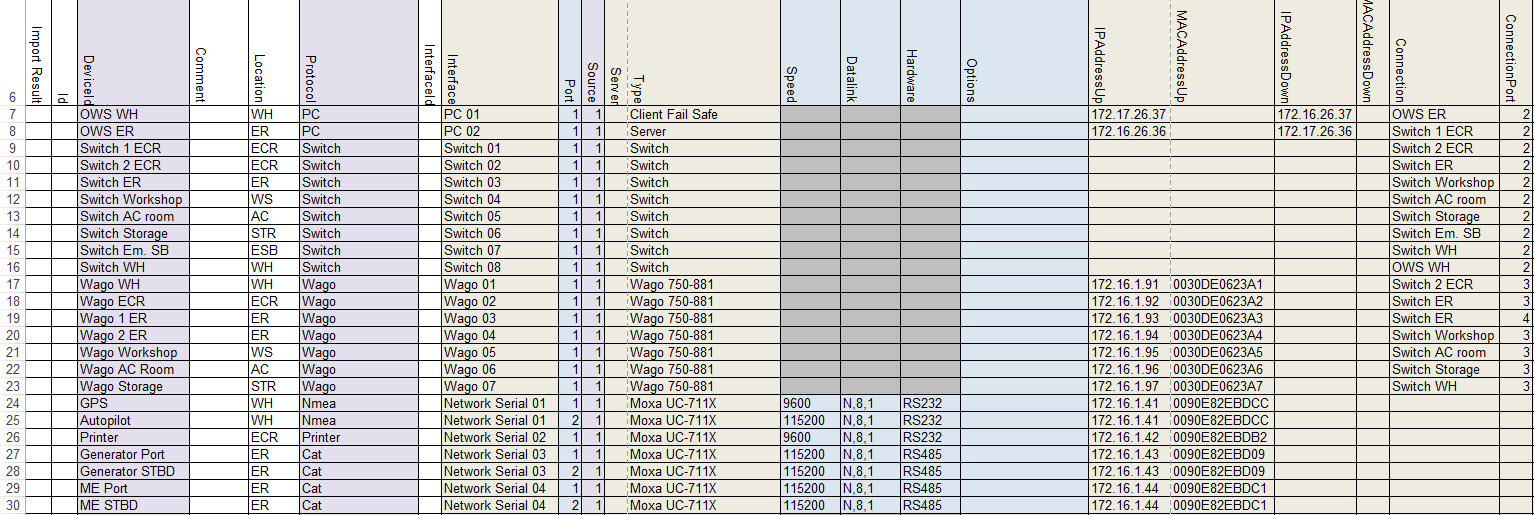


Figure 2‑30: Network Serial addresses 2

Finally assign the Connection and ConnectionPort Where the ConnectionPort will be the first free port on the switch and you will get the following:

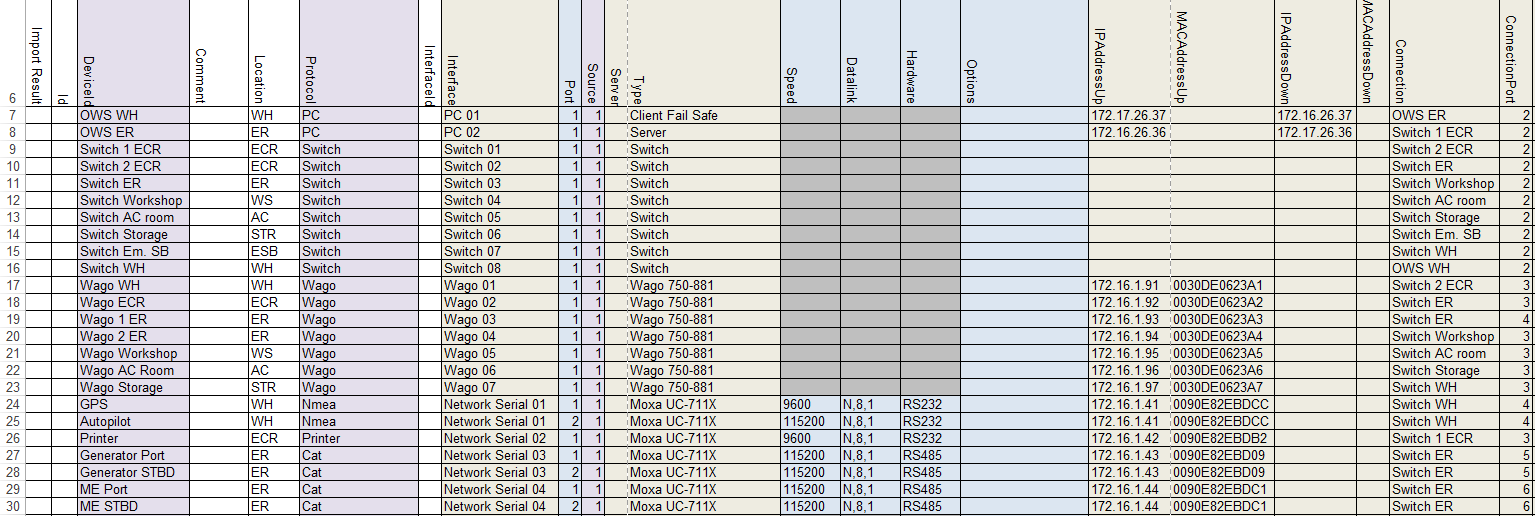


Figure 2‑31: Network Serial addresses 3

*: other Network Serial interfaces can be: ICPdas i7540D, Modbus TCP/IP, Serial TCP/IP and a few others. They mainly work the same way in the devicelist, with the exception that you don’t need a MAC address for TCP/IP.*

Finally we have a few clients in the single line drawing. These are the so called DAP’s (Duty Alarm Panels). As we know from Table 2‑6 the IP range for DAP’s lies within the x.x.1.8y range, where the first one will be x.x.1.81 and so on. While these DAP’s are also in the 172.16.x.x. range, the first DAP will get the address 172.16.1.81.

Also the MAC address is necessary so we put that in the devicelist (DAP’s are in the 00506C range) and also the Connection and ConnectionPort has to be put in. We will finish the devicelist like this:

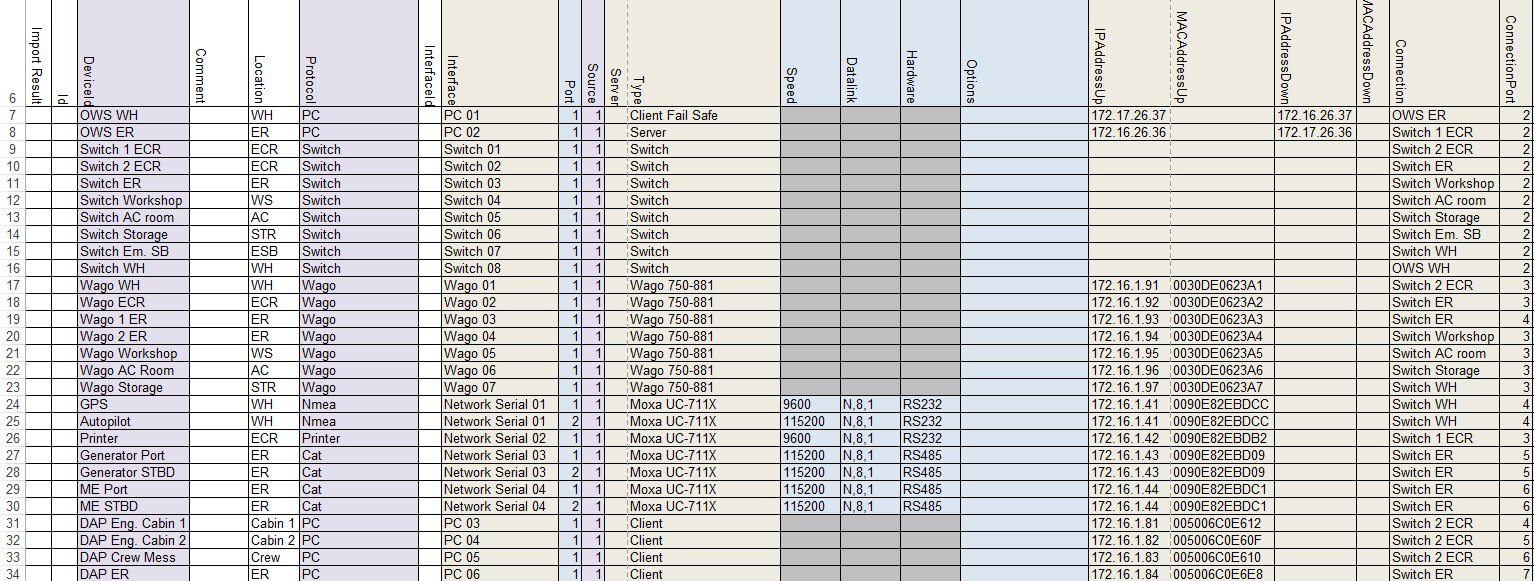


Figure 2‑32: Client addresses

Now the Devicelist is ready you can import it into NavVision to check if it works. We refer to Chapter 4 for further information.

*: We didn’t describe all the possibilities that you can change in the Devicelist, merely the basic ones. Other interfaces or devices can roughly be treated as we described above. If you do find something not working or don’t know how to implement that, please contact Imtech.*

# Sensorlist

## Introduction

In the devicelist we started to list all the devices with their respective interfaces and ports. The sensorlist (tab sensorlist) will break this up even further. It will go from device through I/O to the field that is attached.

Every single I/O that comes into the system will have its own line here in the sensorlist. This is done to control all the incoming data as accurate as possible. Every I/O gets its own Field-ID which will be kept in the database of NavVision. Once the program knows that a certain I/O belongs to a specific field, you can add possibilities to that field to control the I/O. Just as example, you can add min/max values, alarm values, unit types, offsets, inhibits and much more. You can even use the specific I/O in PLC programs, whether internal or external.

With the devices already assigned in the devicelist, you can start out filling the columns in the sensorlist. Be aware that you need all the information on the I/O’s upfront. So for I/O’s on Wago you need to know the sort and type of I/O, but also for protocols such as Modbus, Canbus and other types you will need the right details. Without these details it is almost impossible to make a good sensorlist

## Columns

The sensorlist is also divided in to columns. Some columns are free for your own information, but the colored ones are mainly mandatory. Same as in the devicelist you can find the columns in the sensorlist are labelled in the first row. The fields underneath can be filled with free text or have a drop-down menu where you can choose a tag. These tags are mandatory and the sensorlist won’t accept tags that are not in the list for these columns.

The following columns are in the sensorlist:

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Description** |
| Import Result | Text | For troubleshooting purposes. See Chapter 4.3 |
| ID | Text | A unique ID for the I-O provided by the shipyard or installation company |
| CableLabel | Text | The cable label as labeled in the real installation. Mostly provided by the installation company |
| GroupLabel | Text | Group labels are for dividing I/O into dedicated groups, like Bilge, Fire, Engines etc. |
| Item | Text | The description of the Data Field. Default item text belonging to the Data Field is preferred. The name of the I/O as you want it to appear in the Alarm List. |
| SensorType | Select  (Text) | SensorType defines which subfield or action of the Data Field is set by this value. By default it’s [Standard]. Standard means it’s not defining a subfield or action, but the value of the Data Field itself. (For more options see Table 3‑2 and Table 3‑3). |
| Connection | Select  (NO,NC) | Connection defines the type of connection for digital in- and outputs. Connection is NO by default. If an in- or output is normally closed it’s NC. |
| Device | Select | Identification of the device where the sensor/control or serial device is connected to. This text should be unique for each FT NavVision® device. The text is case sensitive This comes from the devicelist |
| Location | Text | Identification of the substation where the sensor/control is connected to in the FT NavVision® system. Every substation should have a unique text. The text is case sensitive |
| Interface | Select  (text) | Select the type of interface that the data is coming in. For Wago this is divided in the slice’s type-number. For Modbus, Canbus and other protocols it is Serial (Digital/Analog) in or out. |
| Module | Value  (Index) | Module index where the sensor/control data can be found. For CAN bus it is the parameter group number (PGN), for Modbus it is the Modbus mapping and for WAGO it is the slice number. Module 1 for WAGO is the first slice after the 750-626 module. |
| Pin | Value  (Index) | The I/O index on the module for WAGO and the bit offset in the message for serial protocols. (NOTE: The pin index is 1 based) |
| Type | Select | defines the type of module used to read/control the I/O. This is mainly used for WAGO. It can be between 750-400 and 750-612. For Modbus here goes the function code. |
| Min | Value | Minimum instrument value |
| Max | Value | Maximum instrument value |
| DefaultUnit | Select | The default unit used to present this Data Field. (For options see Table 3‑4) |
| Manufacturer | Optional | Manufacturer |
| Supplier | Optional | Supplier |
| Comment | Optional | Comment |
| Revision | Optional | Revision |
| Field | Select  (FTSelect) | The ID of the Field. References to this ID can be found in the file “fieldlist.txt” that is in the root folder of the FT NavVision® software installation after the first time FT NavVision® has been started. |
| Label | Text | The short description of the Data Field when shown in an instrument. Default label text belonging to the Data Field is preferred. The name of the I/O as you want it to appear in an instrument, a value, a button, etc. |
| Rate | Value  (Hz) | Rate describes the number of samples per second of a sensor/control. This is defined by the protocol. Leave empty. |
| Index | Value  (Index) | Index defines when this Data Field Definition [DFD] is valid. The Index column can only be used in combination with a Data Field Definition [DFD] that has the SensorType set to Index and is in the same message as this DFD. Default is empty. |
| DataType | Select  (Unsigned, Signed, Bool, Enum, Float) | DataType is used to define the type of value on serial protocols. For analogue values it’s Float, Signed or Unsigned. For digital values it’s Bool. For enumerations this is Enum. See Enum column. |
| Enum | Value  (Index) | Enum is the index value where the received value should compare to, to switch the Data Field on. If the value is not equal to the Enum index the Data Field is switched off. |
| Count | Value  (Count) | Count is the number of bits starting from the pin index. For a digital value it’s typically 1 with a pin index between 1 and 16 and for analog values it’s for example for Mod bus typically 16 with pin index 1. |
| Multiplier | Value | Multiplier defines the factor between the sensor/control value and the real value.  For inputs/read:  *value = sensor value \* multiplier + offset*  For outputs/write:  *sensor value = (value – offset) / multiplier* |
| Offset | Value | Offset defines the offset between the sensor/control value and the real value. See Multiplier column. |
| Unit | Select | The Unit in which the sensor/control value is received or send. (See Table 3‑4) |
| GroupLocal | Text | Local language text (see chapter 11.1.14 Software installation and commissioning manual 1.9) |
| ItemLocal | Text | Local language text (see chapter 11.1.14 Software installation and commissioning manual 1.9) |
| LabelLocal | Text | Local language text (seechapter 11.1.14 Software installation and commissioning manual 1.9) |
| Filter | Value  (Seconds) | The filter used in the instruments for this Data Field to eliminate short spikes in measurements. Default is 1 second. Maximum is 10 seconds |
| SetpointMin | Optional | SetpointMin |
| SetpointMax | Optional | SetpointMin |
| SetpointMinDelay | Optional | SetpointMinDelay |
| SetpointMaxDelay | Optional | SetpointMaxDelay |
| SwitchTime | Optional | SwitchTime |
| PulseTime | Optional | PulseTime |
| PersistentRequest | Optional | Set request timout on/off (under min/max in NavVision) |
| ExternalRight | Optional | Read, Write or Read/Write rights |
| Decimals | Optional | Set number of decimals in values. (See also chapter 11.2.2.3 Software installation and commissioning manual 1.9) |
| Log | “Y” or “N” | Defines whether a field will be logged for remote monitoring (see Remote monitoring manual v1.0.2) |
| AlarmSMS | Obsolete | Set if an SMS will be sent at alarm |
| AlarmWAV | Filename | The filename of the sound that will be played over the sound card when this Data Field is in alarm. Default is “alarm.wav”. Files can be found in the “sound” sub folder of the FT NavVision® software installation |
| WarningLow | Value  (in “Unit”) | The threshold for the low alarm. Empty is off |
| WarningHigh | Value  (in “Unit”) | The threshold for the high alarm. Empty is off |
| WarningDelay | Value  (Seconds) | The delay for the low and high alarms |
| WarningGroup | Select | The ID of the alarm group that the low and high alarms are assigned to. References to this ID can be found in the file “fieldlist.txt” |
| WarningAction | Text | The action an operator should take when a low or high alarm occurs. |
| CriticalLow | Value  (in “Unit”) | The threshold for the too low alarm. Empty is off |
| CriticalHigh | Value  (in “Unit”) | The threshold for the too high alarm. Empty is off |
| CriticalDelay | Value  (Seconds) | The delay for the too low and too high alarms |
| CriticalGroup | Select | The ID of the alarm group that the too low and too high alarms are assigned to. References to this ID can be found in the file “fieldlist.txt” |
| CriticalAction | Text | The action an operator should take when a too low or too high alarm occurs. |
| InhibitAll | Value  (“” or “Y”) | Inhibit all alarms for a specific field. This will show in the alarmlist. Empty is off. |
| InhibitLevels | Value  (“” or “Y”) | Inhibit all Level alarms for a specific field. This will show in the alarmlist. Empty is off. |
| InhibitTimeout | Value  (“” or “Y”) | Inhibit all Timeout alarms for a specific field. This will show in the alarmlist. Empty is off. |
| InhibitNotReady | Value  (“” or “Y”) | Inhibit all NotReady alarms for a specific field. This will show in the alarmlist. Empty is off. |
| InhibitDefect | Value  (“” or “Y”) | Inhibit all Defect alarms for a specific field. This will show in the alarmlist. Empty is off. |
| InhibitField1 | Select  (FTSelect) | Field That this I/O should be inhibited or not inhibited to. Se definition “Field”. |
| InhibitType1 | Value  (Higher, Lower) | Inhibits the field depending on if the type is Higher or Lower. |
| InhibitValue1 | Value | Value when to inhibit. ( i.e. Inhibit when RPM is Lower than 500). So choose 500 here. |
| InhibitLogic | Value  (AND, OR) | Logic for second inhibit field. Choose between different possibilities. |
| InhibitField2 | Select  (FTSelect) | Field That this I/O should be inhibited or not inhibited to. Se definition “Field”. |
| InhibitType2 | Value  (Higher, Lower) | Inhibits the field depending on if the type is Higher or Lower. |
| InhibitValue2 | Value | Value when to inhibit. ( i.e. Inhibit when RPM is Lower than 500). So choose 500 here. |
| InhibitBeforeDelay | Value  (seconds) | Delay before inhibit kicks in |
| InhibitAfterDelay | Value  (seconds) | Delay after inhibit stops |
| Weight | Optional | Weight |
| CableLength | Optional | CableLength |
| Connector | Optional | Connector |
| Supply | Optional | Supply |
| Consumption | Optional | Consumption |

Table 3‑1: Sensorlist columns

Sensor types can be used for in- and outputs (read/write). The interpretation of the read values and written values differs a bit, so they are described separately

|  |  |  |
| --- | --- | --- |
| SensorType (Mode: Read) | | |
| Option | Sensor | Description |
| Standard | Value | Sensor value represents the state of the Data Field itself (Default) |
| Set | On | Request to turn on |
| Off | No action |
| Reset | On | Request to turn off |
| Off | No action |
| Pending | On | Processing a request. |
| Off | No action |
| Auto | On | Switched by an automatic control sequence |
| Off | Controlled by an operator |
| Manual | On | Controlled by an operator |
| Off | Switched by an automatic control sequence |
| Low Speed | On | Running at low speed |
| Off | Off, when not in “High Speed”. Otherwise no action |
| High Speed | On | Running at high speed |
| Off | Off, when not in “Low Speed”. Otherwise no action |
| Closed | On | Switched off |
| Off | Processing a request, when not “Open” |
| Open | On | Switched on |
| Off | Processing a request, when not “Closed” |
| Ready | On | Ready for use |
| Off | Not ready for use |
| Remote | On | Remote control. Controlled by AMCS |
| Off | Local control. Not controlled by AMCS |
| Ack | On | Acknowledgement of alarm on the assigned field |
| Off | No action |
| Request | On | Request to turn on |
| Off | Request to turn off |
| Push | On | Request to turn on, when off.  Request to turn off, when on. |
| Off | No action |
| Too Low | On | Value is too low |
| Off | Value is not too low |
| Low | On | Value is low |
| Off | Value is not low |
| High | On | Value is high |
| Off | Value is not high |
| Too High | On | Value is too high |
| Off | Value is not too high |
| Failure | On | Defect |
| Off | Not defect |
| Precision | On | High precision frequency counter in 0.01 Hz accuracy up to 10kHz |
| Off | Low precision frequency counter in 1 Hz accuracy up to 100kHz |
| Counter | Value | The changes in this counter value will be added to the field |
| Sign | On | The value read by “Standard” is negative |
| Off | The value read by “Standard” is positive |
| Index | Value | Value is the index of a serial message. See “Index” description |
| Pulse | On | Field’s value is counted 1 up |
| Off | No action |
| Pulse 1/2 | Value | Used in combination with “Pulse 2/2” to detect movement with two proximity switches. |
| Pulse 2/2 | Value |
| Pulse 1/3 | Value | Used in combination with “Pulse 2/3” and “Pulse 3/3” to detect movement with three proximity switches. |
| Pulse 2/3 | Value |
| Pulse 3/3 | Value |

Table 3‑2: Sensor Type mode Read

|  |  |  |
| --- | --- | --- |
| SensorType (Mode: Write) | | |
| Option | Sensor | Description |
| Standard | Value | Requested state of the Data Field itself (Default) |
| Set | On | Request to turn on |
| Off | No action |
| Reset | On | Request to turn off |
| Off | No action |
| Pending | On | Processing a request. |
| Off | No action |
| Auto | On | Request to turn automatic control sequence on |
| Off | Request to turn automatic control sequence off |
| Low Speed | On | Request to run at low speed |
| Off | Request to turn off, when not in “High Speed”. Otherwise no action |
| High Speed | On | Request to run at high speed |
| Off | Request to turn off, when not in “Low Speed”. Otherwise no action |
| Impulse | On | Request to turn on, when off.  Request to turn off, when on. |
| Off | No action |
| Status | Value | Output value represents the state of the field/ device itself  (No control) |
| Ready | On | Ready for use |
| Off | Not ready for use |
| Remote | On | Remote control. Controlled by AMCS |
| Off | Local control. Not controlled by AMCS |
| Too Low | On | Value is too low |
| Off | Value is not too low |
| Low | On | Value is low |
| Off | Value is not low |
| High | On | Value is high |
| Off | Value is not high |
| Too High | On | Value is too high |
| Off | Value is not too high |
| Failure | On | Defect |
| Off | Not defect |

Table 3‑3: Sensor Type mode Write

|  |  |  |
| --- | --- | --- |
| Unit Type | Select | Description |
| Alarm | Alm | Alarm |
| Ampere Hour | Ah | Ampere hour |
| Angle | ° | Angle |
| Angular Acceleration | °/s^2 | Degrees per square second |
| Angular Speed | °/sec | Degrees per second |
| °/min | Degrees per minute |
| Content | % | Percentage |
| G | Gallon [US] |
| M3 | Cubic meter |
| L | Liter |
| Guk | Gallon [UK] |
| Consumption per Distance | l/nm | Liter per nautical mile |
| l/km | Liter per kilometer |
| G/nm | Gallon [US] per nautical mile |
| l/m | Liter per meter |
| Consumption per Time | G/H | G/H |
| G/M | G/M |
| Guk/M | Guk/M |
| Guk/H | Guk/H |
| l/m | l/m |
| G/S | G/S |
| Guk/S | Guk/S |
| l/h | l/h |
| L/S | L/S |
| Counter | x | Count |
| Course | ° | Course |
| Current | mA | MilliAmpere |
| kA | Kilo Ampere |
| A | Ampere |
| Dampening | D | Dampening |
| DistanceContent | nm/G | nm/G |
| nm/l | nm/l |
| km/l | km/l |
| m/l | m/l |
| Force | Pdl | Poundal |
| Lbf | Lbf |
| N | Newton |
| Kgf | kg |
| ForceLength | Kgm | Kgm |
| Lbf-ft | Lbf-ft |
| kips | Kips |
| Nm | Nm |
| Frequency | Hz | Hertz |
| FuelEconomyGaseous | nm/kg | nm/kg |
| m/g | m/g |
| km/kg | Km/Kg |
| FuelEconomyPower | kWh/l | kWh/L |
| kWh/Guk | kWh/Guk |
| kWh/G | kWh/G |
| Length | km | Km |
| mi | mi |
| cm | cm |
| nm | NM |
| ft | Feet |
| fm | Fathom |
| mm | mm |
| m | M |
| in | Inch |
| Luminance | cd m-2 | cd m-2 |
| Magnetic | ° | Magnetic |
| MassSpeed | g/s | g/s |
| t/s | t/s |
| kg/h | Kg/H |
| Name |  |  |
| Number |  |  |
| okta | Okta |
| Percentage | % | Percentage |
| Position | ° | Degrees |
| Pressure | psi | Psi |
| Pa | Pascal |
| kPa | kPa |
| mbar | mBar |
| hPa | hPa |
| Hg | Hg |
| bar | Bar |
| Resistance | ohm | Ohm |
| mOhm | MilliOhm |
| kOhm | KiloOhm |
| RPM | rpm | RPM |
| RPMAccelaration | rpm/s | RPM/s |
| Speed | km/h | Km/H |
| m/min | M/Min |
| m/s | M/S |
| ft/min | Feet/Min |
| kn | Knots |
| B | Beaufort |
| mph | Miles per hour |
| SpeedAcceleration | g | g-force |
| m/s2 | M/S2 |
| Status | Open | Open |
| On | OnOff |
| Switch | Take Over | Take Over |
| S | Switch Off |
| Alarm Group | Alarm Group |
| General Alarm | General Alarm |
| Alarm Deadman Group | Alarm Deadman Group |
| P | Push |
| S | Switch |
| PS | Popup Switch |
| Temperature | K | Kelvin |
| °C | Celsius |
| °F | °F |
| Time | Mn | Month |
| H | Hour |
| D | Day |
| DTL | Date & Time Left |
| D | Date |
| ms | mSec |
| us | uSec |
| Wk | Week |
| M | Min |
| T | Time |
| DT | Date & Time |
| S | Sec |
| Yr | Year |
| True | ° | True |
| Voltage | mV | MilliVolt |
| kV | KiloVolt |
| V | Volt |
| VoltAmpere | VA | VA |
| kVA | kVA |
| VoltAmpereHour | kVAh | kVAh |
| VAh | VAh |
| MVAh | MVAh |
| Watt | MW | MegaWatt |
| W | Watt |
| kW | kW |
| WattHour | Wh | WattHour |
| MWh | MegaWattHour |
| kWh | kWh |
| Weight | lbs | Lbs |
| kg | kg |
| g | Gram |
| t | Ton |

Table 3‑4: Unit Type

## Implementation in the sensorlist

### Introduction

As the sensorlist is way bigger and more complex than the devicelist, we will not fill in all the columns and fields. We will give some excerpts from what you can expect at the different devices and the different columns. On the basis of the single line drawing we used for the explanation of the devicelist, we will give as many examples as possible. After this explanation, you should be capable to work out the rest of the sensorlist.

*: Once you have imported the sensorlist into NavVision, most of the fields will be automatically added. This will be done by NavVision on standard basis. This will not always be right, so you need to check that. We will come back at that in a separate Chapter*

### Import Result

The import result is a checklist. When you have imported the sensorlist, NavVision will generate a few files at which we will come back later. One of these files is the sensorlist\_generated. In this file you will see in the first column the import result. For more information we refer you to Chapter 4.5.

### ID, CableLabel, GroupLabel

These columns are optional. They are not needed for the proper functioning of the program. However it could come in handy when you fill up some of those fields.

The ID column you can use for your own reference. Maybe you use some kind of numbering that is different from the one you get from the shipyard.

Many installation companies use cable labels (numbers) for the connections of the wires at the terminals and/or at the sensor/control. If you fill in these Cable labels in this column, you will have a reference in the sensorlist which is searchable. You also get a reference in NavVision where the Cablelabel is shown in the Wago screen.

In the GroupLabel column you can separate different (alarm) groups and their I/O. This makes it quite easy to search specific I/O or just select a whole group that you need to adjust.

Next figure will show a small example:

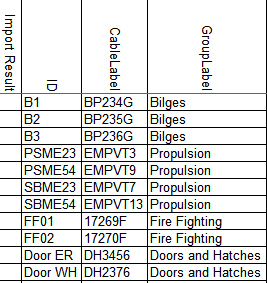


Figure 3‑1: ID, CableLabel, GroupLabel Example

### Item

The Item is somewhat different and needs some attention. In consultation with the installer or even with the shipyard, you need to come up with a descriptive name for each field (I/O, sensor, control). As this is the name that comes up in the logbook and the Alarmlist, you need to be clear about what it is.

Sometimes people come up with texts like “Preferential Trip & Em. Stop System Power Failure”. As you can see it is quite long and also very confusing. It can mean a lot of things.

Maybe this one would be easier to understand if you called it “PMS Power Failure”. It is certainly more descriptive and short and concise.

In other cases, the crew can be very familiar with certain names. The example “N.16 Fr 20-21 Bilge Level High Alarm” may seem confusing, but the crew knows exactly what it means because they have been working with this name for years.

Remember however that the text is free to choose, but it will appear in alarm lists and the logbook. So keep it as simple as possible.

#### Conjunction with SensorType

You also need to understand the conjunction with the “Item” column and the “SensorType” column. As explained in Chapter 3.3.5 SensorType defines which subfield or action of the Data Field is set by the value in that column. So if it is not “standard” you better check the “Item” text again.

For example: A sensortype can be “High Alarm” or “Running” or even just “Alarm”. This means that you trigger an extra action with the sensortype field. Now let’s say that you have the sensortype defined as Alarm. When you put “Bilge ER Alarm” as text in the “Item” field you get it double. With an alarm you now will get “Ext: Bilge ER Alarm Alarm” in your alarm screen. Easy to understand that if you use the sensortype “Alarm” you leave the word Alarm out of the Item-text. This is valid for all the conjunctions between these two columns.

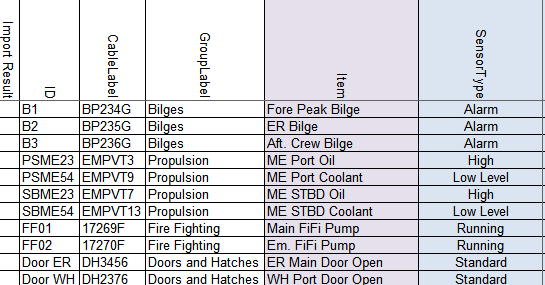


Figure 3‑2: Item example

### SensorType

SensorType defines which subfield or action of the Data Field is set by this value. By default it’s “Standard”. Standard means it’s not defining a subfield or action, but the value of the Data Field itself. (For more options see Table 3‑2 and Table 3‑3).

With “standard” as option in the sensortype column NavVision will only act upon the field itself. So if the field is an alarmfield NavVision will give an alarm when that field gets

triggered. This goes for all the different type of fields. So if for example it is a Pressure field (analogue value) NavVision will show the pressure value. If you don’t fill in anything in the sensortype column, it will automatically be “Standard”.

If no extra action is necessary on a field you probably won’t use the sensortype column. This comes in play when you want something extra. An analogue field that needs a “high” alarm. An output that needs a “Set” request and so on. Before we elaborate on this we need to explain something about the “Fields” within NavVision.

#### Fields

NavVision works with a database with all kind of ID’s in it. These ID’s are represented by fields that are divided into sup-parts. Every action in NavVision revolves around this database of field-id’s. You can use one field over and over again because the main value is set in the database.

Once you connect a sensor or control to a field you can do almost everything you like. For example if you want to control a pump with a hardwired button, you can connect that pump in NavVision to let’s say the field “Pump1”. Through a Wago PLC you now get to control that pump. On a Wago Digital Input, you connect the field “Pump1” and you hardwire a button to the same Slice. Now if you push the button the Wago input will get high. If you put the same field “Pump1” to an output on the Wago. This output will get active as soon as the input gets active. While this is an output, you can hardwire it to the actual Pump1. So than when you push the button the pump will start to run.

These fields you can find in the “fieldlist.txt”. Once that NavVision is started for the first time, you will find it in the root folder. You can open and control this .txt-file best with Excel. For people not familiar with Excel there is a small explanation in Chapter 1.2.

As there is a lot of intelligence in the fields already it is good to understand the interaction between the field and the sensortype. You can mess things up when you use this wrong.

#### Back to SensorType

So, as mentioned earlier, there is a conjunction between the “Item” and the “SensorType” and now also between the “Field” and the “SensorType”. We use the same example as in Chapter 3.3.4.1 to show how it all fits together.

As we mentioned in that chapter, you need to pay attention to the name you use in the Item-column so you do not get confusing or double values. Same goes for the fields and the sensortype. If you choose a field that is already an alarm-field this means that, when the value gets high, the field will give an alarm. So it is not necessary to put an extra alarm in the sensortype column. This is not only double but also can confuse the system or the user. On the other hand, if you use a field that holds Level information, you might want to trigger an alarm when you get to a certain level. This is possible by putting “High Level” in the SensorType column. You see there is quite some interaction between those different columns.

You need to practice a lot with the sensorlist to learn how to work with it. For now we will give an example on how it is not supposed to be concerning “Item” “SensorType” and “Field”.

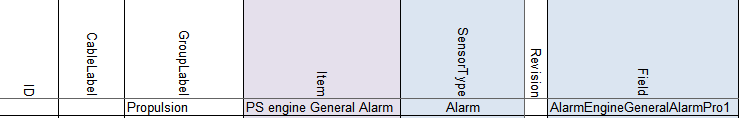


Figure 3‑3: Double fault

As you can see we have an alarmfield in the field column, a sensortype that triggers an alarm and the name in the Item column that will make it double. Easiest in this case is: keep the alarmfield in the Field column, put Sensortype to “Standard” and take “Alarm” out of the Item column name.

### Connection

Connection defines the type of connection for digital in- and outputs. Connection is NO by default. If an in- or output is normally closed it is NC. If you have problems with switches that go the wrong way around or there is an alarm where the sensor itself is not in alarm, this is the first place to look.

### Device

Identification of the device where the sensor/control or serial device is connected to. This text should be unique for each FT NavVision device. The text is case sensitive.

This device is already been set in the devicelist. See chapter 2.5.2 to see how you’ve done that. Now all the I/O that you put into the sensorlist must be connected to the right device, so NavVision knows where to look for it and how to process it.

When you click on a field you can see there is a drop-down menu. In the menu you will find all the previous assigned devices. All you have to do now is choose the right device.

As we look at the single line drawing and we take the example we had earlier we can tell that the Fore Peak Bilge is connected to the Wago AC Room and the ER Bilge is connected to the Wago 2 ER.

The Port Engine is connected to the second port at the second SerialLan in the ER so you choose Serial 2 ER-2 as device. It will look a bit like the following figure:

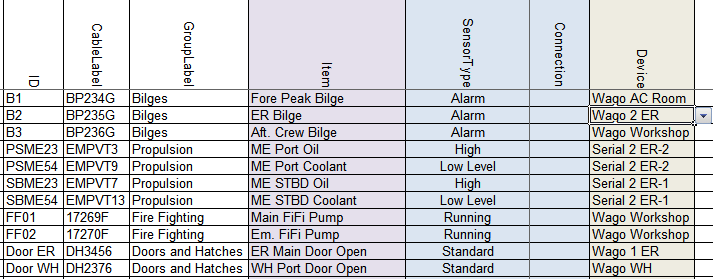


Figure 3‑4: Sensorlist device column

Of course, while you probably will start filling all the I/O’s from one device at the time, you will get a long row with only Wago WH and then for example Wago Workshop. You will see that once you start working with it.

In the example we only have serial and Wago connections, but it can be anything that you filled in as a device. It is probably best to start to fill the list with the Wago devices as these are mostly already assigned. Later on you take the serial connections with for example Modbus or CANbus on it.

### Location

For location you can use the same field as in the devicelist. It is optional, but also usable for sorting the list and/or localizing sensors or I/O’s.

### Interface

Here you define what kind of interface is used to connect the sensor/control to NavVision. For Wago this is divided in the slice’s type-number. For Modbus, Canbus and other protocols it is Serial (Digital/Analog) in or out.

If you have the Wago drawings available, it is easy to choose the right module for that. If you have trouble finding it, you can always fall back to the documentation of Wago. For the protocols you just need to look if it is a digital or analog value and if it is an input or an output. More on these serial interfaces we discuss later.

To give you an idea, we go back to our example. The bilges in the example will be most likely digital inputs. As Wago works standard with 24V it will be a Dig in (24V) you have to choose there. This goes also for the fire pumps and the doors. Probably normal switches so an input of 24V for High or Low (On or Off).

The engine however is somewhat different. As we can see in the SensorType field it is just a digital input where On is High or On is Low Level. However, this is the SensorType Field. This field will give NavVision a reason to calculate an alarm on an analog value. So don’t be misled. This will be an analog field coming in (Oil is a pressure field and Low Level is a level field). So you will have to connect it to an analog interface module on the Wago. This can be 4-20mA, 0-10V or a lot of other sorts. Let’s say the oil pressure field is a 4-20mA signal and the level field is a 0-10V signal. We will come to the following:

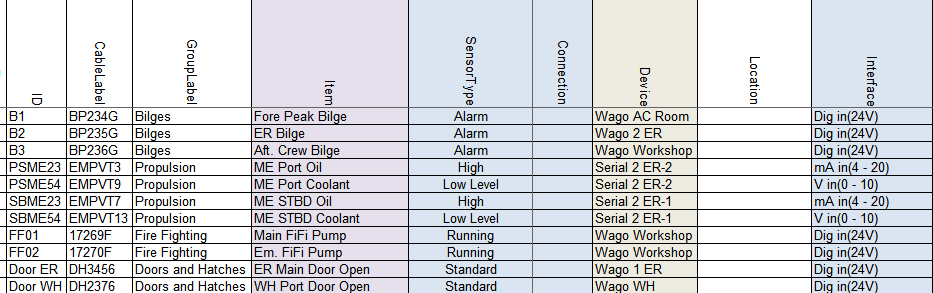


Figure 3‑5: Sensorlist Interface column

### Module

For Wago you start counting the slices after the 750-626 module. Starting with 1 and so on. If you do not filter the sensorlist, than it will be hard to look if the numbers are alright. But as we will explain that in a later stadium we now just have to watch carefully. As example we show you the next figure:

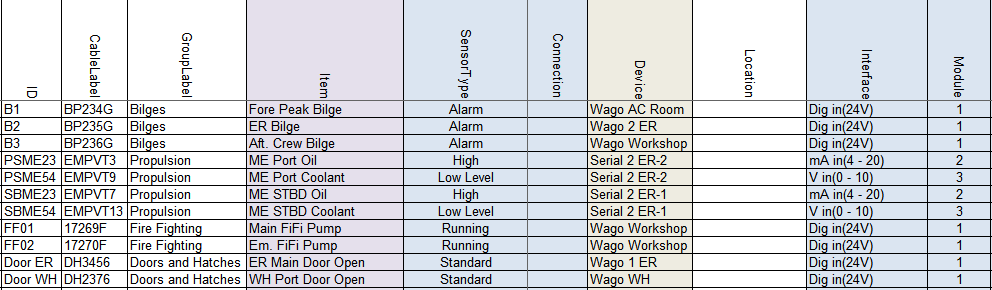


Figure 3‑6: Sensorlist Module column 1

This may look a bit odd, but realize that we put the Dig in (24V) on a module with 8 contacts (Pin). So the first 8 DI you find are on the first module. Same goes for the mA in (4-20). These modules have 4 contacts. It will become clearer in the next paragraph.

For CANbus in this column you put the PGN or Parameter Group Number as index for the I/O. With Modbus you take the Modbus mapping as starting-point. The register of the Modbus mapping you put here. See following example:



Figure 3‑7: Sensorlist Module column 2

### Pin

The I/O index on the module for WAGO and the bit offset in the message for serial protocols. (NOTE: The pin index is 1 based).

#### Wago

So if you look at a Wago slice you will see openings for the wires to be attached. It needs some attention because Wago has a different numbering than NavVision and this can be confusing. First let’s look at the numbering Wago uses:

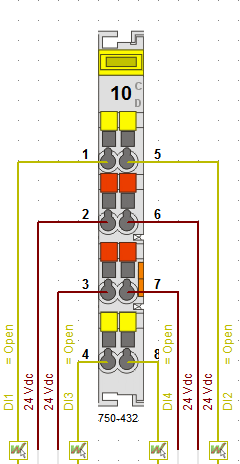


Figure 3‑8: Wago Numbering 1

As you can see Wago numbers the pins vertically so left side 1-4 and right side 5-8.

NavVision has to number different because of program issues. We number the Wago horizontally. So 1=1, 5=2 and so on. You have to keep that in mind to work properly with the sensorlist. The NavVision numbering will look as follows:

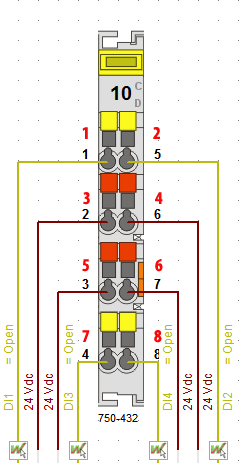


Figure 3‑9: Wago Numbering 2

So when you number it this way in the sensorlist, it will mainly look like the following figure:

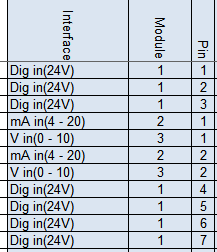


Figure 3‑10: Pin column 1

Or, when you already sorted the sensorlist, it will make it even clearer. See the following figure:

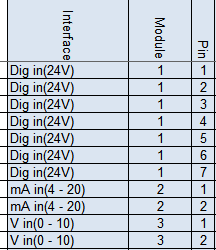


Figure 3‑11: pin column 2

Of course I can’t show you the example from where we started off. While all the connections where on different Wago’s there, we should than have divided all the modules over the different Wago stations. Therefore, before you begin numbering the modules and pins, you need to have all the Wago connections in the sensorlist. Then you can filter the sensorlist first (as explained in Chapter 3.4) and then do the modules and pins.

#### Serial Protocols

For Serial protocols the pin number defines the bit-offset. So if you need to connect to a serial protocol at bit level, this column is where you assign this. Note that the “pin index” is 1 based. So if you need bit 3 for PGN 65280 you have to put 4 in the column. See next figure:



Figure 3‑12: Pin column 3

### Type

#### Wago

For Wago you fill in here the module number. You can find the module number on the Wago slice itself, on the drawings or look it up in the Wago documentation. See following figure:

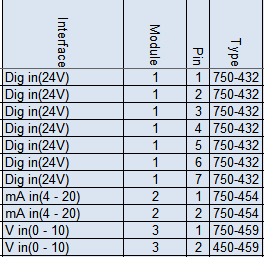


Figure 3‑13: Type column 1

#### Serial Protocols

For CANbus we do not use this column. For Modbus we use this column to define the function code of the Modbus register. So for example if you read actual values in Modbus, this will be Modbus function 04. Type a 4 in the “Type” column. See following figure:



Figure 3‑14: Type column 2

### Min-Max

The columns Min and Max show the range of the data field. This will come back in instruments and value-bars. If you choose them wrong then you get values that go beyond the range of an instrument. Once you see this, you know that you need to change the values. If you get the right data from the shipyard you can fill it in in these fields. For digital data it is Min=0 and Max=1. It is not necessary to fill in the Min- Max-values for digital values. NavVision will do this for you. You can also change these values at a later time.

### DefaultUnit

The defaultUnit is used to set the unit to present this Data Field in. This can also be changed in the instrument or mimic itself, but for big amounts of data it is easier to use the sensorlist. If you do not choose anything NavVision will fill it in for you. For options see the next figure:

|  |  |  |  |
| --- | --- | --- | --- |
| Alarm | High Alarm | Low Alarm | Ampere-Hour |
| Radians | Degrees | Grads | °/sec² |
| rad/sec | °/sec | °/min | Normal |
| Normal | Mirror | Liter | Gallon |
| GallonUK | Cubic Meter | Percentage | L/km |
| G/Nm | L/min | L/Nm | G/S |
| l/h | G/H | Guk/H | G/min |
| L/S | Guk/min | l/m | Guk/S |
| Count | Degrees | Grads | Radians |
| Kilo Ampere | MilliAmpere | Ampere | Dampening |
| kg/m³ | kg/L | lb/gal | nm/G |
| nm/l | m/l | km/l | Poundal |
| Newton | Lbf | Kgf | Kips |
| Newton Meter | Kgm | Lbf-ft | Hertz |
| m/g | nm/kg | Km/Kg | kWh/L |
| kWh/Guk | kWh/G | Fathom | Nautical Mile |
| Feet | mi | cm | Km |
| mm | Inch | M | cd/m² |
| Kg/H | g/s | t/s | Name |
| Okta | Mask | Number | Percentage |
| Degrees | Bar | mBar | kPa |
| Hg | hPa | MPa | Psi |
| Pascal | MilliOhm | Ohm | KiloOhm |
| RPM | Hertz | RPM/s | Km/H |
| Knots | M/Min | M/S | Beaufort |
| Miles per hour | Feet/Min | g-force | m/s² |
| OnOff | Open | Alarm Group | General Alarm |
| Switch Off | Alarm Deadman Group | Switch | Take Over |
| Push | Popup Switch | Kelvin | Fahrenheit |
| Celsius | Date | Day | Date & Time |
| Month | Date & Time Left | Time | Sec |
| Week | Hour | Year | Min |
| mSec | uSec | MilliVolt | KiloVolt |
| Volt | Volt Ampere | kVA | Volt Ampere Hour |
| kVAh | MVAh | Watt | MegaWatt |
| KiloWatt | MegaWattHour | WattHour | kWh |
| Ton | kg | Lbs | Gram |

Table 3‑5: (Default) Unit options

For our example it will be the following:

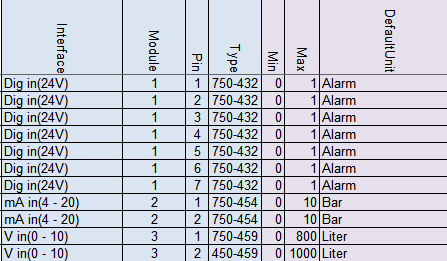


Figure 3‑15: Default Unit column

### Manufacturer

This is an optional field for your own convenience

### Supplier

This is an optional field for your own convenience

### Comment

This is an optional field for your own convenience

### Revision

This is an optional field where you can give a revision number. Easy if you need to see when something has been changed or what has been changed after a certain revision.

### Field

This is one of the most important columns within the sensorlist. This is the place where you assign a dedicated field from the database of NavVision. This field will be inextricably linked

to that I/O, sensor or control. All the in- and outputs and all the calculations, as well as connection to instruments and mimics, will be represented with that field. Also the alarmgroup and behaviour will be defined by that you choose here.

You can understand that it is utmost important that this field is chosen properly and a field is only used for one particular sensor/control. These field-column is also the one that will consume most of the time in building the sensorlist.

As mentioned before these fields can be found in the file “fieldlist.txt” in the root of NavVision after the first start of NavVision.

#### How to work with fieldlist.txt

To find all the right fields you first have to open “fieldlist.txt” the right way. You need to know that, to work with the .txt-file you need to open it in Excel. To do so, right-click on the .txt file and choose “open with” and go for Excel (see Figure 3‑16)

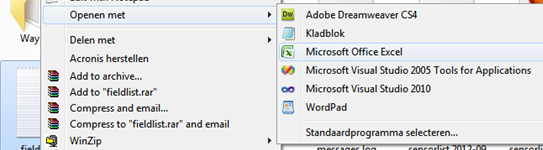


Figure 3‑16: open with Excel

Now the program will open as an Excel sheet, with all the opportunities. There are two things you must do first (this is basic Excel knowledge).

Click in the upper left corner of the sheet (see Figure 3‑17) to select all fields. Put your mouse between row “A” and row “B” (see Figure 3‑18)and doubleclick. The fields now will be all on the right width.

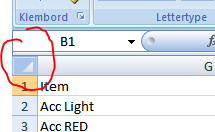


Figure 3‑17: Excel 1

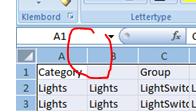


Figure 3‑18: Excel 2

Now select the first row by clicking with you mouse on the number “1” in front of the row. Goto Start>sort and filter and then filter (see Figure 3‑19). Click it

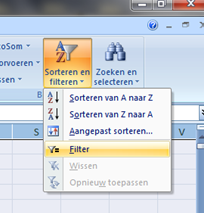


Figure 3‑19: Excel 3

The first row with the index names has now drop down menus and you can choose what to filter. For our example we need Bilges. Goto the index name “Category” click on the dropdown menu, deselect the “select all” checkmark and then select the “bilges” checkmark (see Figure 3‑20). You now have only all the bilges-fields available.

You can narrow it down by going to the index name “Group” and make another selection (see Figure 3‑21). In our case it is AlarmBilge

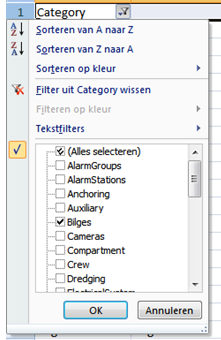


Figure 3‑20 : Excel 4

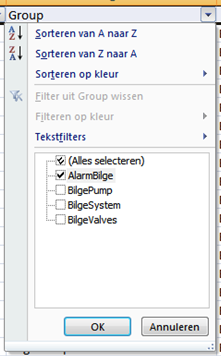


Figure 3‑21: Excel 5

Now we’ve done this we have only the Bilge alarmfields available. You can figure out yourself how you can further narrow it down, or use it for other fields.

#### Back to the Field column

So now we have narrowed it down to the right fields, it is time to give all our I/O a separate field tag. In the adjusted fieldlist.txt we now see all the alarms for bilges available. We need three bilge alarms, so we need three distinctive bilge alarm fields.

In the previous mentioned excel list, goto the column “Field”. As we are just starting, all the fields are still available. So we can choose the first three Bilge Alarm Fields. Select these three fields and copy them (CTRL-C). Go back to your sensorlist and past them into the field column behind the three bilge items. See the following figure.

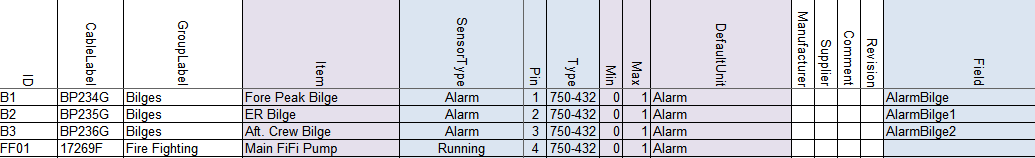


Figure 3‑22: Field column 1

You can follow this for all the other fields and you will get the following:

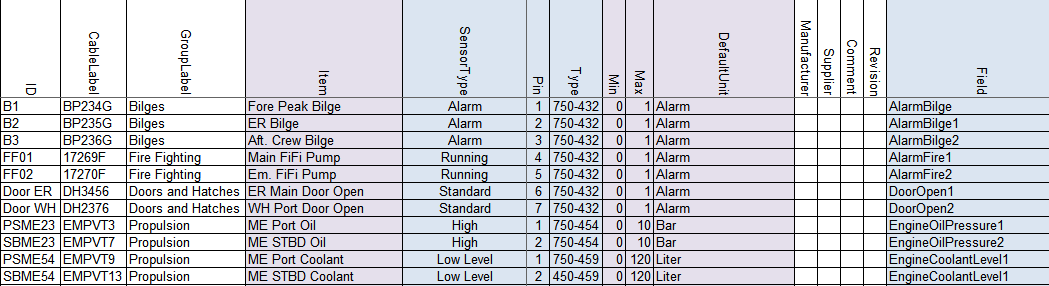


Figure 3‑23: Field column 2

*: with bigger projects it is easy to get mistaken. Easiest way to prevent this is that you color the fields u have used in the fieldlist.txt yellow. That way you will know which ones are used and which are free. Later on we show you that NavVision has a way to trace the faults. See chapter 4.5*

### Label

The Label column exists of the short description of the Data Field when shown in an instrument. Default label text belonging to the Data Field is preferred.

So the easiest way is to copy the “Item” column and just past it into the “Label” column. This way you have a one-on-one connection. Off course this is the text that you find as a label in instruments etc. When the text is too big, it won’t fit in the instrument or just looks sloppy. So if this is the case, just alter the name here to a short description. “Emergency Generator Power Failure” can be changed into “Em. Gen. Power Fail.” And if the default unit is available in an instrument, you can even leave types as “Pressure”, “Voltage”, etc. out of it, cause they will see that it is a “Bar” value or a “V” value. So “Main Engine Lub Oil Pressure” can be set as “ME Oil”

### Rate

Rate describes the number of samples per second of a sensor/control. This is defined by the protocol. Leave empty.

### Index

Index defines when this Data Field Definition [DFD] is valid. The Index column can only be used in combination with a Data Field Definition [DFD] that has the SensorType set to Index and is in the same message as this DFD. Default is empty.

### Datatype

DataType is used to define the type of value on serial protocols. For analogue values it’s Float, Signed or Unsigned. For digital values it’s Bool. For enumerations this is Enum. See Enum column.

### Enum

Enum is the index value where the received value should compare to, to switch the Data Field on. If the value is not equal to the Enum index the Data Field is switched off.

### Count

Count is the number of bits starting from the pin index. For a digital value it’s typically 1 with a pin index between 1 and 16 and for analog values it’s for example for Mod bus typically 16 with pin index 1.

### Multiplier

Multiplier defines the factor between the sensor/control value and the real value.

For inputs/read:

*value = sensor value \* multiplier + offset*

For outputs/write:

*sensor value = (value – offset) / multiplier*

For example: if the temperature is send in from a sensor in whole numbers (210 for 21 degrees) you can put in a multiplier of 0.1. So when the sensor sends 210, it goes through the multiplier and NavVision makes it 210\*0.1=21

### Offset

Offset defines the offset between the sensor/control value and the real value. See Multiplier column.

### Unit

The Unit in which the sensor/control value is received or send. Directly from the sensor control. This field differs from the DefaultUnit by the fact that NavVision has no influence on this one. For options see Table 3‑5.

### Other columns

The rest of the columns in the sensorlist are optional, because NavVision will fill them in for you. These fields will only be used for specific needs. If you want to know what you can do with these columns, it is enough to look in the Sensorlist Table (see Table 3‑1).

## Filter sensorlist

Once you start filling the sensorlist it is good habit that before you fill in the columns module and pin, you filter the sensorlist. This is also common Excel knowledge, but for your convenience we will give a short explanation here.

Let’s say you have filled in a few I/O that you got from a list and you just start to fill in in no particular order. Than it is impossible to address the right module and pin as the list will be extremely long and changes on mistakes will be huge. So before you start with the module and pin columns you will have to filter the sheet.

The columns that you did fill in contain the device-column and the interface-column. With these two you can filter the sheet for a first result.

What you need to filter first is that all the devices are grouped and the interfaces are grouped together. To do this you go to Start>Sort and Filter>Custom sort. You will get a menu like in the folloing figure:

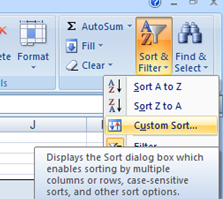


Figure 3‑24: Custom sort

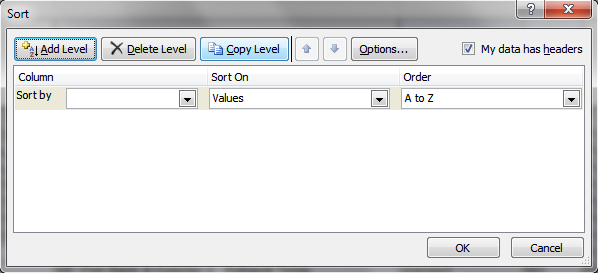


Figure 3‑25: Custom sort window

In this window you can add as many levels as you want to filter out the sheet. We need only two for now, “Device” and “Interface” as you see in the next figure:

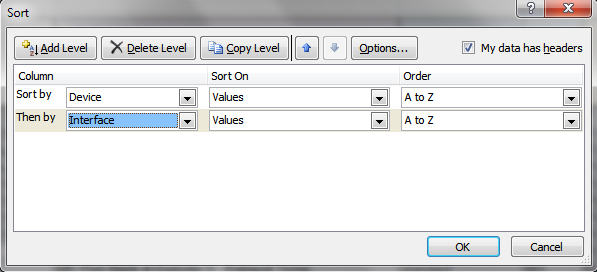


Figure 3‑26: Sorting device and interface

Sorting it this way gives you the devices ordered at the right Wago PLC and you get all the same slices together. This is the first step of filtering that is pretty easy and it gives the following example:

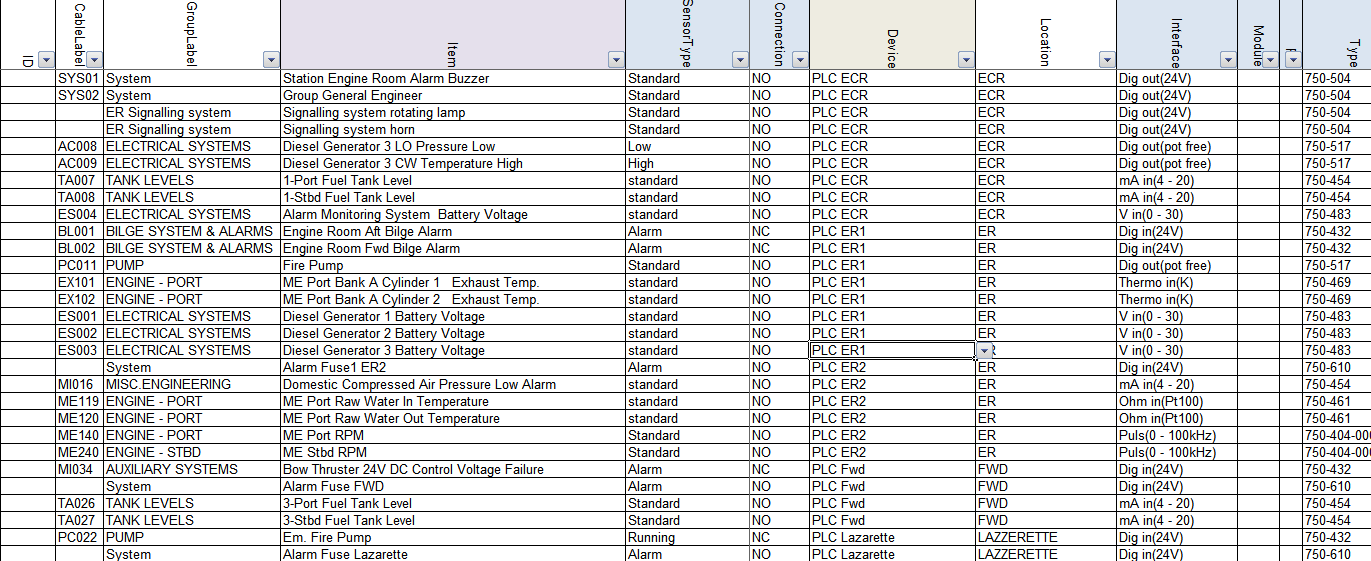


Figure 3‑27: Sorted sensorlist

As you can see we have all the devices put together and within these devices we have all the interfaces put together. Due to the running sequence Wago follows, we need to make some final adjustments by hand. There is a certain sequence that we have to build up the Wago PLC’s in. For more information we refer to Wago. For now we can say that we start the construction of Wago in the following (global) order:

First DI-modules

Than DO-modules

Than AI-modules

Than AO-modules

This is a global distribution, because it sometimes needs some additional action. For now this is enough to understand.

As you look at Figure 3‑27 you can see in the column “Interface” that it worked out pretty well. The only thing in this example that is not right are the modules at line 28 and 29. This is need to know knowledge. These modules don’t work in that position and has to be places before the 750-454 module at line 25.

To do so select the two lines (28 and 29) and cut them.

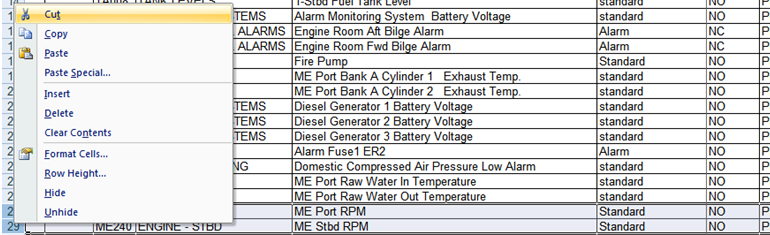


Figure 3‑28: Cut and paste 1

Once you’ve done that you go to the line that you need to insert them and right-click on the number of the row underneath that line. Choose Insert Cut Cells. See following figure”

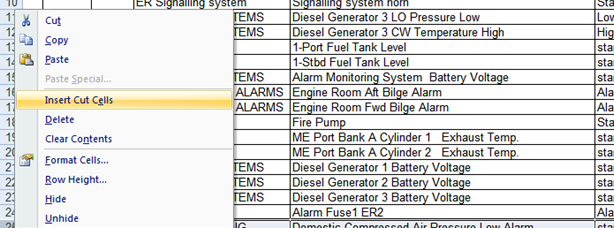


Figure 3‑29: Cut and paste 2

Now you have everything in the right order and you can start numbering the Modules and Pins.

*: You need to have well to excellent knowledge about Wago and Microsoft Excel. We recommend that you get some additional training on this as well.*

## Special issues

There are several special issues that you can put in the sensorlist. Changes you make in NavVision itself will get lost as soon as you import a new sensorlist. To prevent this loss it necessary that you put all the changes you make in NavVision are directly put into the sensorlist. In the hectic of a commissioning it will not always be possible to do that directly, for adjusting the sensorlist at a later time we refer you to Chapter 4.

However we do like to give an example of things you need to change by hand in the sensorlist. For this we assume that you have more than basic knowledge of working with NavVision.

So let’s say that you have a ship with a lot of duty-stations. At some point the crew will ask you to change the names in the alarm/duty mimic, so they can see who is on duty or who they are calling through the NavVision call function.

Given the next example (see Figure 3‑30 and Figure 3‑31) we have changed the names of a few files to match the names as the crew would like to see it. As you will know these names are changed in Fieldsettings>Comment>Crew>CrewAlarms within NavVision. If you do not put this in the sensorlist, each time you import a new sensorlist these names will be changed. This is not desirable, so you need to put these changes into the sensorlist.

If you put this in to the sensorlist, the easiest way to do this is on top of the list. Add some extra rows and start filling the information there. You have to understand that it is NavVision based so the device is NavVision NavVision. SensorType is Standard, Connection is NO and in the “Item” column you fill in the name that you want to show in the alarm mimic of NavVision (see Figure 3‑32).



Figure 3‑30: Duty names



Figure 3‑31: Call names

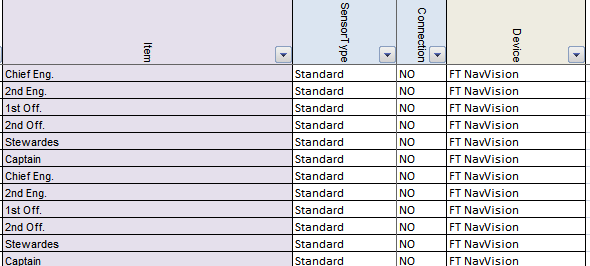


Figure 3‑32: Special issues 1

At the “field” column you assign the right fields (which you will find in the fieldlist.txt see Chapter 3.3.19.10). In the “Label” column you once again fill in the names as you described them in the “Item” column.

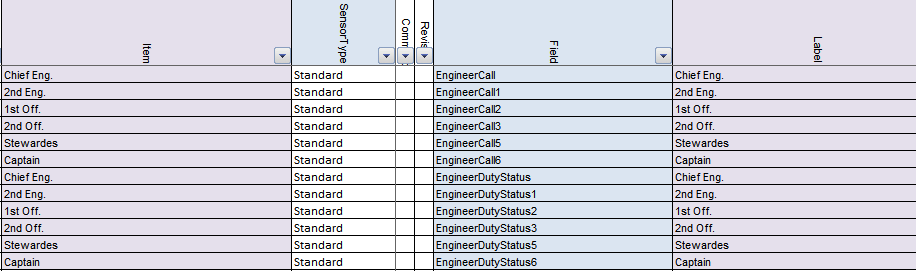


Figure 3‑33: Special issues 2

That is all. NavVision will take care of the rest. Now if you import the sensorlist again, you will keep the names you gave to the Crew Alarms.

# Importing in FT NavVision

## Introduction

Once you are finished with (a part) of the sensorlist, you will at some point need to implement it in NavVision. This is done by importing the sensorlist into NavVision.

In Chapter 1.4 you can see how that is done. Once you have the sensorlist.xls file ready you will put it in the root folder of the NavVision installation. We will go over these steps in the next chapters.

## How to import

Make sure that NavVision is closed and you are in the file explorer. You will have to be in the root folder. Here you will paste the sensorlist.xls file that you just created (see Figure 4‑1).

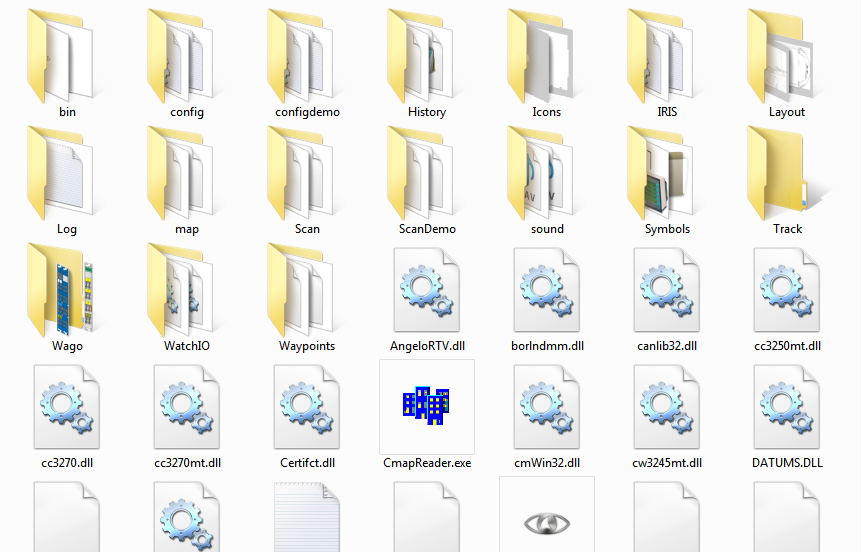


Figure 4‑1: Root folder

Once you have done this, you can start NavVision again. During the startup you will be asked if you want to import the devicelist and/or the sensorlist (this is referring to the 2 tabs in the sensorlist.xls. You answer yes to both the questions (see Figure 4‑2 and Figure 4‑3) and NavVision will continue the startup. At this time the sensorlist will overwrite the existing configuration.

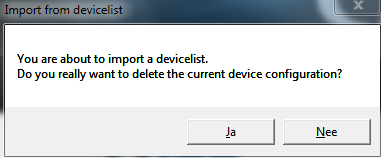


Figure 4‑2: Import devicelist

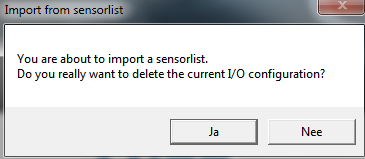


Figure 4‑3: Import sensorlist

Everything you have put into the sensorlist will now be in the configuration of NavVision. This cannot easily be undone, so be very careful if you import. There is a possibility to preserve the old system. Therefor it is necessary that you back up the complete “config” folder. If than anything goes wrong, you can paste the old config folder back.

*: It is always wise to keep a backup of the last working system on for back up sake. Always make a backup of, at least, the “config” folder.*

*: Make sure when importing a sensorlist (or even just working on the system) you work on one workstation only (close down all other stations). This way you prevent other workstations from interfering with your setup through the sync-function in NavVision.*

## Check the import

There is not a simple way to check if the import has been successful. The import function has been tested thoroughly by NavVision so the basic import function will work. It is wise to check the import anyway.

As you are probably the one that changed the sensorlist you will now which items has been changed, so you can check these items in NavVision. Also check if the connections are still allright in network>system layout (see Figure 4‑4) and if the Wago’s are still in place and connected right, etc. For more information on how to check these items we refer to the

“Installation and commissioning manual”.

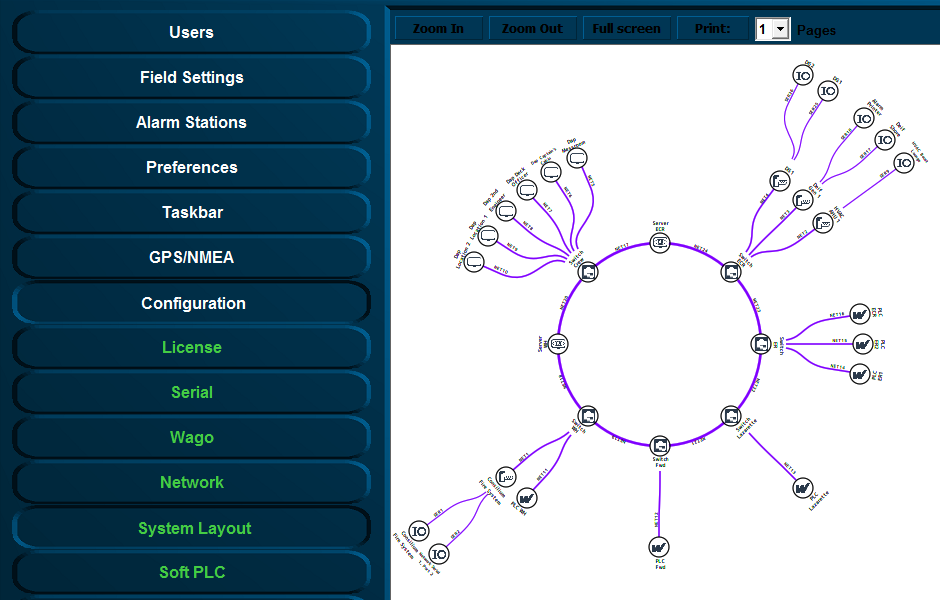


Figure 4‑4: Checking system layout

## Devicelist generated

Once you have made an import the system will make a “devicelist\_generated”. In this file you will find all the changes, faults, etc. that the system found. These are changes that are the differences between your devicelist import and the existing configuration. Also if you have made a mistake in the devicelist, it will be noted here so you can check whether you have to change something. The devicelist\_generated will look like the following figure:

*: Open the devicelist\_generated (or the other generated files which are all HTML-files) with right-click>open with> excel program.*

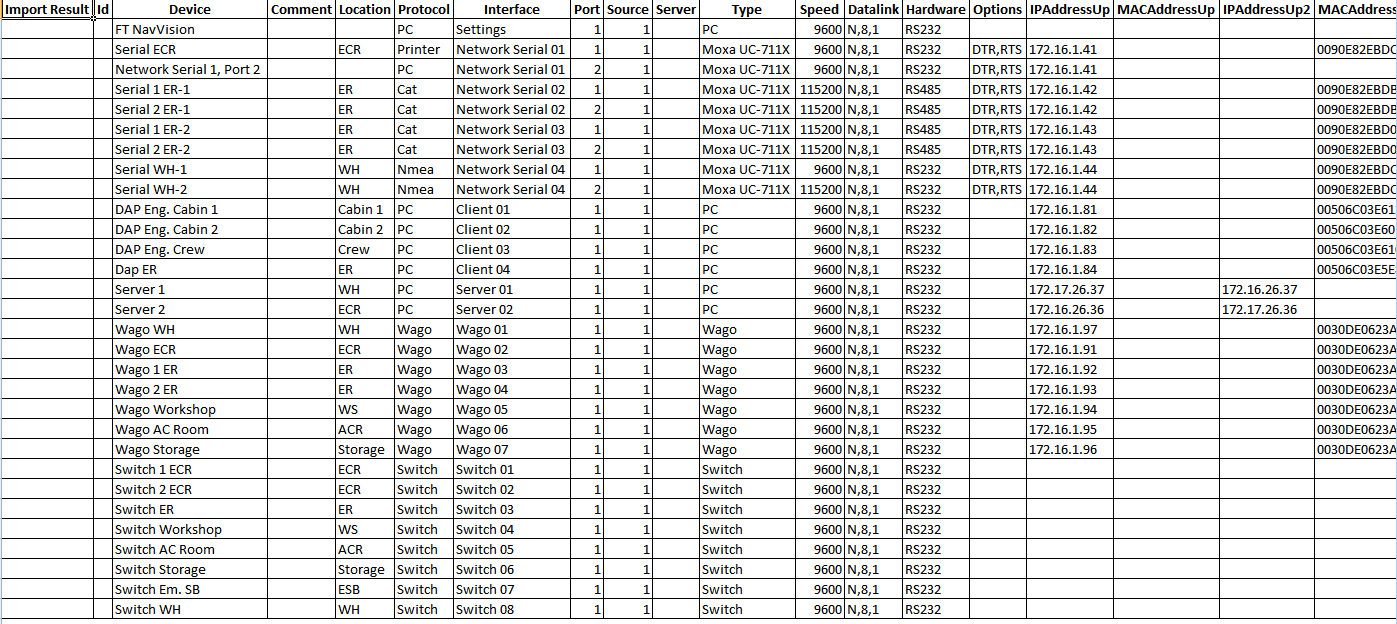


Figure 4‑5: devicelist\_generated good

This is of course when the devicelist was good in the first place. This is the kind of devicelist\_generated that you want to get back, because then you know you did well. When you have made a mistake you will find a comment (with a color) in the first column import result”. You can have something like the following figure:

|  |  |
| --- | --- |
| Field | Description |
| Comment | Comment that something is different in the field |
| Changed | Notice that something has changed |
| Failed | Critical failure somewhere in the field |
| Missing | Field tag is missing |
| New | Field is added since last import |

Figure 4‑6: Fault codes

The “comments” are merely there to make you aware that there is a small problem. Just check the line if there is an inconsistency in words or something. Sometimes it doesn’t even matter that there is a comment while you can deliberately made a difference in something. It doesn’t really affect he program.

The “changed” is there to warn you that there is something altered between the original configuration and the import. It can be two ways. Maybe NavVision changed something because the program noticed that you made a mistake. Maybe you mixed up a protocol or whatever. The fault in the row behind the import result will also be colored yellow. Sometimes there is something changed that differs between the original configuration and what you imported with the sensorlist.

## Sensorlist generated

The sensorlist has the same import result column. It also has the same fault codes as the devicelist. The only extra field that the sensorlist\_generated has is the import result “New” in a green cell. This means that with the import of the sensorlist, you introduced a new i/o or control or that you changed something in the NavVision program itself which is much more likely. In Chapter 5 we will explain that these fields are of much importance to keep the sensorlist up to date. For now you must know what you are looking at when you open up the sensorlist\_generated or the devicelist\_generated. The sensorlist\_generated is mostly much

bigger than the devicelist\_generated, so you can imagine that it will be a lot of work to keep the sensorlist up to date. See the next figure for a small excerpt of a devicelist\_generated:

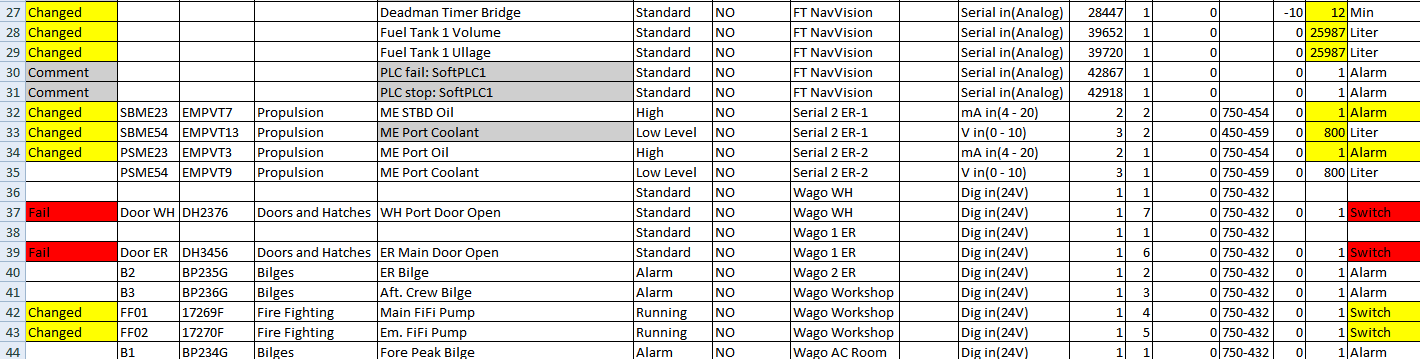


Figure 4‑7: sensorlist\_generated

## Sensorlist generated diff

The sensorlist\_generated\_diff is a help file that shows all the comments, fail and changes together with the corresponding original line (see Figure 4‑9). This is ideal if you are troubleshooting the sensorlist. There you can see what is changed and the line underneath will tell you how it was original. In the next Chapter we will show you how you can make use of this file to keep the sensorlist up to date. There is no need to use it, but some people find it easier to work with. Others just use the sensorlist\_generated. It is up to you what you will use.



Figure 4‑8: Diff example 1

This is a typical example of a “comment”. You can see that NavVision noticed that the name is changed. In the reference line you can see what it used to be. While this is probably the way you want it to be changed, you can ignore this comment.



Figure 4‑9: Diff example 2

Here you see a row that shows a changed state. In this case it is about the deadman timer bridge. Somehow in the program, somebody filled in 30 as the max amount of minutes. Later, probably after a new import, somebody changed it to 12 minutes. NavVision notices this change and point it out for you here. If you feel it is alright you can leave it. You will, however, have to change it in the original sensorlist, or it will come back at the next import.

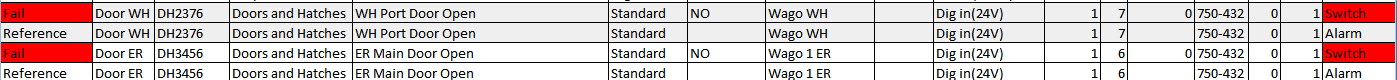


Figure 4‑10: Diff example 3

This concerns a real fault. NavVision will look at the “field” column and see that the Field is not an alarm field, but a switch field. It will notice you that there is a fault and you have to change something in the sensorlist. Either you change the “DefaultField” into switch instead of alarm, or you need to change the “field” into an alarmfield. Either way you will have to adjust the sensorlist.

### Making an export

When you import a sensorlist NavVision automatically generates the “\_generated”fields. There can be a time that you need to have one of these generated files without an import upfront. This can be done by stating NavVision with the extension “EXPORT”.

Find the file “NavVision.exe” in the folder NavVision/bin/ and right click on it. Choose create a shortcut. Right click on that shortcut and choose “properties”. In the target window type EXPORT in capitals at the end of the line (see Figure 4‑11).

Choose OK and start NavVision up by double-click on the shortcut. When NavVision has started you can close it directly. NavVision will have generated the files. Now you can go further as planned.

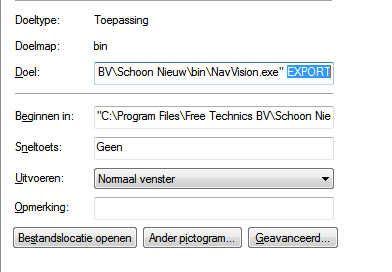


Figure 4‑11: Export shortcut

# Keeping up to date

## Introduction

Now you have seen what the sensorlist is capable of, you might have guessed that the sensorlist is the spill of the system. From the first build, up to changing large amount of data, the sensorlist is the tool for working with NavVision.

It is very important that you keep the sensorlist up to date during commissioning. The best way to do this is probably have the sensorlist open at your laptop and change immediately everything that you change in NavVision on board. We know that it is sometimes very hectic and you don’t have the time to do this directly. In that case it’s best that you change it right after you finished your days’ work. This way you can use the sensorlist the next day again.

We will explain here the different methods of keeping the sensorlist up to date.

## Direct changing

So this is the one that you keep the latest sensorlist open at your laptop, next to the workstation that you are working on. When you alter something directly on the workstation, you can immediately change that in the sensorlist.

I already gave an example in Chapter 3.5 with the crew names. But now let’s say that you are working on the workstation and you find out that the serialnetwork on moxa 1 port 1 has a different baudrate. The seriallan is the 1st one in the ER en you have to change port 1 to a baudrate of 38400 instead of 115200. In NavVision you change this on the workstation and the connection seems to be good.

Next time you import the sensorlist, you might wonder why the port isn’t working anymore. This is why you need to change it in the sensorlist in the tab “devicelist” to make sure next time the import will be in order. So go to your laptop, click on the devicelist tab and change the baudrate accordingly (see Figure 5‑1 and Figure 5‑2).

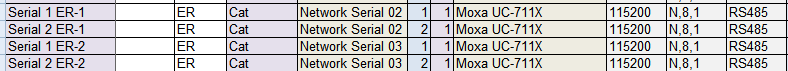


Figure 5‑1: changing baudrate old

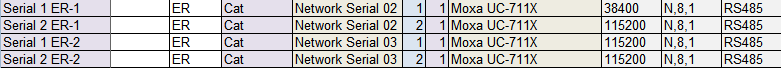


Figure 5‑2: Changing baudrate new

Same goes for the changes in the sensorlist. Again you’re working on the workstation and you notice that you have to change a connection at the Wago. It seems that the connections on the Wago Workshop are switched. The sensor on pin 3 is on pin 5 and the sensor on pin 5 is on pin 3. Of course you can change the wires on the Wago itself but for argument sake we say that you change the fieldnames in the Wago-section of the workstation.

Again you need to change this in the sensorlist or it will get back to the old state as you import the sensorlist again. The original lines you will find in the next figure:



Figure 5‑3: Changing Wago original

Now you can switch the whole line with names, fields and everything (see Figure 5‑4). Realize that you still need to change the pin-number, or nothing will change. For readability this will be the best option and also if you have to change a lot of pin numbers this is more synoptic. There will be an example later.



Figure 5‑4: Changing Wago lines

If it is about small amounts of changes it is easier to just change the pin-number. NavVision doesn’t mind and will put it in the right order into the system. See next figure:



Figure 5‑5: Changing Wago numbers

### Insert

When you need to insert a new connection into the Wago (an extra sensor for example), it could be very easy to do as you can read in the “Installation and commissioning manual”. Just choose a free pin in NavVision Tools>Configuration>Wago. However, don’t forget to put that also in the sensorlist or you will lose that connection again after importing.

Same goes for extra devices in the “devicelist” tab. Just remember: importing a sensorlist will overwrite every change you have made on the system.

## Bigger changes

One of the bigger changes that can take place is that you have to change the order of the Wago slices or you will have to add a Wago slice somewhere. This will mess up the whole configuration. Without using the sensorlist this is almost impossible to do.

Let’s pretend you have the following configuration:

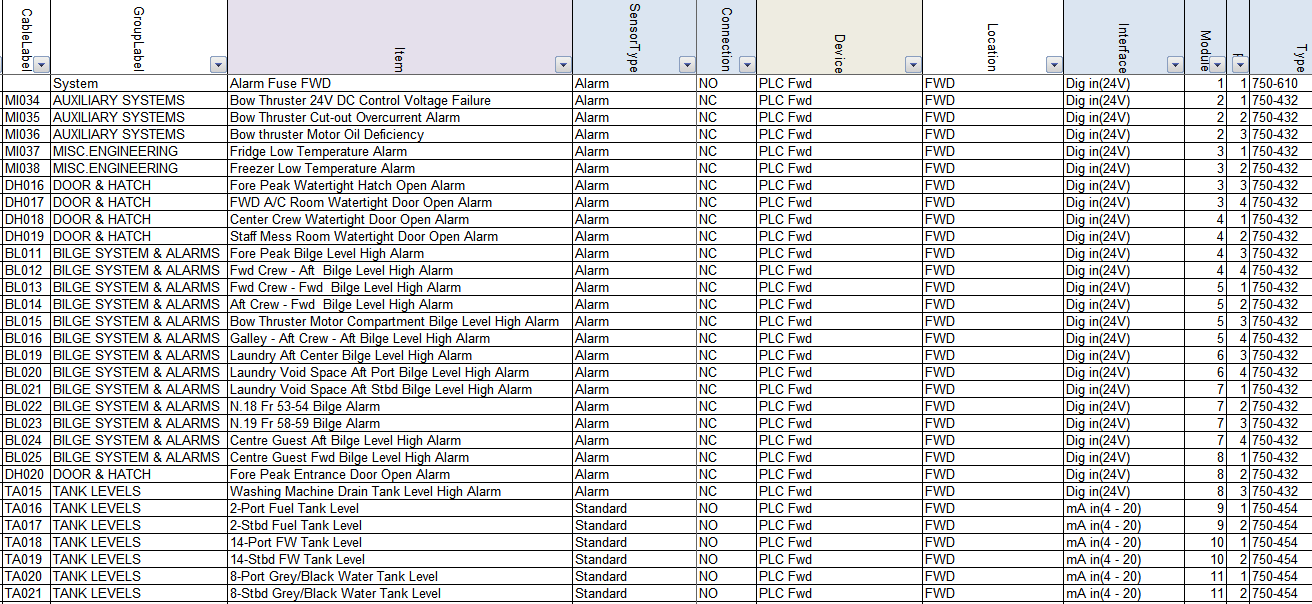


Figure 5‑6: Inserting a Wago slice 1

Now you need to put an extra slice (DI) 750-432 after the 3rd slice in the Wago. If you do that NavVision will see that as a slice without a number and all the fields after slice 3 will go back one slice. You can imagine that is not what we want.

Now let’s do this with the sensorlist. You insert an empty row after the 3rd slice (see Figure 5‑7). Now this will be the 4th slice so at the module column you say it is number 4 and you fill in all the other appropriate fields (see Figure 5‑8).

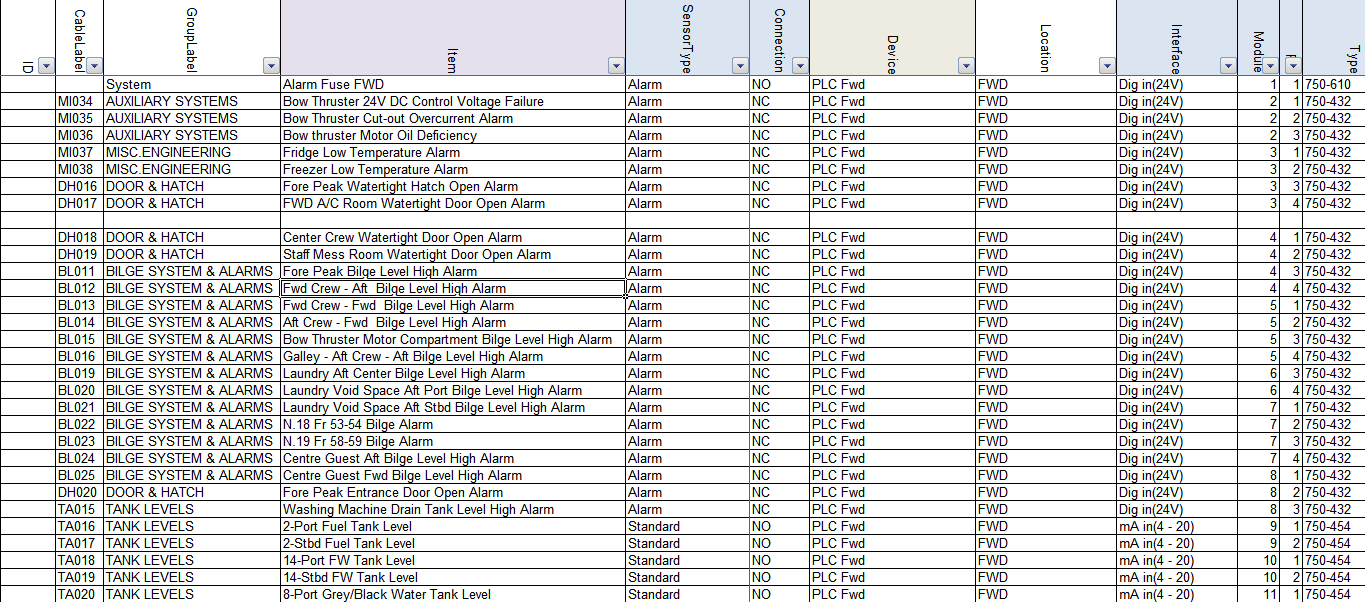


Figure 5‑7: Inserting a Wago slice 2

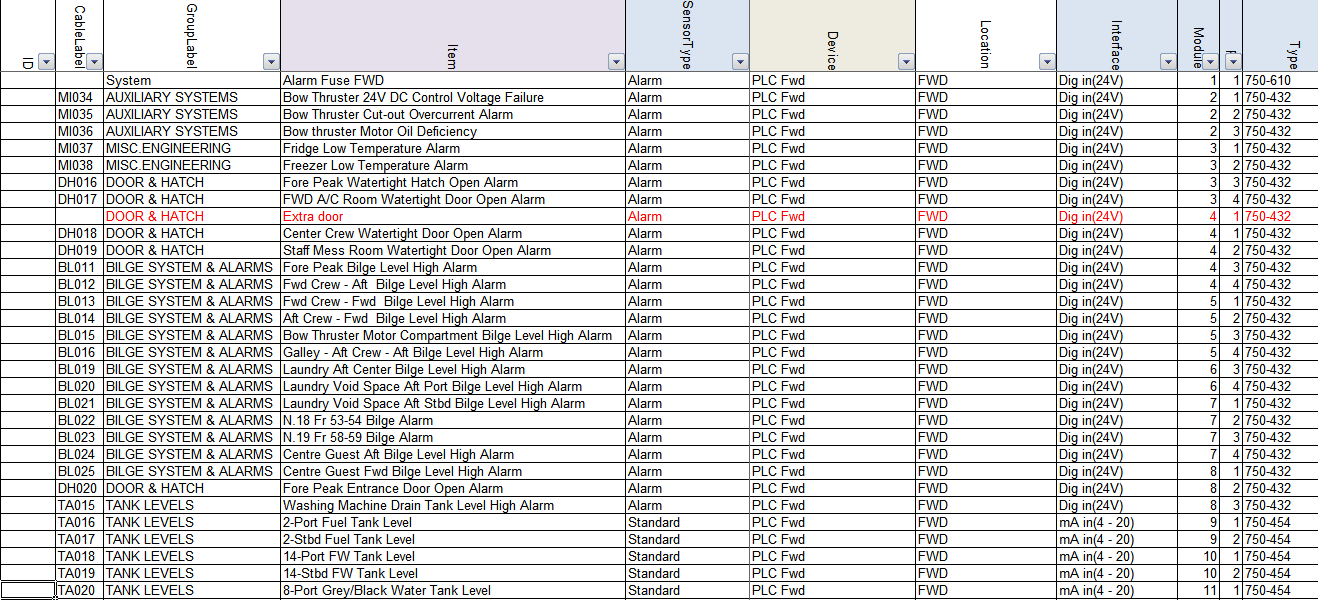


Figure 5‑8: Inserting a Wago slice 3

Now you will have two Wago slices with number 4 so you will need to increase the rest of the module numbers on that Wago. Of course you can do this by hand, but Excel is very helpful in this. Just find a cell with number 1 in it (cause we need to increase the modules by 1) an click CTRL-C to copy the number. Now select all the select all the cells in the module-column that need to be adjusted and right-click. Select “Paste Special” (see Figure 5‑9).

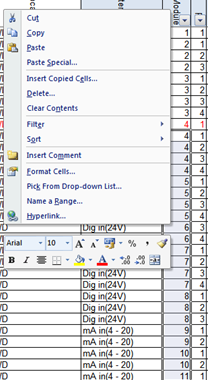


Figure 5‑9: Excel trick 1

In the next window choose “Add” and then click OK (see Figure 5‑10). You will see that all the module numbers has increased by 1.



Figure 5‑10: Excel trick 2

Now you can easily import the sensorlist (after you inserted the new Wago slice) and it will set everything in its right place.

## Keep the sensorlist up to date afterwards

### Introduction

Most likely you will find yourself occupied with work or you will get on board and the crew has made a lot of changes. In both cases it is impossible to use the sensorlist because it probably makes more problems than that it serves you. In that case you need to clean up the sensorlist first. After the clean-up you can use the sensorlist again.

To clean up the sensorlist you need to follow the instructions below. This is, for now, the best way to do this. The bigger the sensorlist is and the more changes, the more time-consuming it will be. But in the end you will only benefit.

### What do you need

You need a complete clean installation of the latest NavVision on your pc/laptop. Keep this one clean and copy your key file (the \*.key.ini) into the folder NavVision/config/network.

If you start at a new project, or wish to make a new beginning, make a new folder and name it after your project. Copy al the files from the clean NavVision folder into your new folder. You will get the following folder:

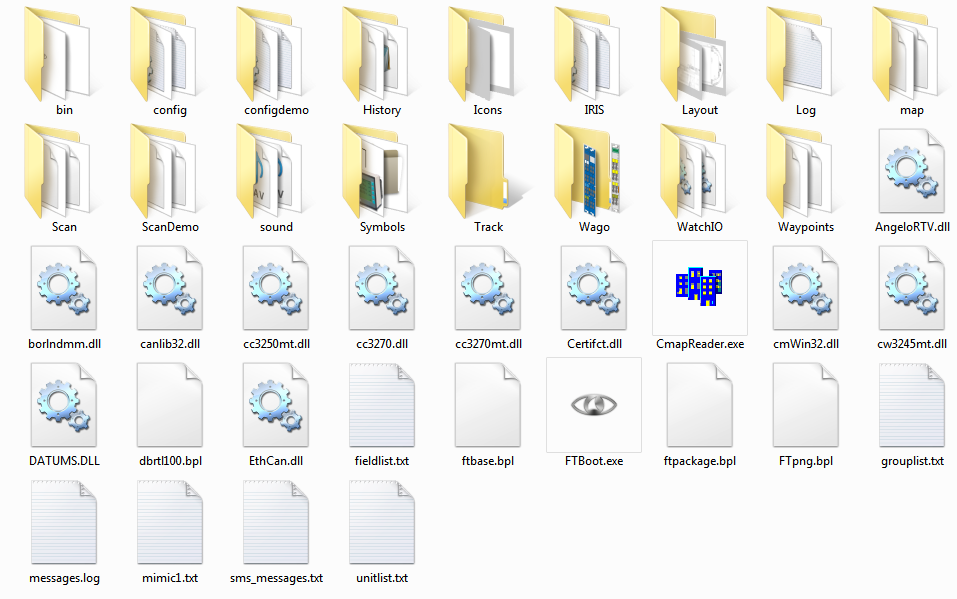


Figure 5‑11: clean NavVision folder

Also you need the config-folder from the installation on board (better back-up the whole NavVision folder). You can do this at the end of the day, when you have finished working on the system, or at a ship that you arrive at for commissioning.

### Cleaning up after a day on board

After you have been on board all day we assume that you have made a backup of the system. Now you do have an existing sensorlist, but we need to find out the changes. Here are the steps you need to take.

#### Copy devicelist.dat and sensorlist.dat

In the backup you took with you from aboard you find two files in the folder NavVision/config/network. These files are:

* Devicelist.dat
* Sensorlist.dat

Now copy these files and paste then in the folder NavVision/config/network of the folder you made on your pc/laptop as in Figure 5‑11. This folder now contains the configuration on board as it was when you left. Don’t start up yet.

#### The old sensorlist

You also have the old sensorlist.xls that you had before you went on board. If you do not already have the file as described, but only the raw sensorlist, we refer you to Chapter 1.4 to see how to save a sensorlist for import.

Copy this sensorlist.xls in to the root of your project folder. It will now look as follows:

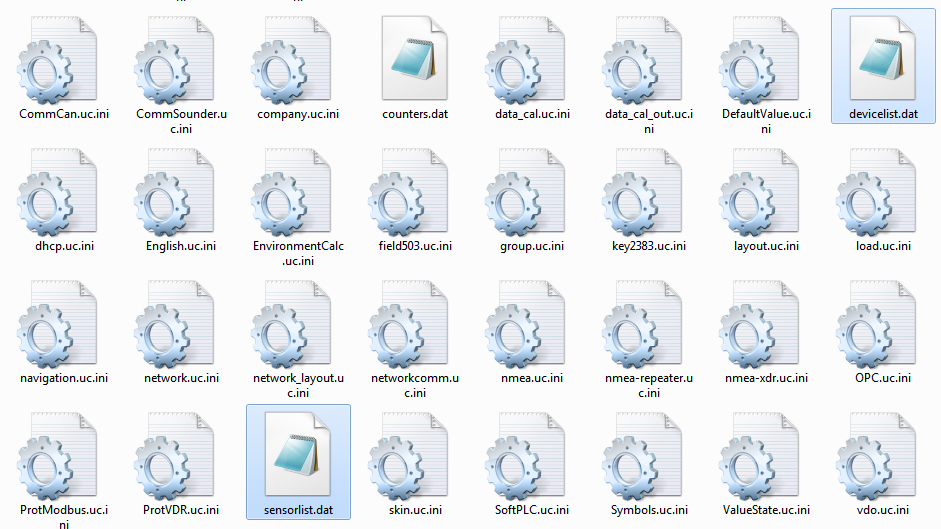


Figure 5‑12: Devicelist.dat and sensorlist.dat in network folder

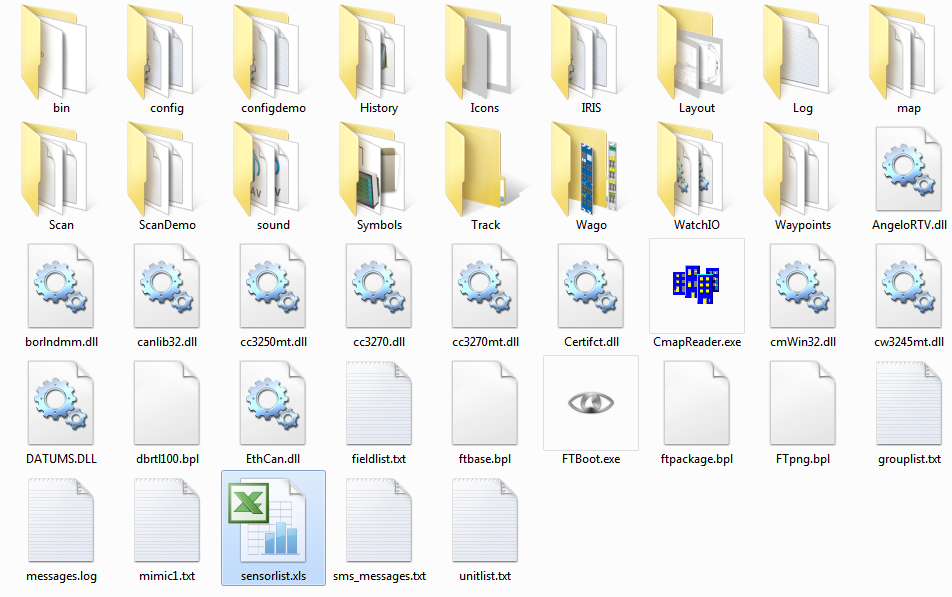


Figure 5‑13: sensorlist.xls in root of project folder

#### Startup your project folder

Now you must start up the NavVision that is in your project folder. To do so, go to the folder NavVision/bin and double-click the NavVision.exe. This way you know that you start the right version.

During startup NavVision will ask you if you want to import the devicelist and after that the sensorlist. Answer both questions with “Yes”. NavVision will start up completely.

After it started up you can shut it down immediately. NavVision will now generate de devices you need. These are:

* devicelist\_generated.html
* sensorlist\_generated.html
* sensorlift\_generated\_diff.html

These files can be found in the root of your project folder which now looks like the following:

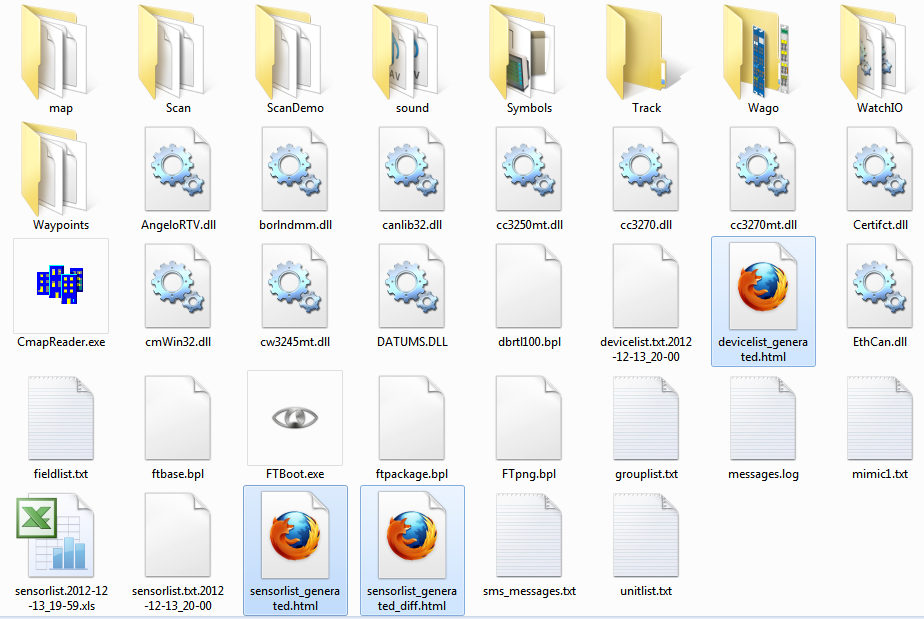


Figure 5‑14: root folder after import sensorlist

#### Inspecting the generated files

What goes for the sensorlist\_generated will also count for the other generated files, so we will only discuss this file here.

Open up the sensorlist\_generated.html (right-click, open with, Microsoft Office Excel). You will now have the sensorlist but also the column ImportResult filled in. If the field is blank than nothing has changed. Just pay attention to the fields that are colored and have a result in it.

This results can be:

|  |  |
| --- | --- |
| Field | Description |
| Comment | Comment that something is different in the field |
| Changed | Notice that something has changed |
| Failed | Critical failure somewhere in the field |
| Missing | Field tag is missing |
| New | Field is added since last import |

Table 5‑1: Import Result fields

This results will almost always be explained by the same color in the row that triggered the code.

Also you can open the sensorlist\_generated\_diff.html to see a reference to the same row (the old value that was there before you imported the sensorlist).

#### Comment

Comment usually indicates a minor problem or no problem at all, but you will need to check them. A simple example is that you see the following line:



Figure 5‑15: Comment example 1

If you look further down the row you will see that the problem is the text “bulb nav light SB 1” as you see in the next figure:



Figure 5‑16: Comment example 2

The fact is that “comment” usually indicates that the text is already in use somewhere in the sensorlist. Also it is possible that the field, in this case “AftNavLightSB” is already in use. Use the search function of Excel to find the text throughout the sensorlist.

In this case we will find that the text and the field is also used in line 71 as showed in the next figure:



Figure 5‑17: Comment example 3

You always have to check closely, but in this case it is fairly easy. Line 44 is the status connection as you will find in the SensorType column and it is connected to a DI-module. Line 71 is Standard connection and is connected to a DO-module. As you know how NavVision works this is no problem. With line 71 you can switch the line on and if the light is on it will give a status back on line 44.

Now you now it is no problem and you can leave the row as is.

*: although it is only a comment, do check all fields for abnormalities. If you are sure it is ok, mark it in the sensorlist.*

#### Changed

Changed indicates that there is a bigger problem. It is a warning. It can be that a value has changed in the min/max settings, or an Item-name is changed or even the interface is changed. Eventually something can be changed in either column.

For your convenience NavVision will show the changed cell in yellow as well. So it is easy to look up. It can even be in multiple cells, so have a good look. See the next figures as example:



Figure 5‑18: Changed example 1



Figure 5‑19: Changed example 2

As you can see there is a yellow colored field that will give you the changed value. In these examples it changed the interface. If you are not sure why it is changed or what was there before, you open up the sensorlist\_generated\_diff.html to see the reference. If we take the second figure as example and we look that up in the sensorlist\_generated\_diff.html, we’ll see the following:



Figure 5‑20: Changed example 3

Now you can check that in NavVision it was defined as mV in(-125-125). As NavVision knows that a Wago 750-469 slice is a Thermo in (K) slice it changed that interface to the right one.

Now that you know that it was changed because of the right reason, you also will have to change it in your sensorlist to keep that up to date.

*: make sure that you check all the changed fields and adjust them accordingly in your sensorlist. It is not possible with a changed field that you leave one unchanged. They all need to be altered in your basic sensorlist.*

#### Failed

Failed is a critical warning. There is something really wrong in that specific line. It can be anything, from missing information to double sensors. You will have to check the line very carefully. Sometimes it will show a red colored cell to show you what is wrong, but other times you will have to dig deeper to find the problem.

Failed always needs to be rectified in your original sensorlist. Here a simple example:



Figure 5‑21: Failed example 1

This is a sensor on a bus-protocol. As you can tell it was put twice in the sensorlist. Bus-protocols can hang on such information, so it is wise, in this case that you remove the Failed line from your original sensorlist.

#### Missing

Missing is an easy one. In this row the field tag is missing. You can go straight to the Field-column and you will find it is empty. See next figure:



Figure 5‑22: Missing example 1

Find the right field as described in Chapter 3.3.19.1 and put that in the original sensorlist.

#### New

Everything that was changed on board and that wasn’t already in the sensorlist will become visible as new. This could be a new sensor on a Wago, but also a complete new device or interface with, for example a bus-protocol.

The next example is when a new device or interface is connected. You will see the following:

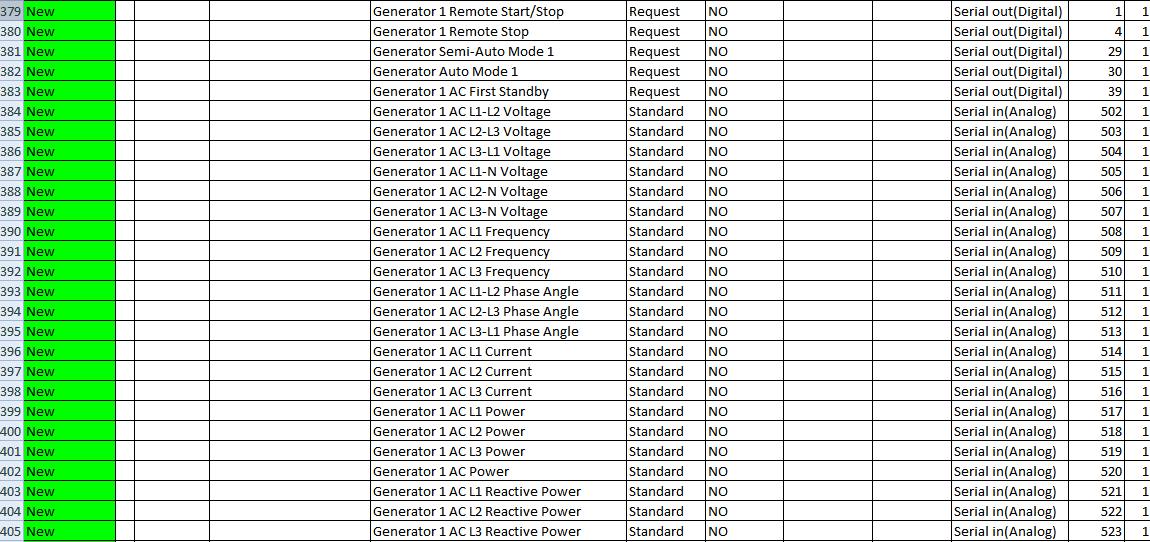


Figure 5‑23: New example 1

You can understand that you have to copy all these lines and paste them into the original sensorlist or they will get lost with a new import.

#### Keep importing

After you checked and replaced all the import results into the original sensorlist, you once again convert it to a sensorlist for import as described in Chapter 1.4 and put it in the root folder of your project folder.

Start NavVision again and import the devicelist and sensorlist. Close NavVision and open the new sensorlist\_generated.html.

If you did well there are no more import results except maybe for a few comments that you left there. If not you will have to repeat this process over and over again until there are no more import results and the sensorlist\_generated\_diff.html is empty.

Once you have reached that point you are finished and your original sensorlist is up to date again.

*: if you arrive on a ship after a long time and the crew has changed a lot, you can follow the same procedures. Just make a backup (or let them send one upfront) and go through all these steps. That way you can start directly with a good and working sensorlist.*

# Special notes

## Introduction

In this chapter we will discuss some special issues or things that are easy to know. It will just be a collection of extra knowledge randomly addressed and will be changed over time.

## PLC

When a PLC program is written and put into the Wago PLC itself it is necessary, especially for the outputs, that NavVision doesn’t have field tags attached. To prevent the PLC program as well as NavVision to address the output on the Wago, you do the following:

The rows in the sensorlist that hold the outputs that already are in use by the Wago PLC program, will need to be adjusted. First you add a “,PLC” after the module number in the column “Type” See the next figure:

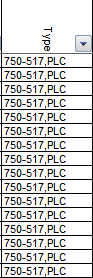


Figure 6‑1: PLC added

This way NavVision knows that the slice is in use by the Wago itself, but will show up in the Wago overview in NavVision see the next figure:

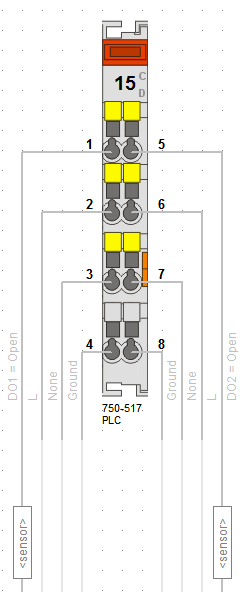


Figure 6‑2: Wago overview PLC

*: Leave the field column empty or it will interfere with the Wago PLC program.*

## Search

When you check the “sensorlist\_generated” you will often find comments. Most of the time it will be that you used the same field-ID in different rows. When you find a comment, go to the column “Type” and copy the field-ID. Press CTRL-F and you get a window where you can search. Paste the copied field-ID and select “Find all”. Now you can scroll through the fields to see if you have used the same Field-ID on multiple rows. If you find it, repair the problem and it will be fixed.

## Setting NMEA in the sensorlist

Since revision 3616 it is also possible to set the NMEA interfaces directly in the sensorlist. This needs an extra explanation cause it works a slightly bit different.

We will focus on the columns that are important. The other columns will all practically work the same as described earlier.

As example we will take a Voith NMEA interface. As you can see in the following figure, the standard columns will be the same as you already learned.

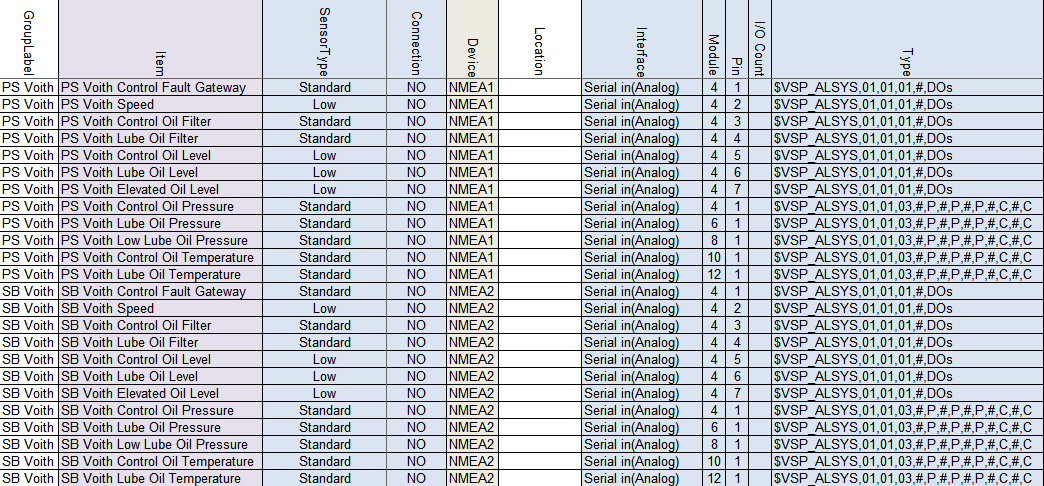


Figure 6‑3: NMEA sensorlist example

Grouplabel, Item, Sensor Type, Connection and Device are the same as described earlier. The alternative columns we’ll describe here.

### Interface

With NMEA you can choose between Serial in(Analog) and Serial out(Analog), depending if you want to receive or send.

### Module

As you will see in the column “type” you set the standard NMEA sentence there. All values are defined between commas in that sentence. To let NavVision know which value you are looking at, you will set the comma after which the value is available in the NMEA sentence. So if you need the value after the 4th comma in the NMEA sentence, you will put a 4 here.

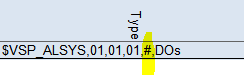


Figure 6‑4: NMEA example 1

*:the “#” sign is just to make it more visible and is not mandatory. You can leave the string without these.*

### Pin

To see which character behind the specific comma you need, under Pin you define the character number. In our example we have on that spot the digital values for the VOITH. So there are 7 zero’s or ones there, each representing one digital input. In our example we define all these values in the first seven rows.

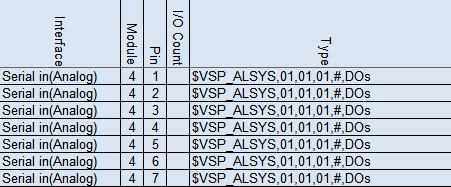


Figure 6‑5: NMEA example 2

*:Make sure that the count column is set to “1” cause you only want to read one character at the time.*

### Type

The Type column is the specific NMEA sentence that you are expecting. Lets analyse a sentence.

$VSP\_ALSYS = talker ID and Sentence Identifier

,01 = digital value

,03 = analog value

,# = wildcard

Or another example:

$GPRMC,220516,A,5133.82,N,00042.24,W,173.8,231.8,130694,004.2,W\*70

$GPRMC = talker ID and Sentence Identifier

,220516 = time stamp

,A = valid or ”V” invalid

Etc.

If you know the characters that are needed, you can fill it in.



The “P” and “C” represent Pressure and Celcius.

*:make sure that if you have an analog value, you set the Count column to the right amount of characters to read. Default is 16, which should be enough in most cases.*

### Count

At the count column you specify how many characters you will read at maximum on that specific location. So for digital values that will be 1. For analog values you will have to look at the original NMEA sentence. It can be that you need to read 4 characters max or 6. Whatever max number of characters you find for that field, you will define here at “count”.

*: The column “Data Type” is necessary if you send NMEA data. You will set the right parameter. If NavVision is reading NMEA data it will ignore “Data Type” so you can leave it blank.*

### Count

At the count column you specify how many characters you will read at maximum on that specific location. So for digital values that will be 1. For analog values you will have to look at the original NMEA sentence. It can be that you need to read 4 characters max or 6. Whatever max number of characters you find for that field, you will define here at “count”.

## WatchIO in the sensorlist

Since revision 3904 The new setup of implementing WatchIO is in effect. WatchIO is the main protocol of the Unimacs bridges.

We will focus on the columns that are important. The other columns will all practically work the same as described earlier.

First make sure that the devices are set in the Devicelist. NavVision has to know where to look for the device. The following figure shows the mandatory fields for the devicelist:

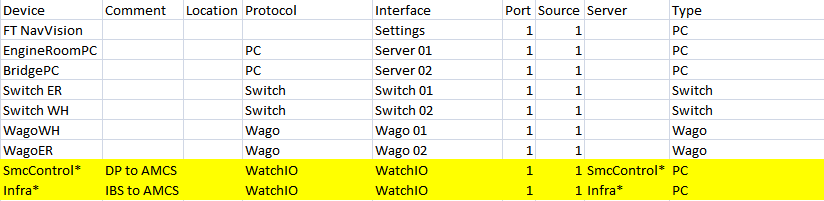


Figure 6‑6: Devicelist WatchIO

\

|  |  |
| --- | --- |
| Field | Description |
| Device | The name of the WatchIO view the asterix (\*) at the end is needed for redundancy |
| Comment | Free text |
| Location | Free location name |
| Protocol | Protocol must be WatchIO |
| Interface | Interface must be WatchIO |
| Port | Every protocol has to have a unique port number |
| Source | Mostly one |
| Server | Same as Device |
| Type | Type must be PC |

Table 6‑1: Devicelist WatchIO

In the sensorlist we also have a slightly different approach. While the main fields will be the same we will focus on the differences by the following example”

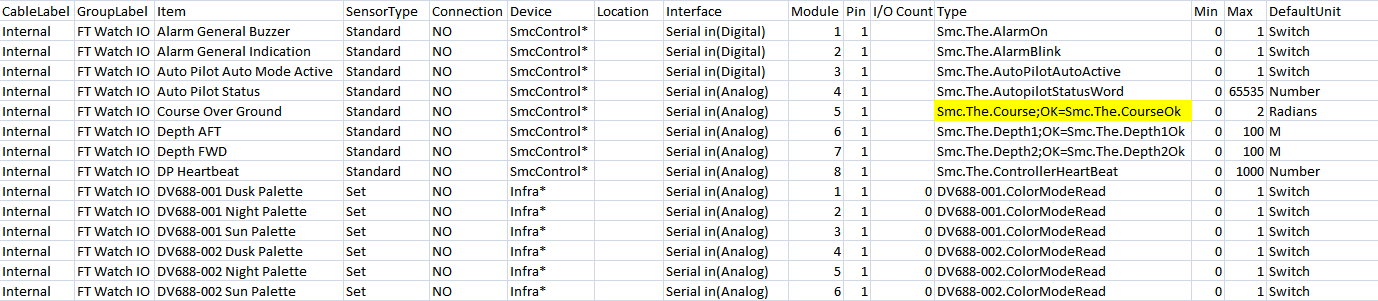


Figure 6‑7: Sensorlist WatchIO

|  |  |
| --- | --- |
| Field | Description |
| Cable label | Free (preferably “Internal”) |
| Group Label | Free |
| Sensortype | Preferably “Standard” only when you want to set something or you just need a part of the variable. For the last one use “Enum” |
| Connection | NO |
| Device | Choose the one from the devicelist |
| Location | Free |
| Interface | Serial in Analog or Digital |
| Module | A unique number. Start with 1,2,3…..etc. |
| Pin | 1 |
| I/O count | Free |
| Type | Here you need to put the type Variable. You can get that from the Unimacs program. For status you also put the OK-Variable behind a semi-colon. |
| Min | Normal Min settings |
| Max | Normal Max settings |
| DefaultUnit | Normal DefaultUnit |

Table 6‑2: Sensorlist WatchIO

The rest of the fields is to be treated the same as mentioned earlier in this document.

*: for further configuration refer to the “ACC-Software-Installation-and-Commissioning-Manual”*

## CANopen in the sensorlist

Since revision 3792 it is possible to select the CANopen protocol in NavVision. It is also possible to configure it in the sensorlist. CANopen is an application layer and communication profile that is used in several marine industries.

*: make sure that you read the “Software installation and commissioning manual”. The ICP needs some additional configuration to make the communication work.*

We will focus on the columns that are important. The other columns will all practically work the same as described earlier.

As example we will take a Naiad CANopen interface. As you can see in the following figure, the standard columns will be the same as you already learned.

First make sure that you select “CANopen” as the protocol in the Devicelist. This will make NavVision aware to expect the CANopen-protocol on that port.

### By hand

Make sure that you get the latest version of the manual from the manufacturer. It will give you the information you need to fill in to the sensorlist. In our example we use a manual from Naiad.

We are looking for the message details.

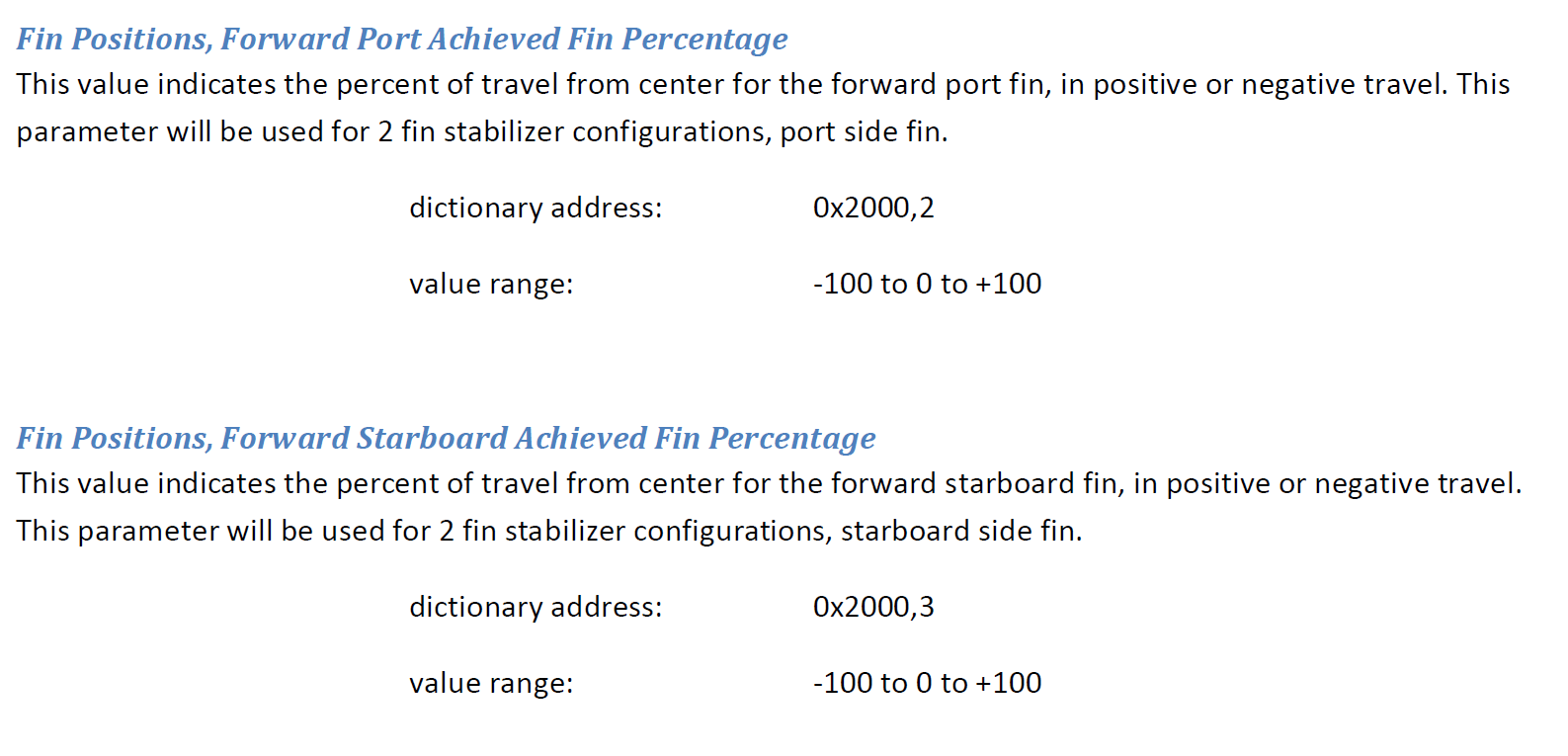


Figure 6‑8: Naiad manual

This information we are going to use to fill the column “Module” in the sensorlist. As you see it consists of structures with an ID. If, for example you want to read the “Fin position” you notice that the structure is 0x2000. For the forward Port one the ID is 2, for the Forward Starboard one the ID is 3. This will result in the value 0x200002 or 0x200003 that you will need to put into the “Module” column. This way you can find all the information on the I/O.

You still need to fill in the other columns as usual. This will result in a sensorlist as following:

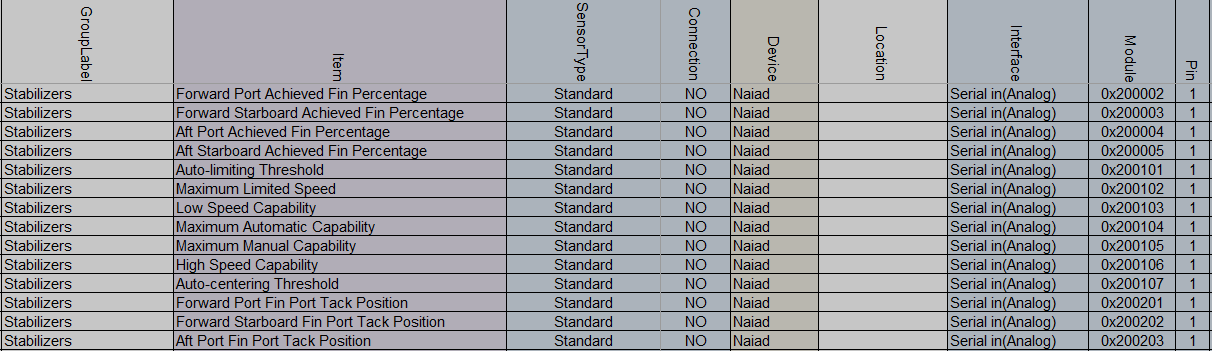


Figure 6‑9: Example CanOpen

Don’t forget to fill in the rest of the columns such as Field etc.

### By import

If the manufacturer can send you an EDS-file of the protocol (i.e. Naiad.eds), you can import the data into NavVision.

Rename the EDS-file to the network-connection it is attached to (i.e. canopen1\_1.eds) and put it in the rootfolder of NavVision. Start and close-down NavVision and the structure will be available in the sensorlist\_generated. Take care that it only goes for the column “Module” so you have to fill all the other mandatory columns by hand.

*: The structure of the eds file for import is canopen1\_1, or canopen1\_2, or canopen2\_1. The first value is the interface-index (what number is the interface, i.e. 1 for the first ICP and so on). The second value is the source-ID. This is most likely 1, but depending on what they have set it could be any number.*