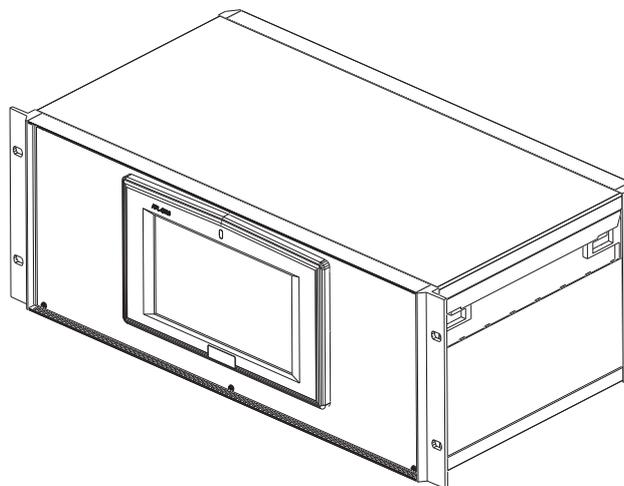
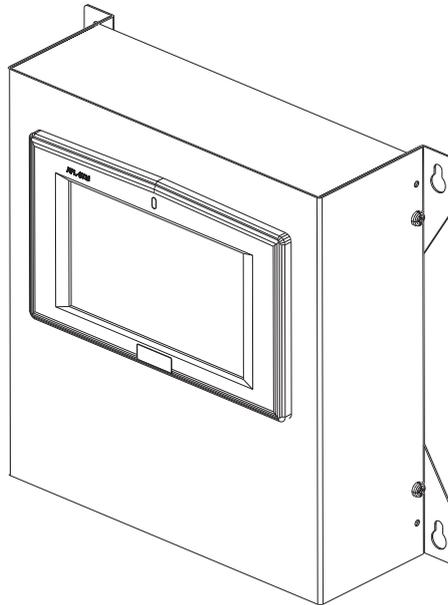


# Operating Instructions

## Tankvision Multi Scan NXA83B

### DCC Communications Configuration



# Table of Contents

- 1 Document information ..... 3**
  - 1.1 Target audience for this manual ..... 3
  - 1.2 Version history ..... 3
  - 1.3 Document function ..... 3
  - 1.4 Documentation ..... 4
- 2 Basic safety instructions ..... 5**
  - 2.1 Requirements for the personnel ..... 5
  - 2.2 IT security ..... 5
  - 2.3 Designated use ..... 5
  - 2.4 Workplace safety ..... 6
  - 2.5 Operational safety ..... 6
  - 2.6 Product safety ..... 6
- 3 Data Communications Controller ..... 7**
- 4 DCC Config Tool ..... 8**
  - 4.1 Stopping and Starting DCC Services ..... 8
  - 4.2 Configuring a Host or Slave port ..... 9
  - 4.3 Configuring the database connection details .. 10
  - 4.4 Viewing the license details ..... 16
- 5 DCC Host Configuration ..... 18**
  - 5.1 DCC Host ..... 18
  - 5.2 Configuring Host COM Ports ..... 18
  - 5.3 Configuring Host Modem Ports ..... 21
  - 5.4 Configuring Host Network Ports ..... 24
  - 5.5 Configuring OPC Network Ports ..... 27
  - 5.6 Configuring Scheduled Connections ..... 29
  - 5.7 Redundancy ..... 31
  - 5.8 DCC Options ..... 33
  - 5.9 Modbus TG protocol configuration ..... 33
- 6 DCC Host Runtime ..... 45**
  - 6.1 DCC Host ..... 45
  - 6.2 Comms Monitor ..... 45
  - 6.3 Watching a Tank ..... 47
  - 6.4 Viewing Modem Log ..... 48
  - 6.5 Connection Status ..... 49
  - 6.6 Enhanced Raw Data ..... 51
- 7 DCC Slave Configuration. .... 54**
  - 7.1 DCC Slave ..... 54
  - 7.2 Configuring Slave COM Ports ..... 54
  - 7.3 Configuring Modem Ports ..... 57
  - 7.4 Configuring Network Ports ..... 57
  - 7.5 Protocol Configuration ..... 58
  - 7.6 Modbus Protocol Configuration ..... 61
  - 7.7 Enraf Protocol Configuration ..... 85
  - 7.8 Extended Modbus Configuration ..... 92
  - 7.9 Datacon Slave Configuration ..... 92
  - 7.10 Relay Contact Output Configuration ..... 93

- 7.11 TankMaster Configuration ..... 99

- Index..... 103**

# 1 Document information

## 1.1 Target audience for this manual

This manual is giving detailed information on the system capabilities and architecture. It supports project and sales engineers in designing the system architecture during acquisition and execution phase. Furthermore during operation time of the system all servicing personnel in need of detailed knowledge about the system capabilities.

## 1.2 Version history

Document version	Valid for SW version	Changes to the previous version
BA01292G/00/EN/01.14	3.0.10	Initial version
BA01292G/00/EN/02.15	3.0.12	Change of order code structure

## 1.3 Document function

### 1.3.1 Used symbols

#### Safety symbols

Symbol	Meaning
 <small>A0011189-EN</small>	<b>DANGER!</b> This symbol alerts you to a dangerous situation. Failure to avoid this situation will result in serious or fatal injury.
 <small>A0011190-EN</small>	<b>WARNING!</b> This symbol alerts you to a dangerous situation. Failure to avoid this situation can result in serious or fatal injury.
 <small>A0011191-EN</small>	<b>CAUTION!</b> This symbol alerts you to a dangerous situation. Failure to avoid this situation can result in minor or medium injury.
 <small>A0011192-EN</small>	<b>NOTICE!</b> This symbol contains information on procedures and other facts which do not result in personal injury.

#### Symbols for certain types of information

Symbol	Meaning
 <small>A0011193</small>	<b>Tip</b> Indicates additional information.
 <small>A0011195</small>	<b>Reference to page</b> Refers to the corresponding page number.
1. , 2. , 3. ...	Series of steps
 <small>A0018373</small>	Result of a sequence of actions

## Symbols in graphics

Symbol	Meaning
1, 2, 3 ...	Item numbers
1., 2., 3. ...	Series of steps
A, B, C ...	Views
	<b>Hazardous area</b> Indicates a hazardous area.
	<b>Indicates a non-hazardous location</b> Safe area (non-hazardous area)

## 1.4 Documentation

### 1.4.1 Operating instructions

Document number	Instrument	Type of Document
BA01288G/00	<ul style="list-style-type: none"> <li>▪ Multi Scan NXA83B</li> </ul>	Operation manual
BA01290G/00		Installation and Maintenance manual
BA01291G/00		Configuration manual
BA01292G/00		DCC Configuration manual
BA01289G/00		OPC Tank Data Server manual
BA01297G/00		Web Client System Operation manual
BA01287G/00		Weights and Measures Additions
BA01296G/00		Redundancy manual

## 2 Basic safety instructions

### 2.1 Requirements for the personnel

The personnel for installation, commissioning, diagnostics and maintenance must fulfill the following requirements:

- Trained, qualified specialists: must have a relevant qualification for this specific function and task
- Are authorized by the plant owner/operator
- Are familiar with federal/national regulations
- Before beginning work, the specialist staff must have read and understood the instructions in the Operating Instructions and supplementary documentation as well as in the certificates (depending on the application)
- Following instructions and basic conditions

The operating personnel must fulfill the following requirements:

- Being instructed and authorized according to the requirements of the task by the facility's owner operator
- Following the instructions in these Operating Instructions

### 2.2 IT security

We only provide a warranty if the device is installed and used as described in the Operating Instructions. The device is equipped with security mechanisms to protect it against any inadvertent changes to the device settings.

IT security measures in line with operators' security standards and designed to provide additional protection for the device and device data transfer must be implemented by the operators themselves.

Endress+Hauser can be contacted to provide support in performing this task.

### 2.3 Designated use

#### 2.3.1 Applications

##### **Inventory control**

By using Tankvision Multi Scan to monitor the tank level and stored volume of valuable liquids remotely, owners or operators of tank farms or terminals for petroleum products and chemicals (liquids) can visualize the volume of the stored medium in real time. The data can be used to plan the inventory and distribution. The data can also be used to manage tank farm operations like pumping or transferring products. Tankvision has its unique concept using network technology. Without using proprietary software, the users can visualize and manage their valuable liquids stored in the tanks by a web browser. Tankvision Multi Scan is a flexible and cost effective solution due to its scalable architecture. The application coverage goes from small depots with only a few tanks up to refineries.

##### **Inventory Calculations**

Tankvision Multi Scan calculates based on measured variables and tank capacity tables:

- Observed/Gross volumes,
- Net volumes and
- Mass

of products like

- Hydrocarbons,
- Liquefied gases,
- Asphalt.

They are corrected according to international standards, including API/ASTM tables 5A, 5B/6, 53A, 53B/54, 23/24, LPG.

This includes temperature corrections at 15 °C, 60 °F and alternative temperatures.

Additionally, available pumpable volumes and water volume are calculated.

### **Remote configuration of measuring equipment**

Some on-site operations can be avoided using remote configuration of measuring equipment during commissioning or maintenance (the availability of this feature may depend on the system configuration).

### **Application areas**

- Tank farms in refineries
- Ship loading terminals
- Marketing and distribution terminals
- Pipeline terminals
- Logistic terminals for tanks storing products like crude oils, refined white and black products, chemicals, LPG

## **2.4 Workplace safety**

For work on and with the device:

- Wear the required personal protective equipment according to federal/national regulations.
- Switch off the supply voltage before connecting the device.

## **2.5 Operational safety**

Risk of injury!

- Operate the device in proper technical condition and fail-safe condition only.
- The operator is responsible for interference-free operation of the device.

### **Conversions to the device**

Unauthorized modifications to the device are not permitted and can lead to unforeseeable dangers

- If, despite this, modifications are required, consult with Endress+Hauser.

### **Repair**

To ensure continued operational safety and reliability,

- Carry out repairs on the device only if they are expressly permitted.
- Observe federal/national regulations pertaining to repair of an electrical device.
- Use original spare parts and accessories from Endress+Hauser only.

## **2.6 Product safety**

The device is designed to meet state-of-the-art safety requirements, has been tested and left the factory in a condition in which it is safe to operate. The device complies with the applicable standards and regulations as listed in the EC declaration of conformity and thus complies with the statutory requirements of the EG directives. Endress+Hauser confirms the successful testing of the device by affixing to it the CE mark.

### 3 Data Communications Controller

The Data Communications Controller (DCC) sub-system consists of a number of services which are responsible for transmitting tank data to and from the PC.

The DCC sub-system is required for both standalone and client server configurations. In a standalone environment all DCC services run on the same PC as the client modules. In a client server environment all DCC services run on the PC dedicated as the database server.

The three main services are:

- **DCC Host** – responsible for retrieving data from gauges and storing it in the database
- **DCC Slave** – responsible for sending tank data to other computer systems
- **OPC Server** – responsible for providing tank data to OPC clients

The first two are automatically started when the PC is booted, whilst the OPC Server is started when an OPC client tries to connect to the tank gauging system.

There are a number of other services that are grouped within the DCC sub-system but are not concerned with data communications:

- **Hermes** – responsible for synchronising data between an in-memory real time database (RTDB) and the main SQL Server database
- **RTDB Accessor** – this service is used by other services ( such as the Events Server ) to access the RTDB database directly
- **Events Server** – responsible for distributing alarms/events to any Alarm Event Viewer displays running on this system or on client machines

The Hermes and RTDB Accessor services require no configuration.

After the DCC has been configured there is relatively little interaction between it and the users. However some maintenance operations may be required from time to time.

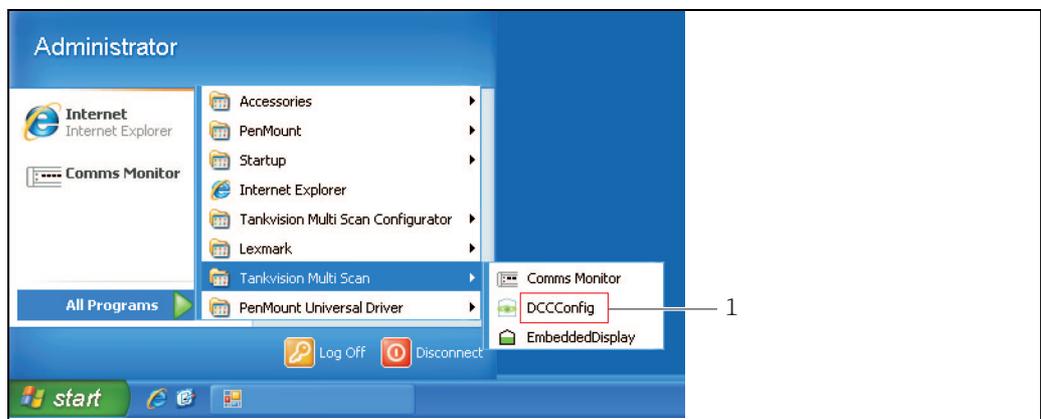
## 4 DCC Config Tool

The various services which make up the DCC sub-system are controlled and configured using a program called DCC Configuration, which can be started from Windows desktop or from the menu item:

**Windows Start → All Programs → Tankvision Multi Scan → DCCConfig**



1 DCCConfig Desktop shortcut icon



1 Windows menu item: DCCConfig

On older systems this will instead be located as an icon in the standard windows Control Panel.

DCC Config consists of a number of tabbed pages which can be used to start/stop services, add serial ports and configure database links. The different screens may be viewed by clicking on the tab for the desired page. Changes may be saved using either the **Apply** or **OK** buttons.

All changes will be automatically saved and the services will automatically re-configure themselves with the new data.

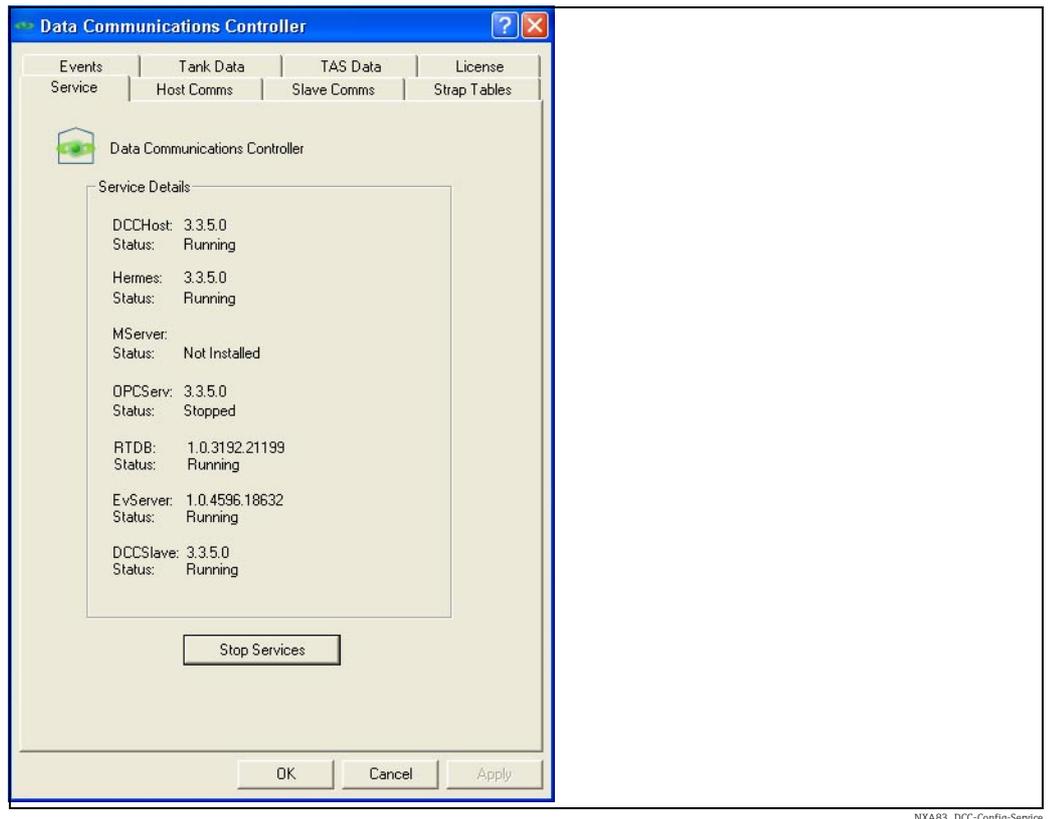
Care should be used when editing parameters in these screens, as wrong data could stop services from operating correctly.

Only system administrators or similar should use the DCC Config tool.

### 4.1 Stopping and Starting DCC Services

The DCC runs as a set of services, therefore when the PC is restarted they are automatically started. However there may be times when it is required to stop or restart the services.

To Stop the DCC services, double click the DCC Config icon in Control Panel, then select the **Service** tab.



NXA83\_DCC-Config-Service

A list of all the service that comprise the DCC communications sub-system will be displayed, showing the current status of each service.

To stop the DCC services click on the **Stop Services** button.

When all Service Details have changed to Stopped the button will change to **Start Services**.

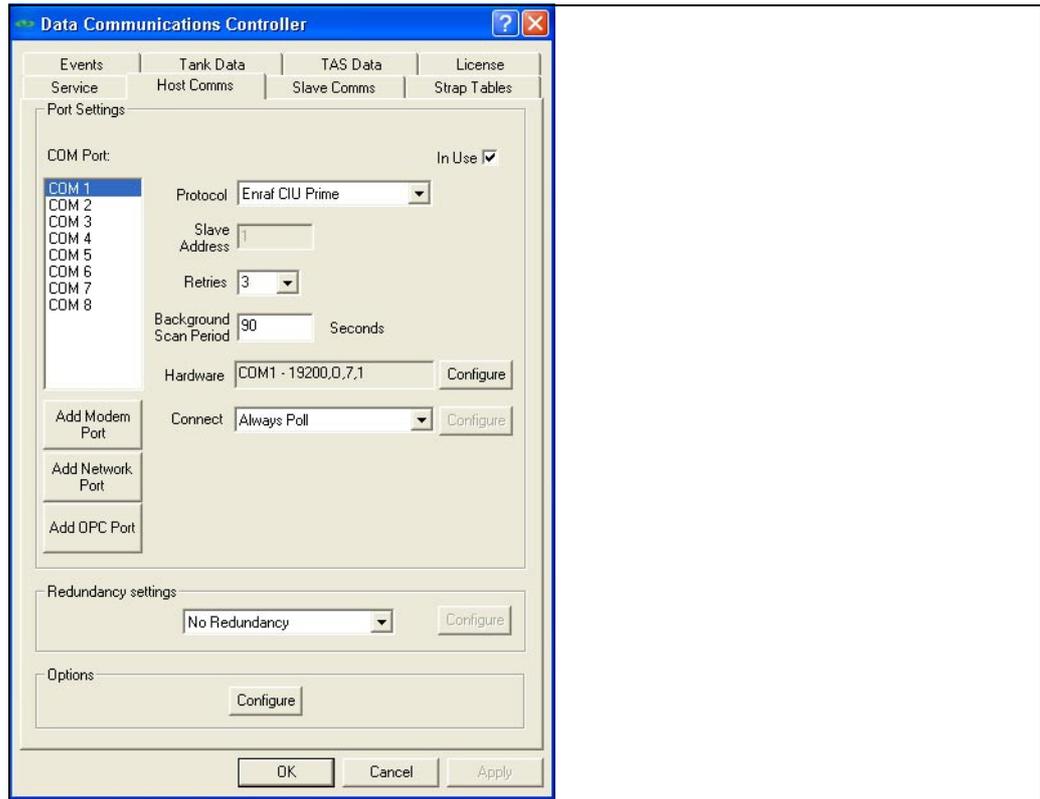
To start the DCC Host again simply click the **Start Services** button. After a few moments all installed services should change to Running.

## 4.2 Configuring a Host or Slave port

Host ports are responsible for collecting data from Datacons and gauges, calculating Tank Inventory, and processing command requests from the system client.

Slave ports are responsible for sending data to other computer systems, such as the main DCS for the refinery, using protocol such as Modbus or Enraf.

To configure a port click on the **Host Comms** or **Slave Comms** tab in the DCC Config tool.



NXAB3\_DCC-Config\_Host-Comms

For details on configuring the ports, refer to later sections in this manual.

### 4.3 Configuring the database connection details

The DCC sub-system uses the SQL database to store configuration data and live data returned from the gauges.

There are a number of tab screens in DCC Config which allow the SQL database connection to be configured.

#### NOTICE

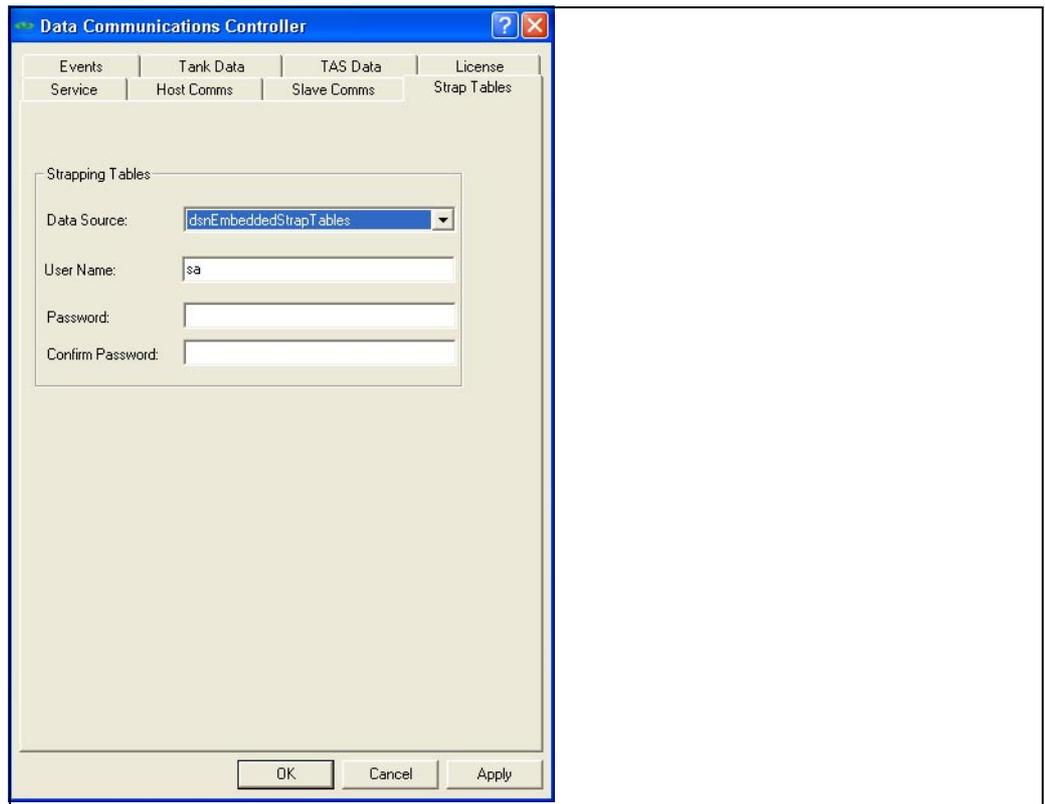
**The configuration details are automatically setup during installation. Any changes of the configuration details could stop the correct operation of the whole tankgauging software. Therefore:**

- ▶ Do not change the configuration details during operation.

The following configuration details are detailed here simply for information purposes and in cases where data may have become corrupted etc.

#### 4.3.1 Strap Tables

This screen is used to configure the data source for the strapping tables. This is a database that stores the strapping data for each tank as a separate table, and is used by DCC Host to load the strap data for each tank.



NXA83\_DCC-Config\_Strap-Tables

The data source for the strapping tables should never require configuring as the parameters will be setup during the installation of the system. However, if for some reason the data is wrong or corrupted, then the data source can be selected from a list of data sources configured in the PC.

The standard default data items are:

- **Data Source** – "dsnEmbeddedStrapTables"
- **User Name** – "sa"
- **Password** – gh3 | 1966

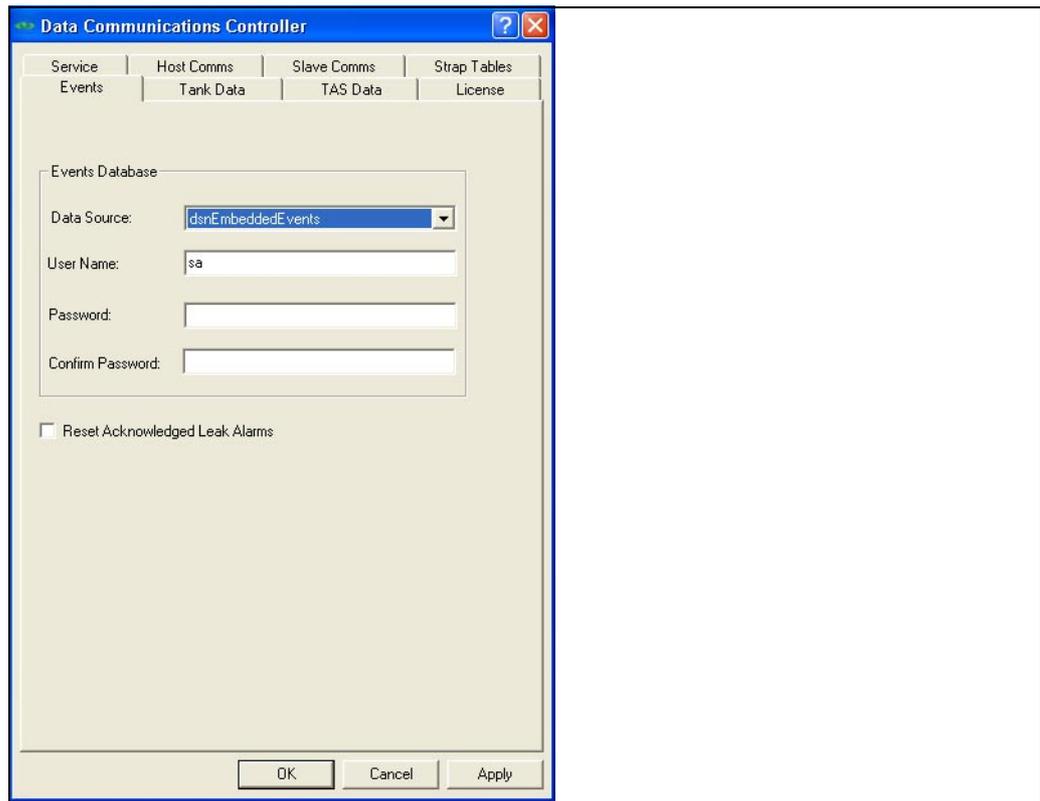
#### **NOTICE**

**Entering incorrect parameters could stop DCC Host from calculating volumes or could even force it to shut down. Therefore:**

- ▶ Take extreme care when entering the parameters.

### 4.3.2 Events

This screen is used to configure the data source where any events (i.e. alarms) generated by the system will be stored. This is a database on the hard disk that holds all events and alarms in a single table. DCC Host saves any alarms it generates in this database which may then be displayed on the applications client screens.



NXA83\_DCC-Config\_Events

The events data source should never require configuring as the parameters will be setup during the installation. However, if for some reason the data is wrong or corrupted, then the data source can be selected from a list of data sources configured in the PC.

The standard default data items are:

- **Data Source** – "dsnEmbeddedEvents"
- **User Name** –"sa"
- **Password** – gh3 | 1966

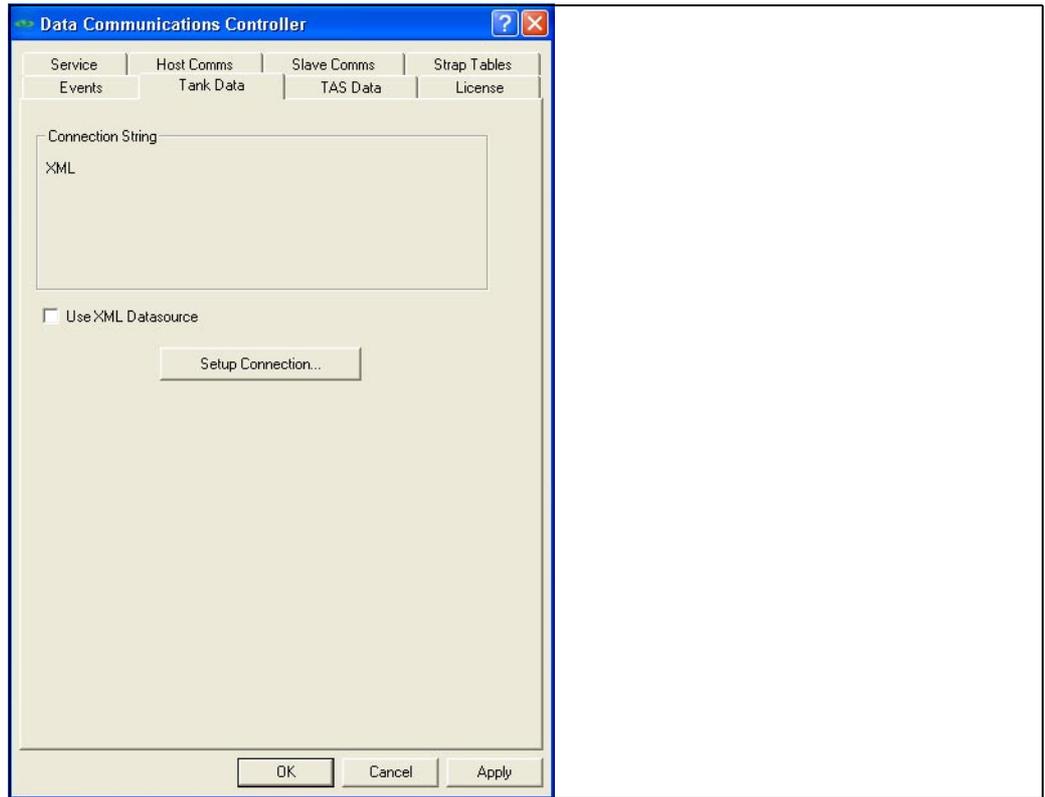
#### **NOTICE**

**Entering incorrect parameters could stop events being generated by DCC Host, including safety features such as gauge alarms etc. Therefore:**

- ▶ Take extreme care when entering the parameters.

### **4.3.3 Tank data**

This screen is used to configure the connection details for the main database where all tank data will be stored on the hard disk. This includes data from DCC Host such as levels, temperatures, volumes etc. It is then used by the system clients to display the tank data to the users.



NXA83\_DCC-Config\_Tank-Data

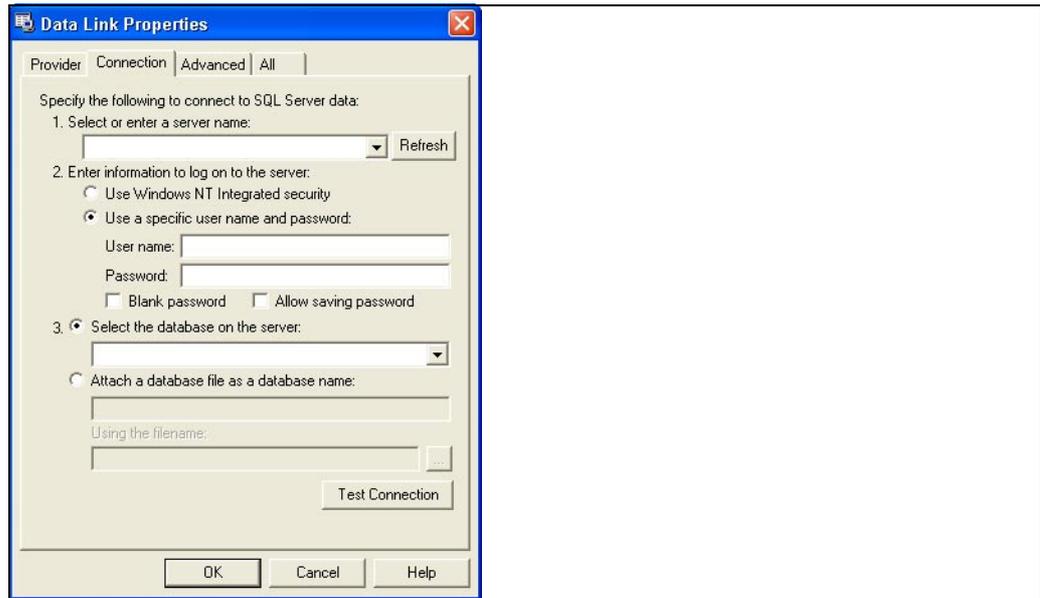
The Tank Data connection details should never require configuring as the parameters will be setup during the installation. However, if for some reason the data is wrong or corrupted, then a new connection can be set up by clicking on the **Setup Connection** button.

For certain products an XML database is used, instead of the standard SQL database. In this case the **Use XML Datasource** item will be ticked. If this item is not already ticked then leave it unticked, as this type of datasource is only used on embedded products, which will have already been setup before being shipped.

#### NOTICE

**Existing details are not saved and the system operation could be damaged, when the Setup Connection button is clicked. Therefore:**

- ▶ DO NOT click on the **Setup Connection** button unless the connection details definitely require changing.



NXA83\_DCC-Config\_Tank-Data\_Data-Link-Properties

This screen allows a new database connection to be established and saved. The exiting connection data will NOT be shown on this screen, so all details will have to be entered afresh.

The default settings (which will be the same for 99% of installations) are:

1. **Select or enter a server name** – "(local)"
2. **Enter information to log on to the server** – "Use a specific user name and password",  
User name - "sa", Password – gh3 | 1966
3. **Select the database file as a database name** - "Tanks"

If in doubt about any options, click on the **Cancel** button to abort, do not click on **OK**.

#### **NOTICE**

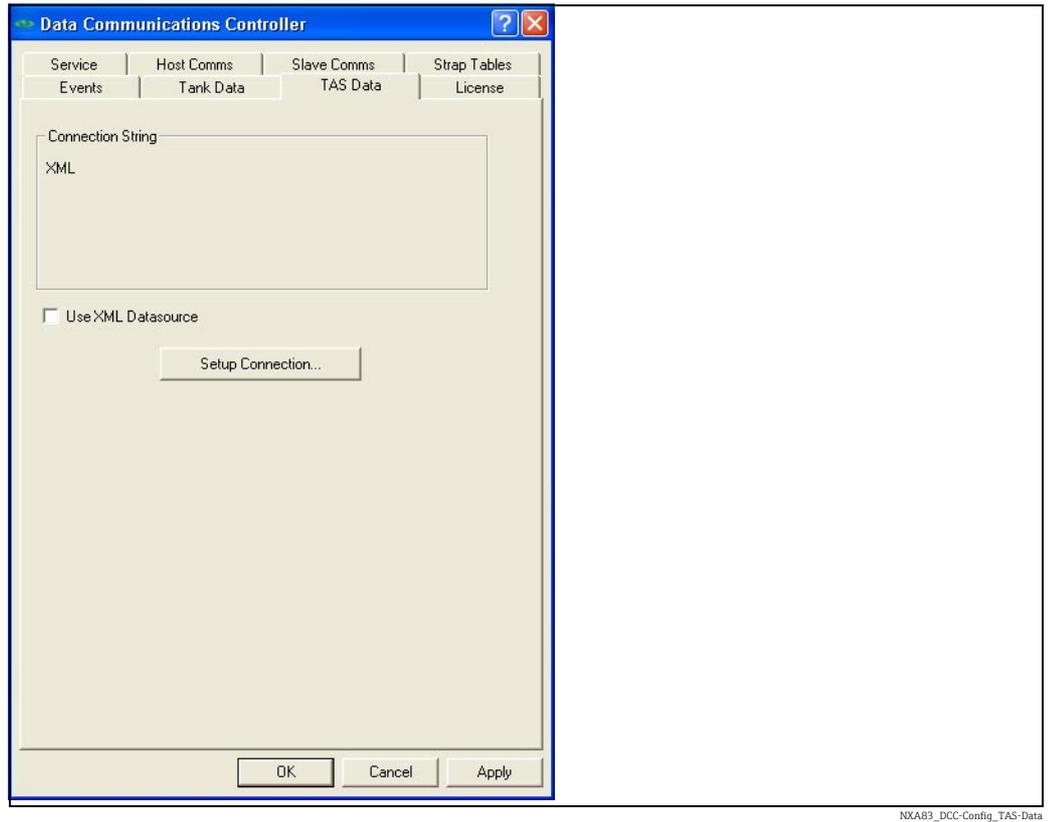
**Clicking on the OK button will save any data entered here, even if it was left blank. This can cause the connection string to be invalid and stop DCC functioning correctly.**

**Therefore:**

- ▶ Take extreme care when changing the settings.

#### **4.3.4 TAS Data**

This screen is used to configure the data source for the terminal automation system (if installed).



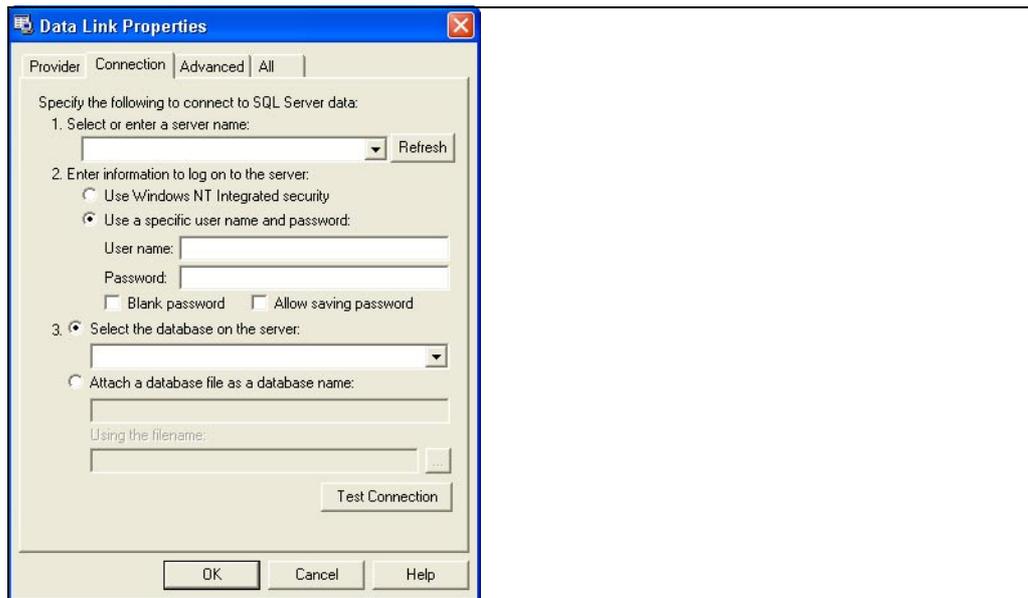
The TAS Data connection details should never require configuring as the parameters will be setup during the installation of the communications components. However, if for some reason the data is wrong or corrupted, then a new connection can be set up by clicking on the **Setup Connection** button.

For certain products an XML database is used, instead of the standard SQL database. In this case the **Use XML Datasource** item will be ticked. If this item is not already ticked then leave it unticked, as this type of datasource is only used on embedded products, which will have already been setup before being shipped.

#### **NOTICE**

**Existing details are not saved and the system operation could be damaged, when the Setup Connection button is clicked. Therefore:**

- ▶ DO NOT click on the **Setup Connection** button unless the connection details definitely require changing.



NXA83\_DCC-Config\_TAS-Data\_Data-Link-Properties

This screen allows a new database connection to be established and saved. The exiting connection data will NOT be shown on this screen, so all details will have to be entered afresh.

The default settings (which will be the same for 99% of installations) are:

1. **Select or enter a server name** – "(local)"
2. **Enter information to log on to the server** – "Use a specific user name and password",  
User name - "sa", Password – gh3 | 1966
3. **Select the database file as a database name** - "BookStock"

If in doubt about any options, click on the **Cancel** button to abort, do not click on **OK**.

#### **NOTICE**

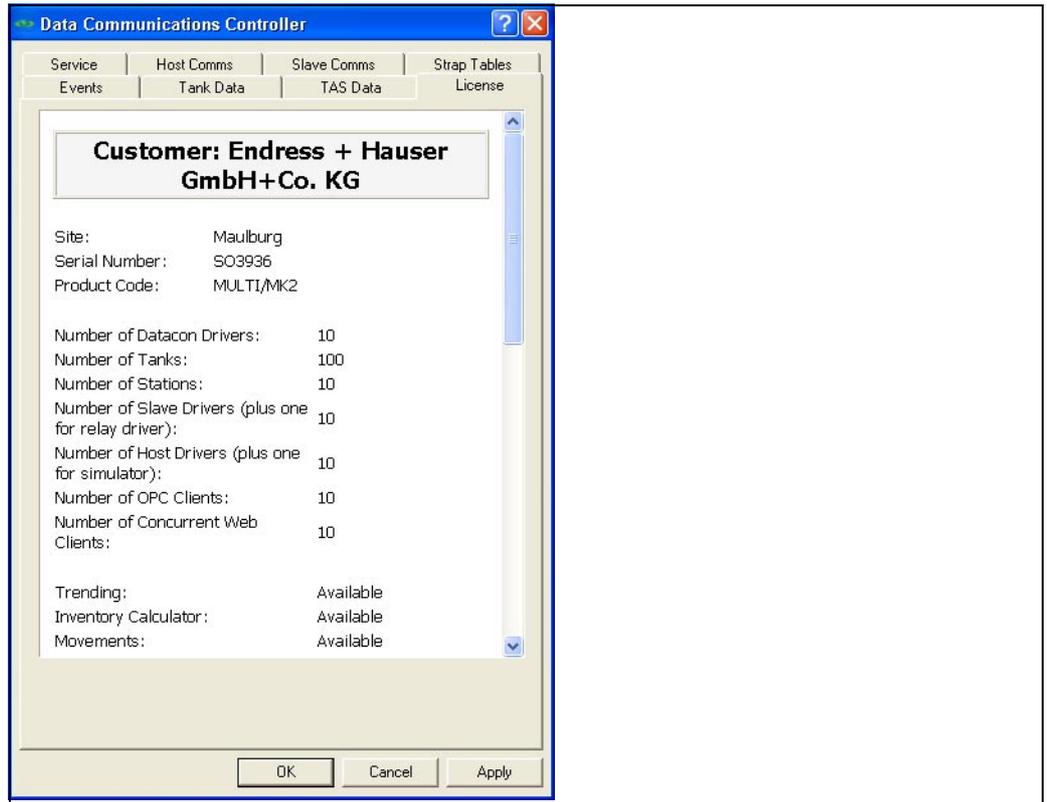
**Clicking on the OK button will save any data entered here, even if it was left blank. This can cause the connection string to be invalid and stop DCC functioning correctly.**

**Therefore:**

- ▶ Take extreme care when changing the settings.

## 4.4 Viewing the license details

The license details for the DCC services and indeed the whole tank gauging installation can be viewed by clicking on the **License** tab.



NXA83\_DCC-Config\_License

Licensing items that are used within the DCC communications sub-system include:

- **Number of Host Drivers:** This is the total number of host ports that can be configured. Please note that 1 host license is always used for the built in Simulator port. Therefore the number of host ports that can be configured via DCC Config is always 1 less than figure displayed here.
- **Number of Slave Drivers:** This is the total number of slave ports that can be configured. Please note that 1 slave license is always used for an internal digital driver. Therefore the number of slave ports that can be configured via DCC Config is always 1 less than figure displayed here.
- **Number of OPC Clients:** This is total number of OPC clients that may be connected at the same time.

## 5 DCC Host Configuration

### 5.1 DCC Host

The DCC Host service is responsible for collecting data from Datacons and gauges, calculating Tank Inventory, and processing command requests from the client.

All tanks configured in the Gauge Configuration screens require at least one COM port to be defined. This is the primary port that DCC Host will poll for data for that tank. For gauges configured for redundant operation then a backup COM port may also be configured.

Each COM port must be defined in the DCC Host sub-system, and may be a direct connection via an RS232 (or RS485 etc.) port on the PC or be a remote connection using a modem.

Network and OPC connections are also supported and are treated in a similar way to physical COM ports.

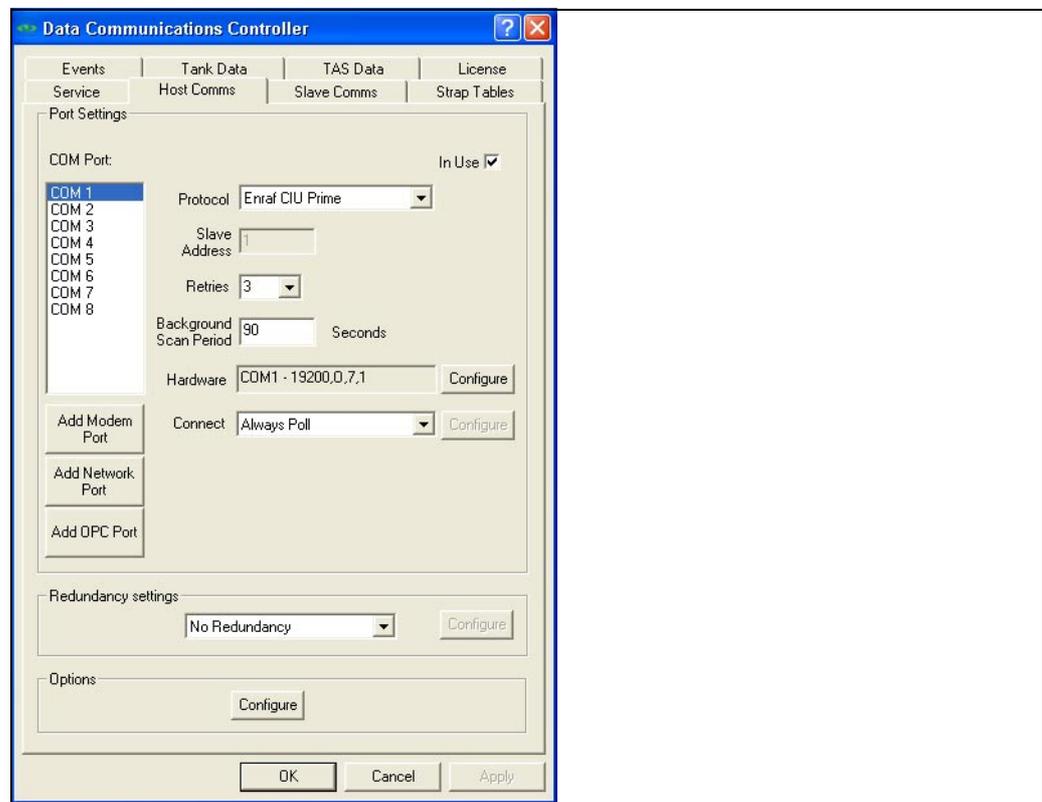
The DCC Host service is capable of polling on multiple COM ports simultaneously, but before it can do so they must be configured within the service.

### 5.2 Configuring Host COM Ports

The DCC Host service requires the selection and configuration of the communication ports that are to be used. These may be RS232/422/485 serial ports or may be modems used to dial up remote devices.

Multiple serial ports and modems may be configured, up to the maximum licensed.

To configure the ports, run the DCC Configuration item, and then click on the **Host Comms** tab.



NXAB3\_DCC-Config\_Host-Comms

A list of all physical COM ports (RS232/485/422 etc.) will be displayed together with any logical Modem, Network and OPC ports configured (they will have an offset of 50 for modems, 70 for network and 110 for OPC connections i.e. COM51, 71 etc.). See the later sections on Modem Ports and Network connections for details on how to add and configure these ports.

Select the COM port to be configured and make sure the **In Use** box is ticked. The protocol and communications parameters can now be edited.

The **Protocol** field determines what is to be connected to the COM port. The current selection options are:

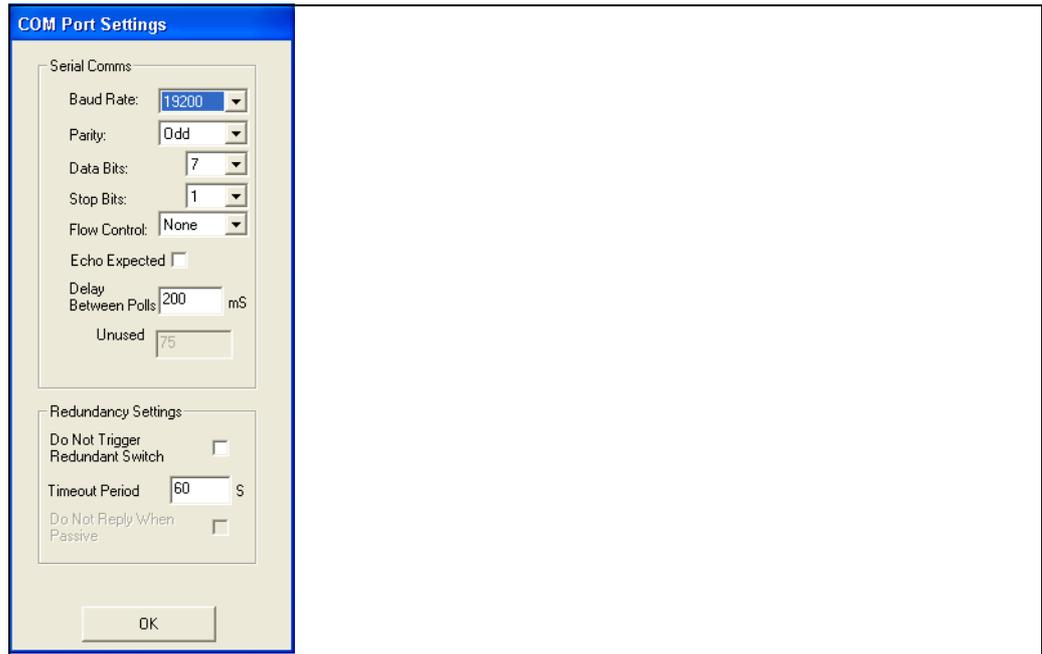
- Datacon
- Enraf CIU
- Enraf CIU Prime
- Enraf BBPM
- Whessoe WM550
- Whessoe WM500
- Whessoe ASCII
- ModbusTG (Generic Modbus Tank Gauges)
- LNG Modbus (SI 6280 and 6290 gauges etc.)
- Hart
- GPE
- L & J
- Hectronic
- OPC TG (Generic gauges from an OPC server)
- Varec Mark/Space
- TankMaster Serf
- Tokyo Keiso FW9000

The **Slave Address** only applies to Datacon devices and should be set to the address of that device.

The **Retries** determines the number of consecutive failed polls before a communications failed alarm is generated.

The **Background Scan Period** determines the update times for certain gauge parameters, including: Density, Observed Temperature, Pressures, Vapor Temperature, Multi-Element Temperatures and Water Level.

The communication parameters will be displayed in the **Hardware** section. For a serial COM port the baud rate, data bits, parity and stop bits will be displayed. These should be set to the same settings of that of the Datacon, CIU or gauges being polled. They may be altered by clicking on the **Configure** button to the right, and editing the parameters in the dialog screen.



NXA83\_DCC-Config\_Host-Comms\_COM-Port-Settings

If the serial communications hardware produces an echo of sent characters (i.e. some RS485 or 20 mA current loop devices) then tick the **Echo Expected** field to tell the driver to handle the echoed characters.

The **Delay Between Polls** field allows the minimum time for each poll to be set. For instance if this is set to 250 ms and it takes 100 ms to carry out a poll, (i.e. send a request and receive a reply), then DCC will pause for the remainder of the time (150 ms) before carrying out the next poll.

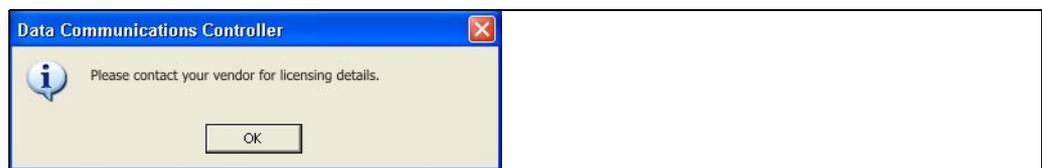
Modem ports will display the telephone number in the Hardware section. The telephone number and modem may be configured via the Configure button. See the later section on Modem Ports for more details (→ 21).

Network ports will display in the Hardware section the slave computer name (or IP address) and the port it will use to communicate. These may be configured via the **Configure** button.

The **Connect** section determines when to connect to the COM port or Modem and poll the gauges; usually this will be **Always** for direct serial ports and Scheduled for modem ports. The **Always** option will set the interface to always try to connect and continue polling for gauge data. See the later section on Scheduling Connections for more details.

Select the **Redundancy settings** option to that required; usually this will be left as **No Redundancy**. See section on Host Redundancy for more details (→ 31).

A warning message may be displayed if a new interface is added by selecting the **In Use** box and the number of licensed host interfaces has been exceeded.



NXA83\_Data-Communications-Controller\_Host-Comms\_interfaces-exceeded

When all host interfaces have been configured or modified, click on the **Apply** or **OK** buttons to save any changes. There may be a momentary delay whilst the details are saved and the DCC Host service automatically re-configures itself with the new configuration.

### 5.3 Configuring Host Modem Ports

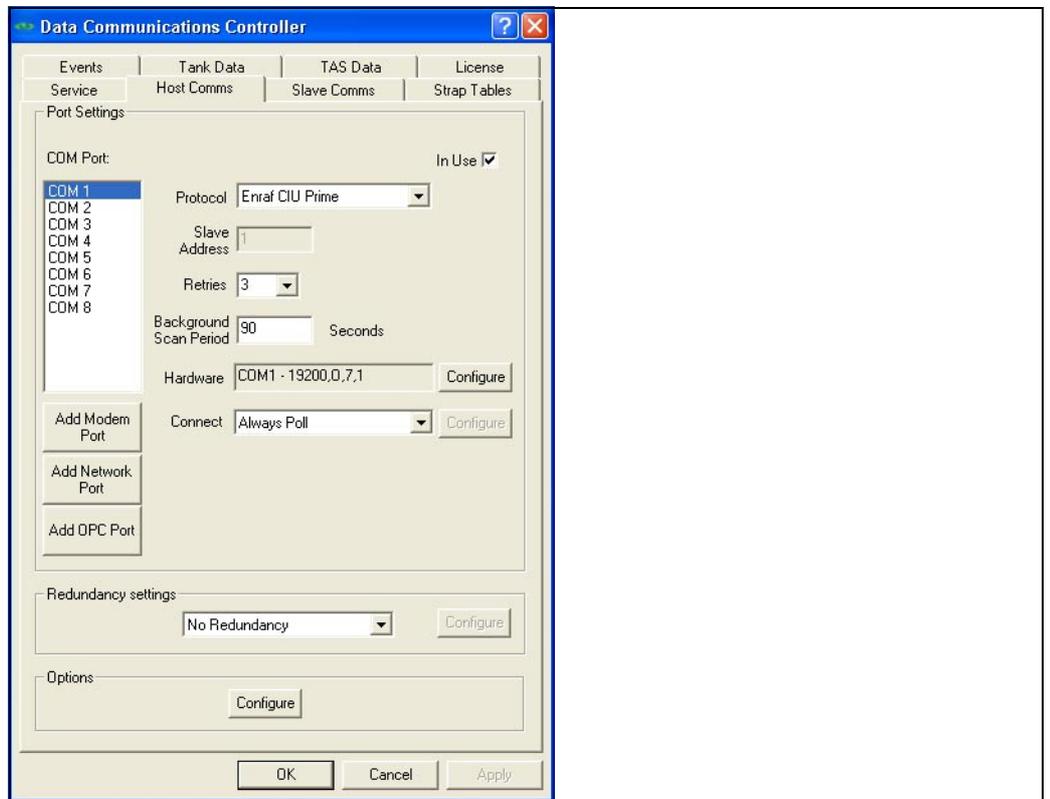
The DCC Host service can have interfaces configured to poll gauges on remote sites using dial-up Modems. These are added to the Host as software COM ports denoted by an offset of 50, thus the first modem COM port will be COM 51.

They are treated as if they were physical COM ports on the PC but the difference being that multiple software COM ports may be the same physical modem device. The reason for this is that each software COM port is configured with a single telephone number and so if multiple remote sites need to be polled then a number of software COM ports can be configured to share the same local modem. However multiple modems on the PC are also supported.

-  Software COM ports are the only way of using modem communications in DCC Host.
  - If a PC has an internal modem installed, it may appear as a COM port on the system, i.e. COM 3 – this should NOT be used in DCC Host!
  - If an external modem is used, do NOT use the actual physical COM port it is connected to. Always add the modem to the operating system then add a new software COM port to DCC Host.

Software COM ports are treated the same as real COM ports with regards to licensing.

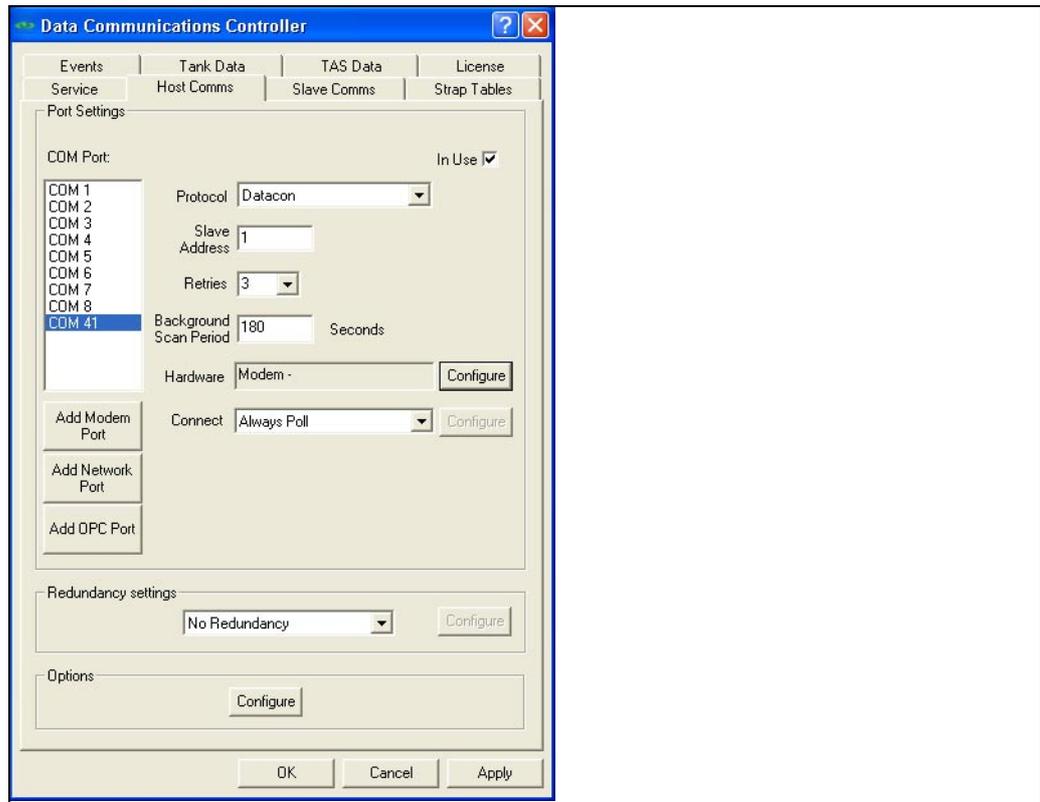
To configure the Modem ports, run the DCC Configuration tool, and click on the **Host Comms** tab.



NXAB3\_DCC-Config\_Host-Comms

Any existing software COM ports will appear in the COM port list as COM 41, 42 etc.

To add a new software COM port, click on the **Add Modem Port** button. A new COM port will be added, starting at COM 41.

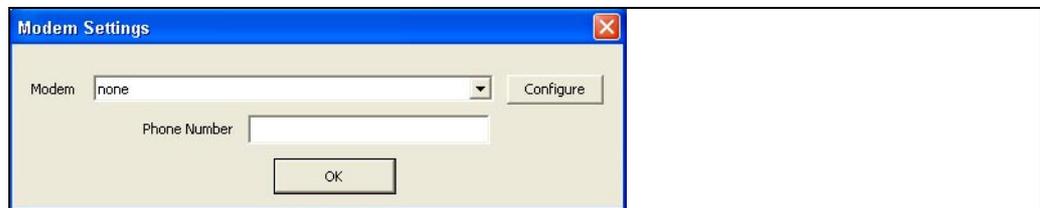


NXA83\_DCC-Config\_Host-Comms\_Add-Modem-Port

The **Protocol** type, **Slave Address**, **Retries**, **Background Scan Period** and **Redundancy settings** fields all function in the same way as for a normal COM port and have been described in a previous section.

The **Connect** field should normally be set to scheduled for a modem interface, see the section on Connections for further details.

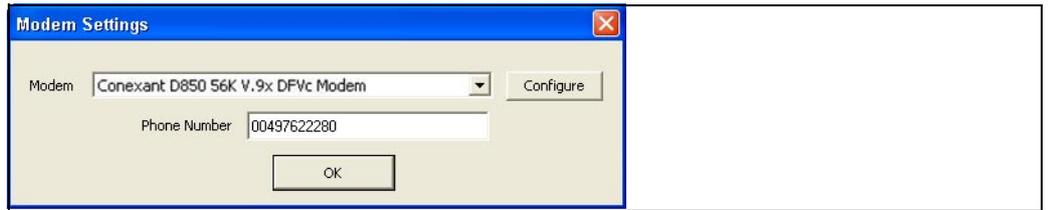
The Hardware section requires configuring with the modem device on the PC and the telephone number of the remote site/system. To do this click on the **Configure** button to the right of the Hardware field.



NXA83\_DCC-Config\_Host-Comms\_Modem-Settings

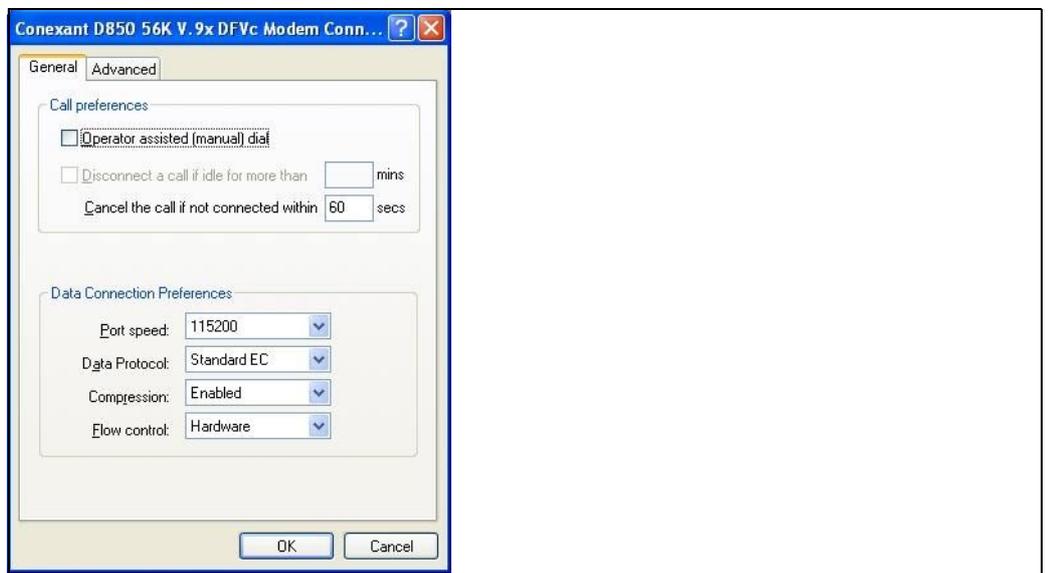
Select the **Modem** from the drop down list of all modems configured on the PC. These may be internal modem boards or external modems connected to a physical COM port. To appear in the list modems should be added to the operating system via the **Phone And Modem** options item in the Control Panel.

Enter the **Phone Number** of the remote site or device. This should include the area code and any external line code if using a local switchboard (a direct outside line is not required).



NXA83\_DCC-Config\_Host-Comms\_Modem-Settings\_Phone-Number

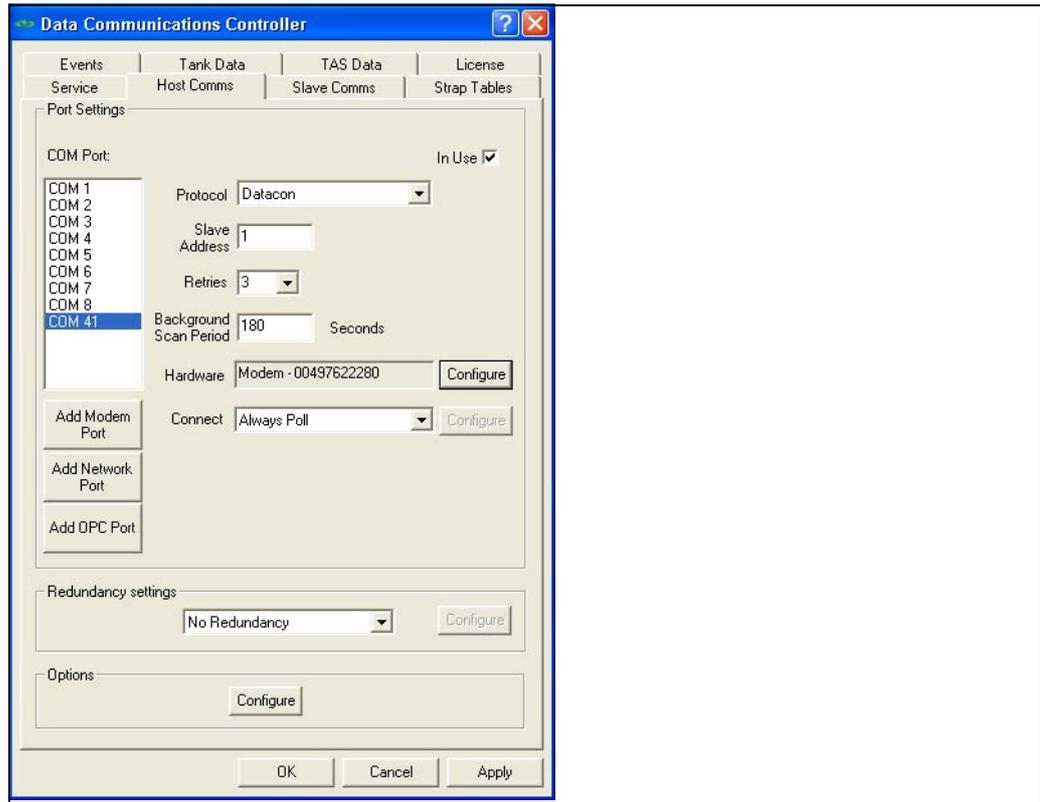
The modem settings can be configured via the standard control panel item **Phone And Modem** options and it is generally recommended that these settings should be used for all interfaces using that modem. However certain settings can be overridden for the individual interface by clicking on the **Configure** button, which will display the standard windows screens for configuring modems. These may be modified and saved for this interface and are then used when it accesses the modem.



NXA83\_DCC-Config\_Host-Comms\_Modem-Configuration

Refer to the windows help for details of the modem settings. Click **OK** the save any settings.

After the modem has been selected and the phone number entered click **OK** to save the new details. The phone number will now be displayed in the **Hardware** field.



NXA83\_DCC-Config\_Host-Comms\_Modem-Port

## 5.4 Configuring Host Network Ports

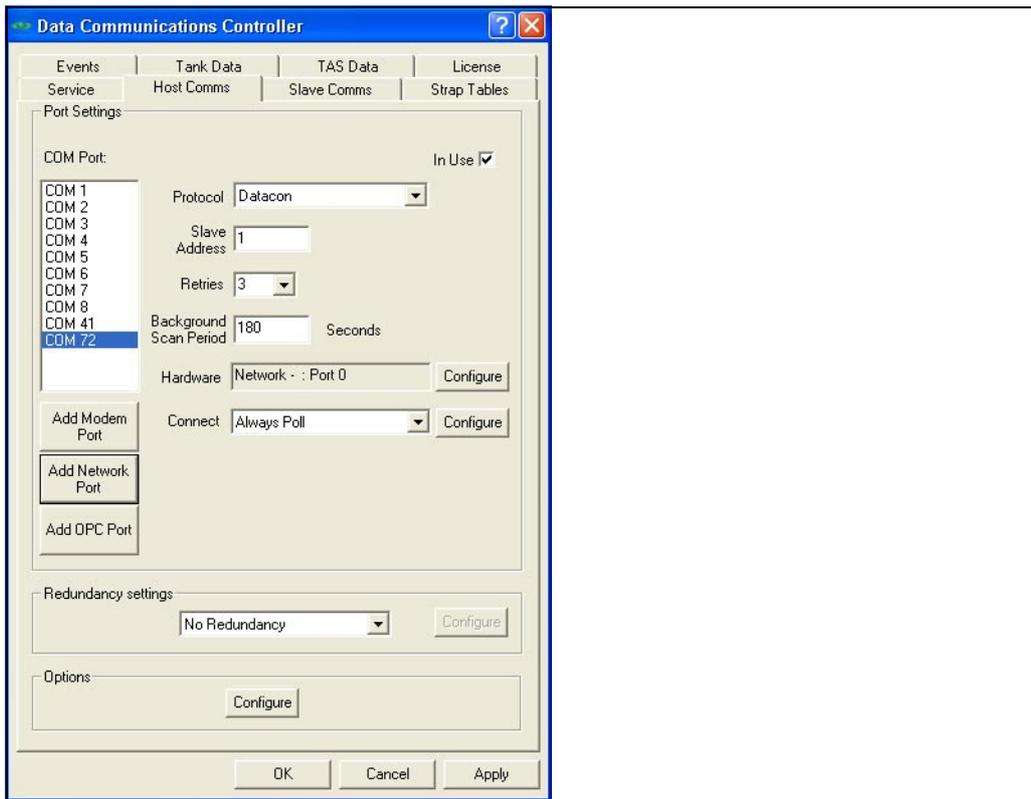
The DCC Host service can have interfaces configured to poll gauges via a standard network connection. These are added to the Host as software COM ports denoted by an offset of 70, thus the first network COM port will be COM 71.

They are treated as if they were physical COM ports on the PC but the difference being that multiple software COM ports may use the same physical network adapter. There is no support for multiple network adapters on the PC except when using network sharing software.

Software COM ports are treated the same as real COM ports with regards to licensing.

A network port makes no difference to the protocol selected to run on that port, except for the Modbus protocol. If that protocol is selected then it will use the Modbus variant called ModbusTCP.

To configure the Network ports, run the DCC Configuration tool and click on the **Host Comms** tab.



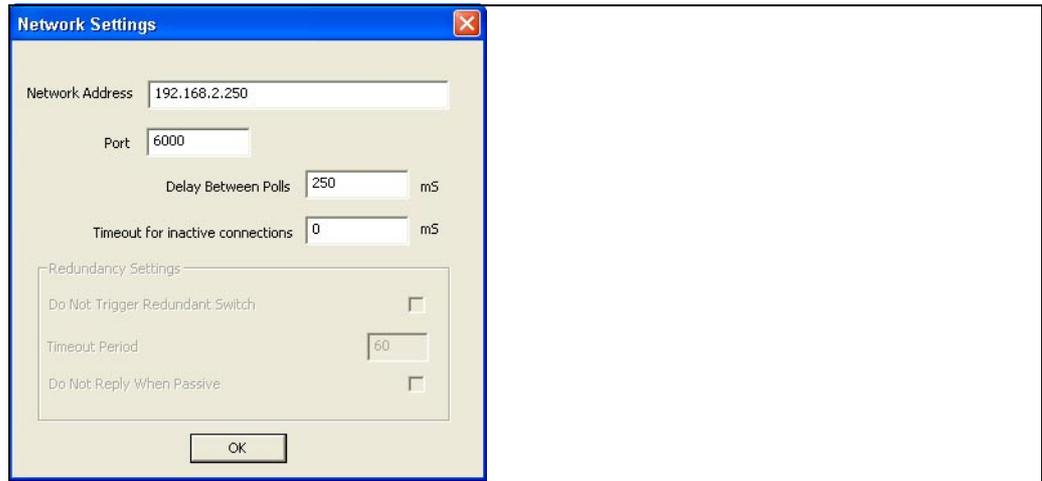
Any existing software COM ports will appear in the COM port list as COM 71, 72 etc.

To add a new software COM port, click on the **Add Network Port** button. A new COM port will be added, starting at COM 71.

The **Protocol** type, **Slave Address**, **Retries**, **Background Scan Period** and **Redundancy** Settings fields all function in the same way as for a normal COM port and have been described in a previous section.

The **Connect** field may be set as required, as the network port will support scheduled and permanent connections. See the section on Connections for further details (→ 49).

The **Hardware** section requires configuring with the network name or IP address of the remote network device and the IP port that it will use to communicate. To do this click on the **Configure** button to the right of the **Hardware** field.



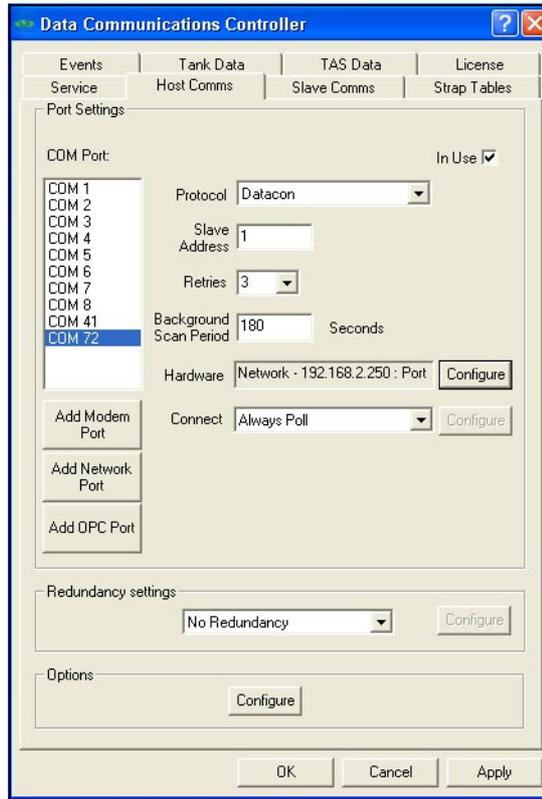
Enter the **Network Address** of the remote network device (i.e. PC) to connect to. This may be entered as the name of the device (i.e. PC name) or as its TCP/IP address, for example 192.168.1.250.

If the name of the remote device is to be used, ensure that it is known to the system that DCCHost is running. This is usually the case if both systems use the same DNS server.

The **Port** field defines the IP port that the remote system is listening on.

Communications via the network is usually extremely fast, in the order of a few milliseconds to poll for tank data. For various reasons this may be a problem (network traffic too high, excessive CPU time used on this process etc.) and can be slowed down by entering a **Delay Between Polls** time. It is not recommended that this figure is reduced below 100 ms.

Click **OK** to save the network settings. The Network details will be displayed in the **Hardware** section.



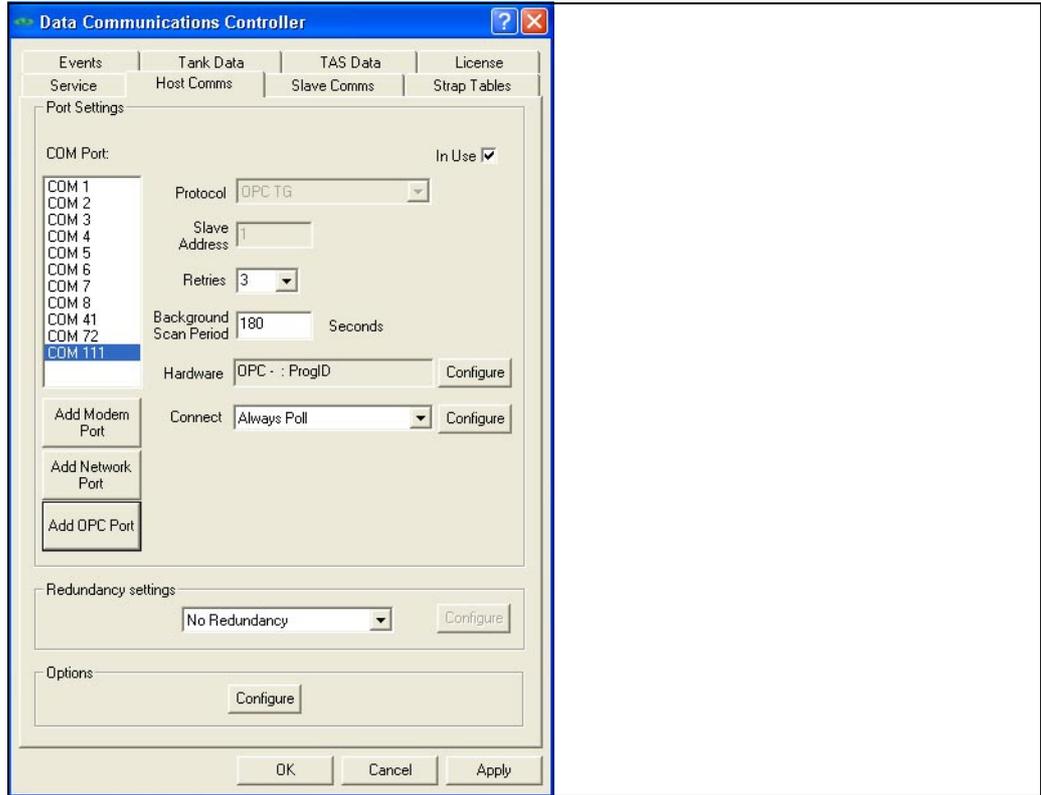
NXA83\_DCC-Config\_Host-Comms\_Hardware

## 5.5 Configuring OPC Network Ports

The DCC Host service can have interfaces configured to poll gauges from an OPC Data Access (DA) server via standard network connection. These are added to the Host as software COM ports denoted by an offset of 110, thus the first network COM port will be COM 111.

They are treated as if they were physical COM ports on the PC but the difference being that multiple software COM ports may use the same physical network adapter. Software COM ports are treated the same as real COM ports with regards to licensing.

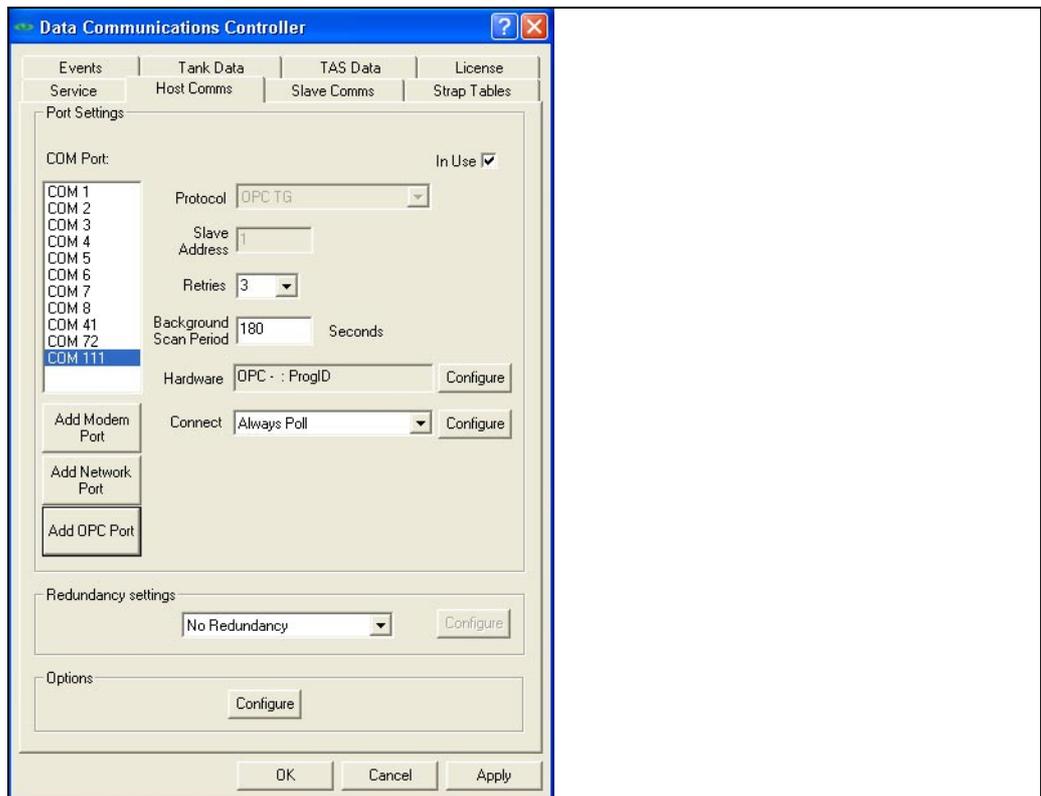
To configure the OPC ports, run the DCC Configuration tool and click on the **Host Comms** tab.



NXA83\_DCC-Config\_Host-Comms\_OPC-Network-Port

Any existing OPC COM ports will appear in the COM port list as COM 111, 112 etc.

To add a new OPC COM port, click on the **Add OPC Port** button. A new COM port will be added, starting at COM 111.



NXA83\_DCC-Config\_Host-Comms\_OPC-Network-Port

The **Protocol** type will always be set to “OPC TG”. The **Retries**, **Background Scan Period** and **Redundancy settings** fields all function in the same way as for a normal COM port and have been described in a previous section.

The **Connect** field may be set as required, as the OPC port will support scheduled and permanent connections. See the section on Connections for further details (→ 49).

The **Hardware** section requires configuring for the OPC server settings. For more information on these settings and for full details on configuring an OPC TG interface, refer to the separate manual “OPC Tank Data Server” (BA01289G).

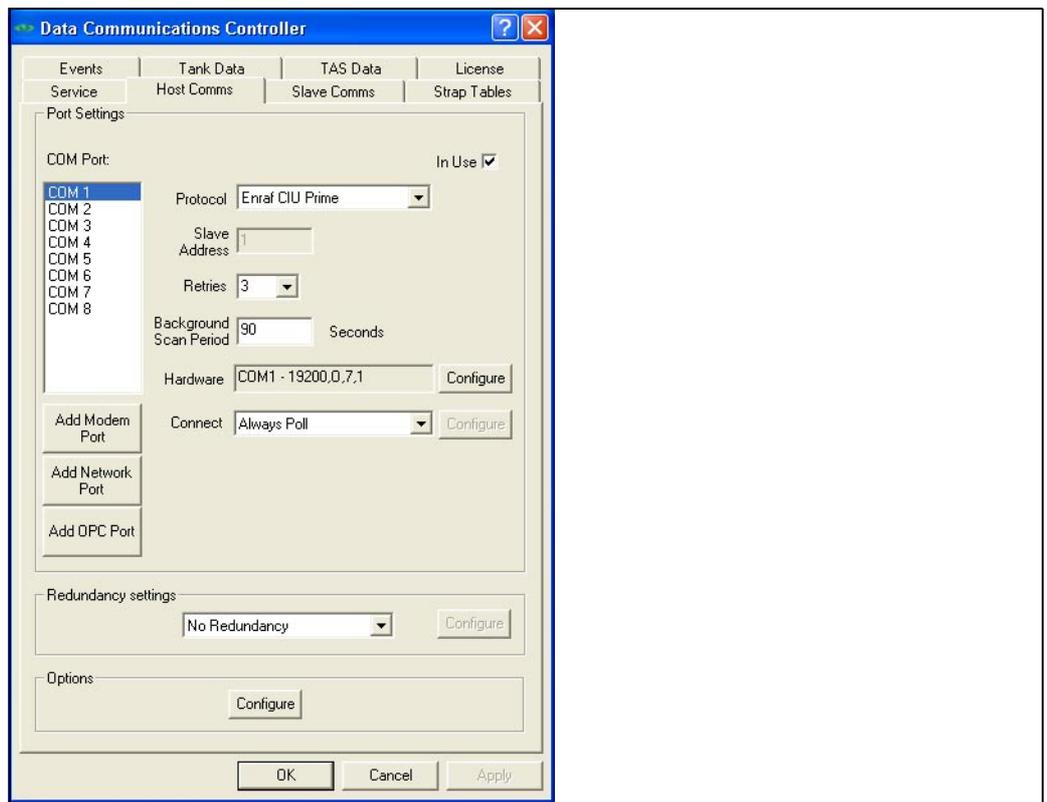
## 5.6 Configuring Scheduled Connections

Interfaces configured in DCC Host will normally be set to being **Always** connected to the gauges. This means that as soon as the interface is started it will connect to the serial COM port, dial out on a modem or connect to the remote network device, and then continue polling the gauges indefinitely. If it becomes disconnected from the gauges (i.e. a modem loses the line or the network device is disconnected) it will keep trying to re-connect and poll the gauges. This type of connection should be configured on interfaces using local COM ports to communicate to gauges.

Interfaces may also be set to **Scheduled** connection, where the interface will connect and poll the gauges only at user defined times. Once connected it will poll the gauges for all primary and background data and then disconnect i.e. hang up the modem connection. This type of connection is usually configured where an interface uses a modem to dial up remote sites.

Network ports may be configured as either type.

To configure the connection type of an interface, run the DCC Configuration tool then click on the **Host Comms** tab and select the port to be configured.

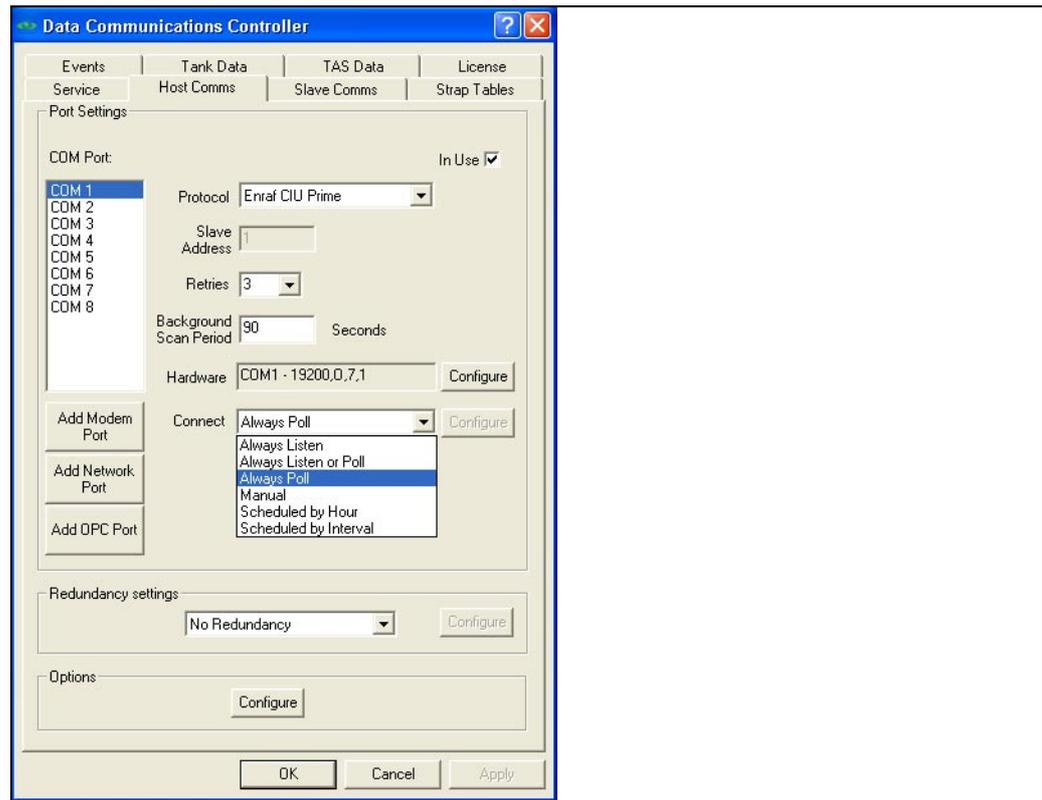


NXA83\_DCC-Config\_Host-Comms

The Connect field will normally default to **Always Poll** (or **Always Listen** in the **Slave Comms** tab).

To alter this, select from the list of options from the drop down list for the field. These include:

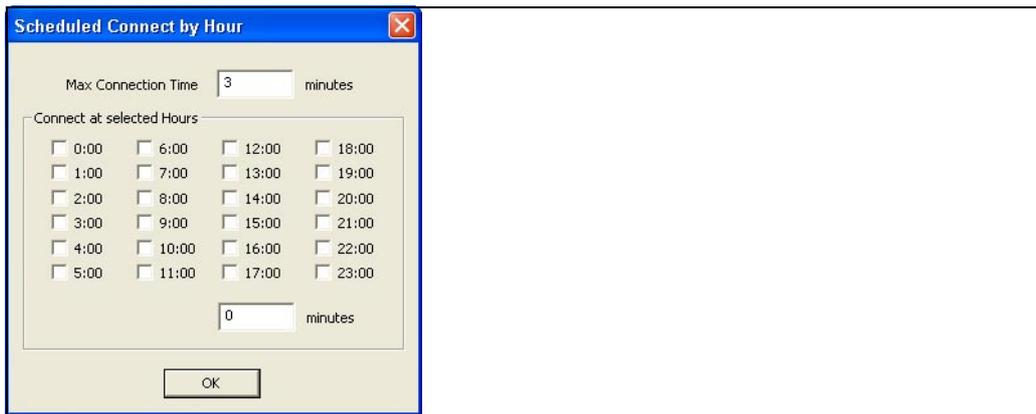
- **Always Poll** – Interface will always try to be connected and polling the gauges
- **Manual** – Interface will only connect and poll gauges if manually commanded by user
- **Scheduled by Hour** – Interface will connect on the hours selected by the user
- **Scheduled by Interval** – Interface will connect periodically
- **Always Listen** – mode for slave interfaces
- **Always Listen or Poll** – not supported at the moment



NXA83\_DCC-Config\_Host-Comms\_Connect-Field

If either of the **Scheduled** options has been selected then connection times must be configured. To do this click on the **Configure** button which will now be displayed to the right of the **Connect** field.

If the **Scheduled by Hour** option was selected then the hours on which the interface will connect may be defined in the dialog box displayed:



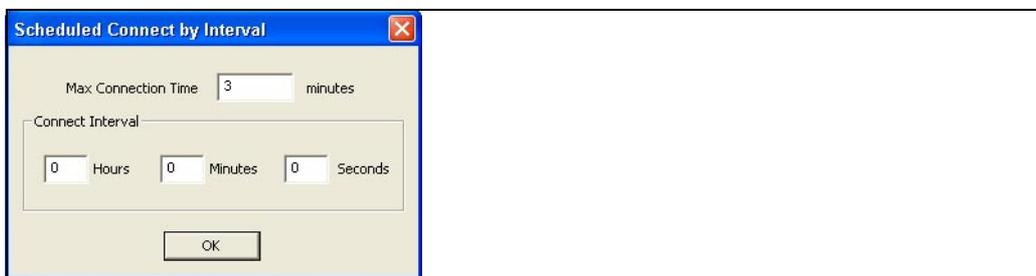
NXA83\_DCC-Config\_Host-Comms\_Scheduled-Connect

The **Max Connection Time** field defines the maximum time in minutes that the interface will remain connected and polling the gauges before it will disconnect. Normally the interface will disconnect automatically after it has retrieved all data from then gauges, so this field should be set longer than the time it takes to update all data. Its main use is as a limit to the time a modem will remain connected, if problems occur polling the gauges.

The **minutes** field in the **Connect at selected Hours** section defines the minutes past the hour that the connection will take place.

If there are problems with scheduled connections where some data is not being retrieved then it may be that the **Max Connection Time** is too short and should be set longer.

If the **Scheduled by Interval** option was selected then the interval between connections must defined in the dialog box displayed:



NXA83\_DCC-Config\_Host-Comms\_Scheduled-Connect-Interval

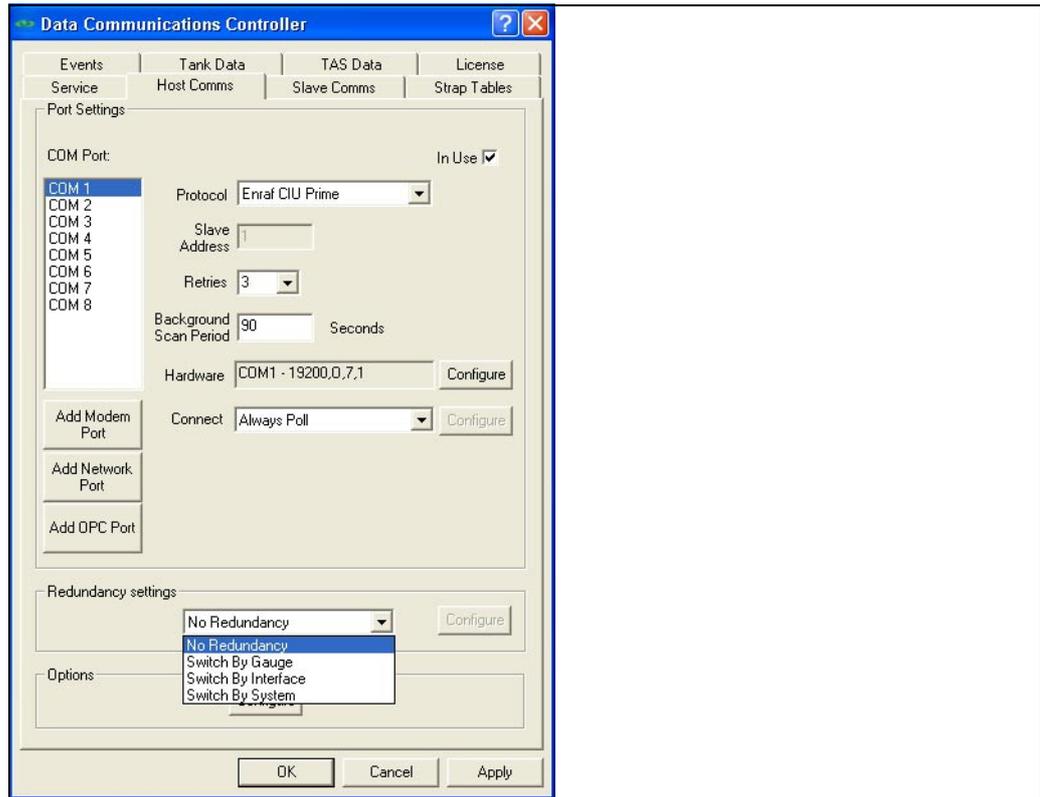
The **Max Connection Time** is the same as for the hourly connection display.

## 5.7 Redundancy

The DCC Host service supports redundant communications to gauges using a pair of communication ports (Physical COM ports, modems or Network connections).

To use redundant communications gauges must be configured with a primary and a backup port. Only one port at a time will be used to poll for data, initially this will be the primary port. If this port fails then the gauge will be polled on the backup port until that fails, when it will switch back to the primary and so forth. If both ports have failed then it will keep switching between the two ports until one responds.

To configure redundant communications, run the DCC Config tool and click on the **Host Comms** tab.



NXA83\_DCC-Config\_Host-Comms\_Redundancy

Any ports that are to be used as part of a redundant communications system should then have their **Redundancy settings** field set to either of the following:

- **Switch By Interface** – where gauges will only switch to the Backup COM port if ALL gauges have failed on that interface
- **Switch By Gauge** – where gauges will switch individually

The modes defined above apply only to ports running on the same system, and have no connection with other systems.

- **Switch By System** – this mode is used to control two identical systems where gauges will be polled on only one live system. The other non-live system will retrieve its data from the live system. This mode is normally used for embedded systems and is more fully explained in a separate manual on redundancy.

If the **Redundancy setting** is configured as **No Redundancy** then a gauge will NOT switch between interfaces regardless of whether it has been configured with a backup port.

After the interfaces have been configured then gauges may be configured with redundant operation. To do this, select the gauge in the **Gauge Configuration** screen, then set the **Comm Port** field to the desired primary interface port and the **Backup Comm port** field to the backup interface port.

NXA83\_Device-Configuration\_Port-Address\_Comm-Port

1 Gauge configuration screen

## 5.8 DCC Options

A set of options that alter the way DCCHost operates may be configured. These are global settings that apply to all interfaces in DCCHost.

Click on the **Configure** button in the **Options** section. Current supported options include:

- **Manual Levels set Programmable Alarms** – by default a manual value will not trigger a programmable alarm, ticking this option will reverse this functionality.
- **Servo/Stow set Programmable Alarms** - by default when a gauge is moving due to a stow/unstow or servo command then programmable level alarms will be suppressed, ticking this option will reverse this functionality so that the current position of the sensing head will be used to trigger alarms.
- **Log Profiles to CSV file**
- **Display Error Messages**
- **Auto control the manual mode for water level** – ticking this option will set DCC to automatically set the water field into manual mode if the value from the field is invalid.
- **Set Flow to Zero if below threshold** – by default all flow data will be saved regardless of what the value is. This may lead to a flow being displayed that is due to the "ripple" of the product. Ticking this option will set DCC to clamp all flow rates to zero if the value is below the flow rate threshold value entered in the System Settings.
- **Suppress Alarms when Displacer Above Liquid Surface** – ticking this option will set DCC to ignore any level alarms returned from the gauge, but only when the displacer is above the product surface i.e. not following level. For instance when servoing or stowing etc.
- **Suppress Alarms when Displacer Below Liquid Surface** – ticking this option will set DCC to ignore any level alarms returned from the gauge, but only when the displacer is below the product surface i.e. not following level. For instance when profiling or finding water.

## 5.9 Modbus TG protocol configuration

Modbus TG gauges are assigned to Modbus TG interface ports and are used to poll for tank data from slave devices using the standard Modbus protocol. These slaves may be PLCs, Remote I/O units or gauges equipped with a Modbus interface.

Modbus TG gauges are often used as Auxiliary gauges, to bring in background data items for primary gauges.

Modbus TG gauges may require more configuration than other gauges due to their generic nature.

There are two steps to configuring a gauge:

- Step 1: Configuring gauges on the port with a basic Modbus map, using the standard Gauge Configuration screens
- Step 2: If required, configuring an advanced Modbus register map in an XML config file

### 5.9.1 Modbus TG Basic Map

Gauges must be configured to use the Modbus TG driver. This can be done in the normal way using the standard **Gauge Configuration** screens.

A very basic Modbus map may be defined for the gauge using the following fields:

- **Device Type** – the gauge must be set to "Modbus TG"
  - **Device Address** – the Modbus RTU address of the slave device
  - **Modbus Address** - the Modbus address of the register that returns Product Level. This starts from zero and should not include the Modbus function code offset. For example, a Modbus register full address of 40001 should be entered as 0 (and Modbus Type should be set to 3).
- NOTE** - not used for Auxiliary duty gauges!
- **Data Type** – set to "S"
  - **Modbus Function** – the Modbus function code to be used for the Level register (and temperature if used) i.e. 3 or 4
  - **Auxiliary 1** – an optional field that defines the Modbus address of the register that returns the temperature

This will set the Modbus TG to poll for the level data (for non-Auxiliary duty gauges only) using a standard set of decoding parameters:

- Input range 0 to 32 000 in millimeters
- Encoded as a two byte Short with Little Endian format
- A value outside the range 0 to 32 000 will be decoded as a bad value.

If the **Auxiliary 1** field has been entered then the Modbus TG will poll for temperature data using a standard set of decoding parameters:

- Input range  $\pm 3200$  °C sent times 10, i.e. +32 °C encoded as 320
- Encoded as a two byte Short with Little Endian format
- A value of 8000 Hex will be decoded as an illegal value.



If the above encoded format for level (and temperature if enabled) is supported by the slave device and no other data (e.g. density etc.) is required, then no other configuration is needed for the ModbusTG driver.



Level will never be polled for Auxiliary duty gauges.

## 5.9.2 Modbus TG Advanced Map

If the very basic Modbus map defined when the gauges are added to the system is not enough then it may be overridden with a much more advanced map, defined in a separate XML format file. This allows more flexible encoding of data to be defined and also permits a much greater range of data parameters to be polled for.

A configuration tool, "Modbus Config.exe", is available to allow the user to configure the XML file. This is located in the standard installation directory.

The user may also directly edit the configuration file and details of how to do this are included in this document. However it is recommended that direct editing should only be used in extreme circumstances, as it may lead to an invalid file structure.

The advanced map consists of a set of Interfaces, each of which may have:

- A set of default parameter details which apply to ALL gauges on the interface
- A set of gauge parameter details which apply only to the selected gauge

All parameters are optional and need not be defined in the map, as they will default to a set of standard values (defined later in this document).

### Modbus Configuration Tool

To configure the advanced Modbus map, select the menu option

**Configuration → DCC Host Configuration.**

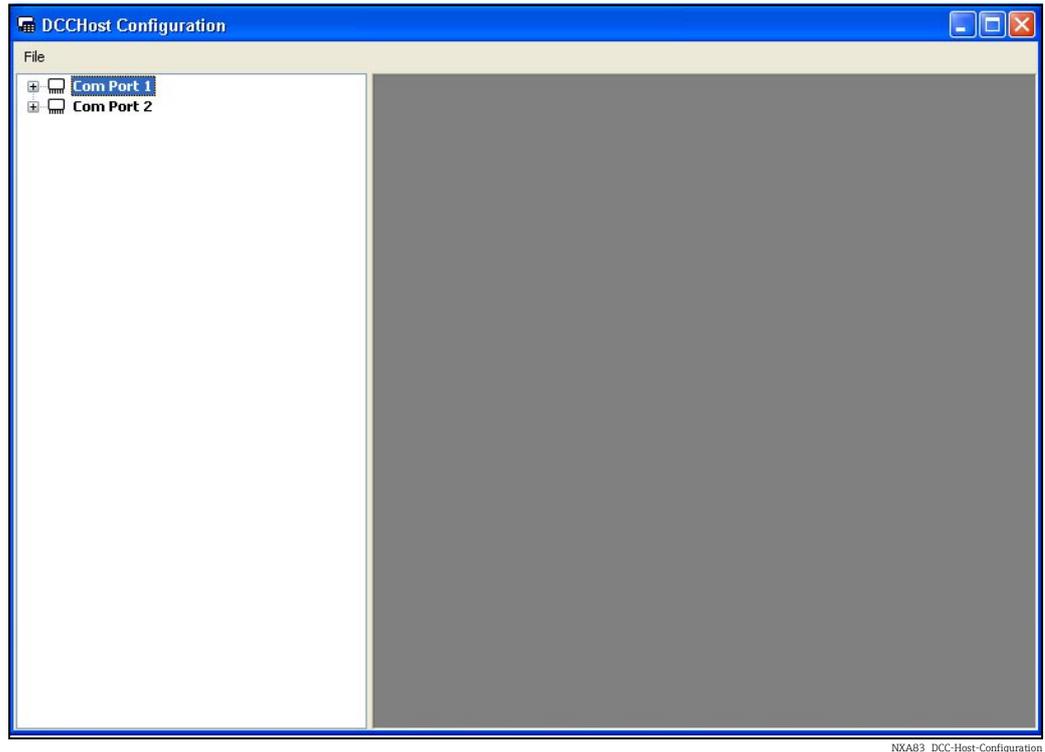
The DCC Host Configuration page allows advanced configuration of certain host drivers, including the Modbus TG driver.

A list of all COM ports that have drivers that require advanced configuration will be shown in the left hand panel. This will include all Modbus TG interfaces that have at least one Modbus TG gauge assigned to them.



If no Modbus TG interfaces appear in the list, then ensure that all Modbus TG interfaces have been defined and at least one gauge of type "Modbus TG" has been assigned to each interface.

The list of COM ports can be expanded to display all gauges assigned to that port.



Each COM port will have a list of all tanks assigned (including the gauge duty, i.e. Primary, Secondary etc.) and also a section called **Defaults**.

The Modbus fields that are to be polled for each gauge will be displayed beneath the gauge they apply to. These fields include product level, temperature, pressure etc. and will only be polled by the Modbus TG interface if they are added to the configuration in this tool.

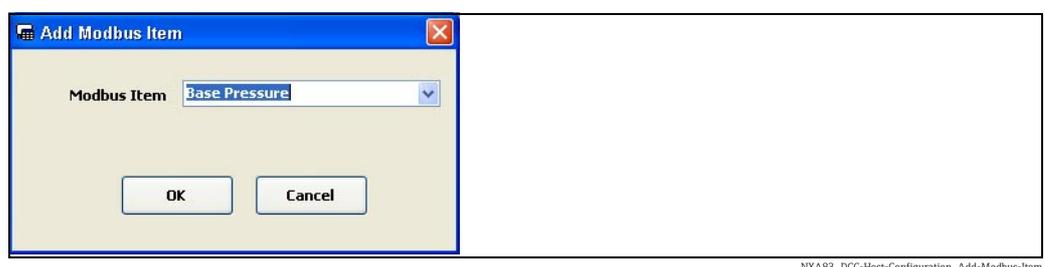
 For non Auxiliary duty gauges, Product Level will always be polled for, even if not configured here – see section on Modbus TG Basic Map (→  34).

The **Defaults** section allows the user to configure Modbus field definitions that will be added to ALL gauges on this Modbus TG interface. If a field is added to the **Defaults** section it will be polled by a gauge even if the field has not been added to the individual gauge.

Each gauge can be configured with Modbus field definitions that apply only to that gauge. All field data entered for an individual gauge will override data entered in the **Defaults** section. Thus data common to all gauges may be entered in the **Defaults** section, then individually tailored for each gauge as required.

### Adding a Modbus Field

To add a new Modbus field to a gauge or the Defaults, then right click on the field and select **Add** from the pop-up menu. Then select the field from the **Add Modbus Item** dialog displayed.



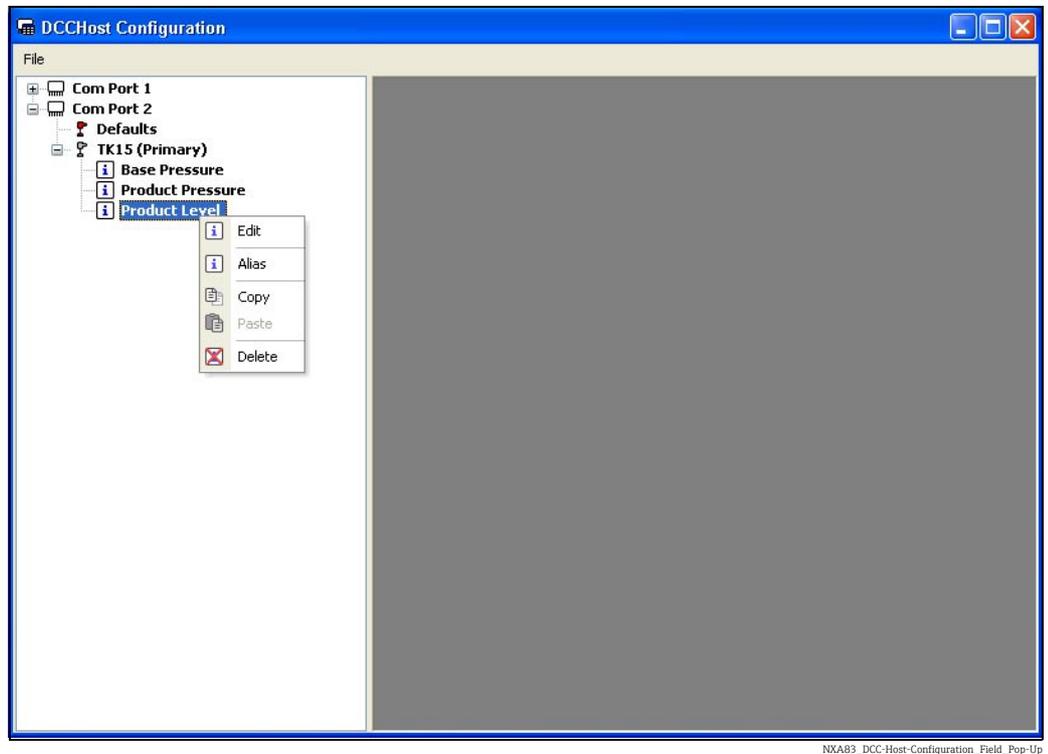
The new field will be displayed in the left hand panel.

### Deleting a Modbus Field

To delete a Modbus field, right click on the field and select **Delete** from the pop-up menu.

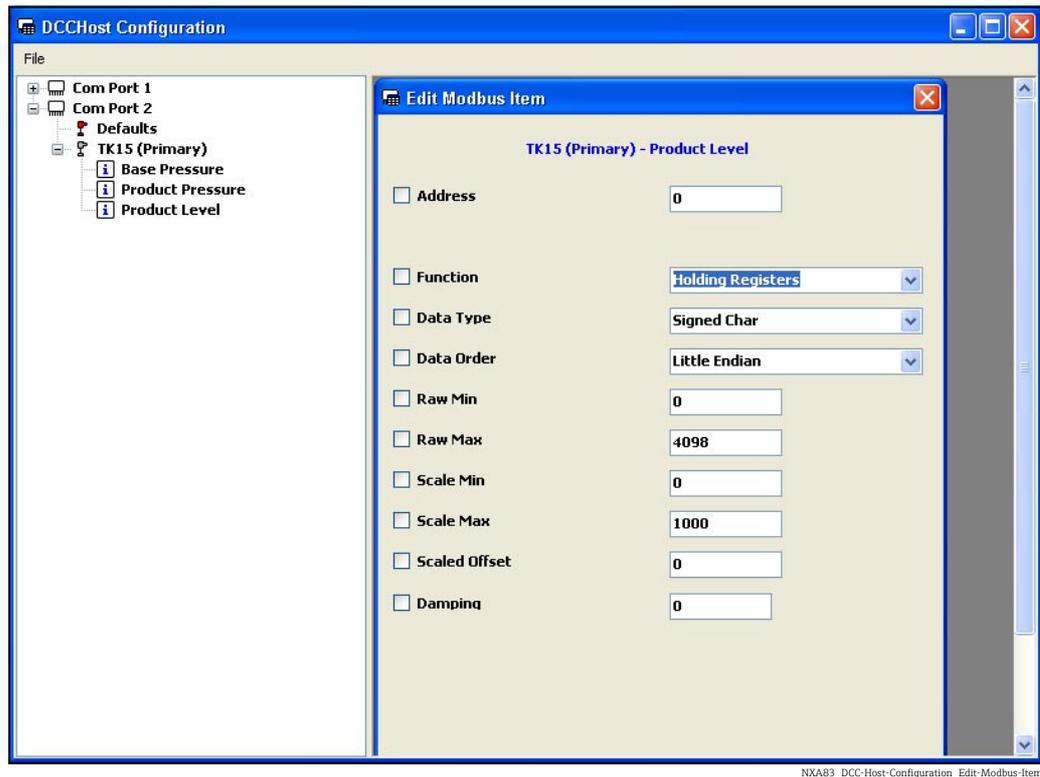
### Editing a Modbus Field

To edit a Modbus field, right click on the field and select **Edit** from the pop-up menu.



NXA83\_DCC-Host-Configuration\_Field\_Pop-Up

This will display the current set of definitions for the field in a window in the right hand panel. Multiple Modbus fields may be edited at the same time.



The parameters are optional and are only used if the left hand box is ticked. Any parameters not ticked will use the default values, defined later in this section.

The range of parameters displayed will depend on the data field selected.

Common to most Modbus fields are the following parameters:

- **Address** – the Modbus address of the register or coil, from zero
- **Function** – the Modbus function type i.e. Holding Register, Coil etc.
- **Data Type** – the format of the data to be requested. May be one of the following:
  - Signed Char --> (range -127 to +127 using 1 register)
  - Unsigned Char --> (range 0 to 255 using 1 register)
  - Short --> (range -32767 to +32767 using 1 register)
  - Unsigned Short --> (range 0 to 65535 using 1 register)
  - Integer --> (range -2147483647 to 2147483647 using 2 registers)
  - Unsigned Integer --> (range 0 to 4294967295 using 2 registers)
  - Float --> (using 2 registers)
  - Double --> (using 4 registers)
- **Data Order** – the byte order of the data, may be one the following:
  - Little Endian --> (Standard format for MS Windows based systems)
  - Big Endian
  - Little Logical
  - Big Logical --> (often used by DCS systems for Float datatypes)
- **Raw Min** and **Raw Max** – the raw data range. Used in scaling data (see below) and also if the value is outside the raw limits then the value will be set to illegal, i.e. DN04
- **Scaled Min** and **Scaled Max** - the scaled data range. The Scaled value must be converted to the default units used within the TankGauging database (Level=mm, Temperature=°C, Density=Kg/l, Pressure=BarA). For status fields it is recommended that these should be set to the same as the raw limits. Data will be converted using the formula:  

$$\text{ScaledValue} = ((\text{RawValue} - \text{RawMin}) : (\text{RawMax} - \text{RawMin})) \cdot (\text{ScaledMax} - \text{ScaledMin}) + \text{ScaledMin}$$
- **Scaled Offset** – an optional offset to be applied to the scaled value, in the database units for the data type (Level=mm, Temperature=°C, Density=Kg/l, Pressure=BarA)

- **Damping** – not used for status values. A percentage value that is used to damp values that fluctuate rapidly. This field should be entered in the range 0 to 100, with 0 and 100 being treated the same as NO damping. Formula used is:  
DampedValue = OldValue + ((NewValue – OldValue) · damping : 100)

If the Modbus field is a Status (i.e. Base Pressure Status, Product level Status etc.) then the following parameters will also be available:

- **Status Type** – only used for status fields. Defines the test used on the input value to determine if the field is valid i.e. Good. If the test is true for illegal or bad then the field is Invalid and will be set to DNO4. May be one of the following:
  - None
  - One illegal value
  - One good value
  - Bits set good
  - Bits set bad
- **Status Test Value** – only used for status fields. Defines the value used to test any special case for validity, depending on the value of statustype. If statustype is 3 or 4 then any of the bits in this value will be tested.

If the Modbus field is an Alarm (i.e. Level High High, Temp High etc.) then the following parameters will also be available:

- **Alarm Type** – defines the test used on the input value to determine if the alarm is active. May be one of the following:
  - None
  - Equals Value Sets Alarm
  - Bits set Alarm (i.e. Any of bits in Alarm Test Value will set the alarm)
  - Bits Cleared set Alarm
  - Greater Than Value Sets Alarm
  - Less Than Value Sets Alarm
- **Alarm Test Value** – the value used to test the alarm activation status of the incoming data
- **Hysteresis** – the value used to determine when alarm is cleared. Only used for types "Greater Than Value Sets Alarm" and "Less Than Value Sets Alarm"

Any Field parameters not defined for an individual gauge will use those defined in the Defaults section.

If the Field parameter has not been defined in the Defaults section then the following defaults will be used by the Modbus TG interface when polling for data:

- slaveaddress = **Device Number** entered in **Gauge Configuration** screen
- address = 0
- function = 4
- dataorder = Short
- rawmin = 0
- rawmax = 32 000
- scaledmin = 0
- scaledmin = 32 000
- statustype = 1 (status fields only)
- statusvalue = 8000 hex (status fields only)
- damping = 0 (i.e. not used) (value fields only)

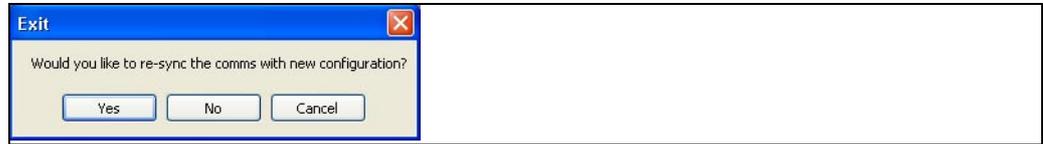
 For Product Level the default values will be those set in the **Gauge Configuration** screens. If **Product Temperature** has been defined in the **Gauge Configuration** screens then those defaults will be used.

 The Modbus PLC address for the field will be the **Device Address** field entered in the **Gauge Configuration** screen.

Click **OK** in the edit box to save the parameters for the Modbus field.

### Saving the Modbus Map

To save the new Modbus configuration to the Modbus xml file, close the DCC Host Configuration tool. If any changes have been made, the following will be displayed.



NXA83\_DCC-Host-Configuration\_Exit\_Re-Sync

Click **Yes** to save the data and automatically force the DCCHost service to reload the new Modbus map for any ModbusTG interfaces.

## 5.9.3 Field Definitions

The Defaults section and any Gauge mode can contain a set of Modbus fields which will be polled by the Modbus TG.

Each field node can contain one or more attributes which define what the field is and how the Modbus TG driver will decode the incoming data when polled.

The Modbus TG interface is specifically designed to poll for tank gauge data and thus data fields are polled in a defined way.

There are three basic types of fields in the ModbusTG driver, which operate in slightly different ways. The fields are:

- Data Fields – used to get the actual value of a data item, such as level, temperature etc.
- Status Field – optional, used to get the validity of an associated data item. A status field is denoted by name ending in 'Status'.
- Alarm Field – used to bring in an alarm. An alarm field is denoted by 'Alarm' in name of field.

Certain fields are polled at high priority, and will always be polled regardless of whether they have been added to the advanced Modbus map or not. These are:

- Product Level
- Product Level Status



The above fields will NEVER be polled for an auxiliary duty gauge.

The following fields will be polled at high priority if they have been defined in the advanced Modbus map:

- Product Temperature
- Product Temperature Status
- DensityHigh Alarm
- DensityLow Alarm
- LevelHigh Alarm
- LevelHighHigh Alar
- LevelLow Alarm
- LevelLowLow Alarm
- TempHigh Alarm
- TempLow Alarm
- GaugeStatus

Other fields are only polled on background scans, and only if they have been defined in the Modbus advanced map. The background scan period is defined per interface and defaults to three minutes. These fields include:

- BasePressure
- BasePressureStatus
- FreeWaterLevel
- FreeWaterLevelStatus
- Density
- DensityStatus
- ObservedTemperature
- ObservedTemperatureStatus
- ProductPressure
- ProductPressureStatus
- VapourPressure
- VapourPressureStatus
- VapourTemperature
- VapourTemperatureStatus
- ElementTemperature 1 to 16
- ElementTemperature 1 to 16 Status

From the above list it can be seen that most fields are in pairs, with a value field and a status field. Thus the Product Level field has a corresponding Product Level Status field etc.

The validity of a value field is determined by the following:

- If the incoming data is outside the raw scaling limits then the value will be set to illegal status i.e. DN04.
- If a status field has been added (i.e. level status added as well as level value) then it determines the status of the value.

A status field may be added to the map in addition to the value field. These are generally used in cases where the status of the data is set in a separate Modbus field (coil or register). However the status field may be set to the same Modbus address as the value.

For status fields there are an extra set of parameters that determine how to decode the status. They work in the following manner:

- If the status is outside the raw scaling limits then the value will be set to illegal status i.e. DN04.
- If the status is within the raw limits then the Status Type parameter will be used to determine how to test for illegal status using the Status Test Value.

Many Modbus maps implemented by slave devices do not have separate status registers (or bits) but instead set the value to some error figure (i.e. full scale etc.). In this case the status field should be set to the same Modbus address and parameters of the value field (i.e. same raw scaling parameters).

Alarm fields must have the Status Type and Status Test Values fields configured. They are used to test the incoming data to check whether the alarm is active or inactive. In this case if the data tests Good then the alarm is active, if it tests Bad then the alarm is inactive.

#### 5.9.4 User Defined Alarms

User Defined Alarms are a set of alarms generated by certain gauge type, the user may configure the text to be displayed when the alarm is activated and cleared using the **Event Server Config** tool (see the Configuration manual, BA01291G).

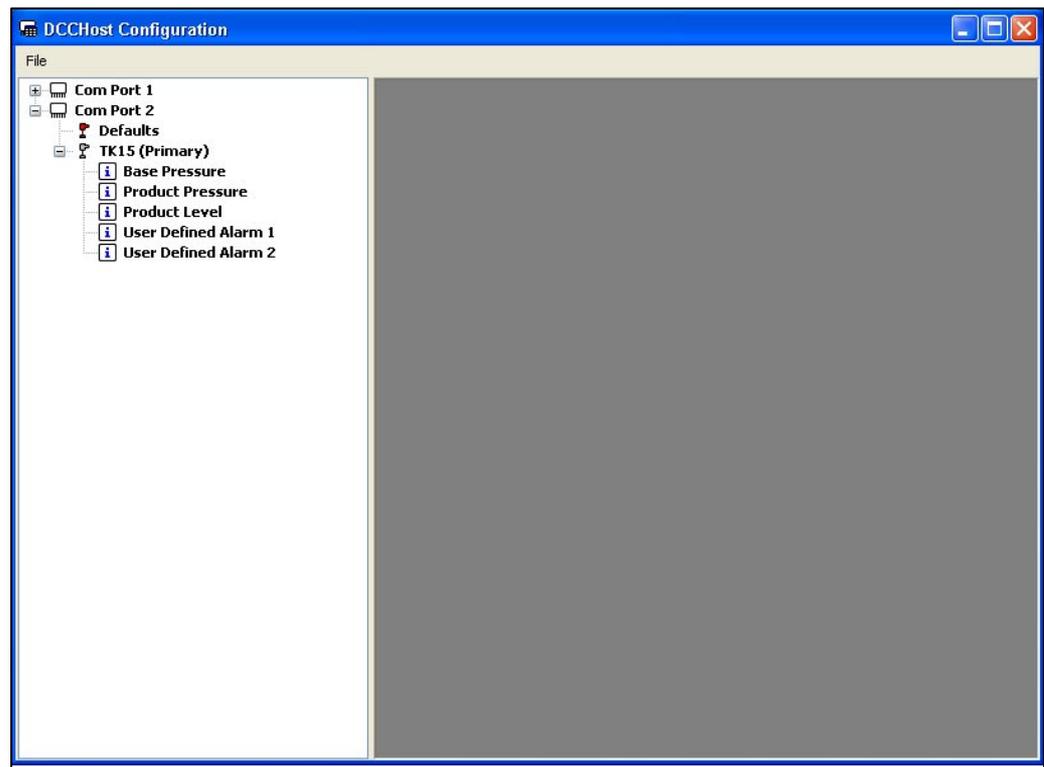
There are 32 such alarms per gauge:

- User Defined Alarm 1
- User Defined Alarm 2
- User Defined Alarm 3
- etc. ... up to

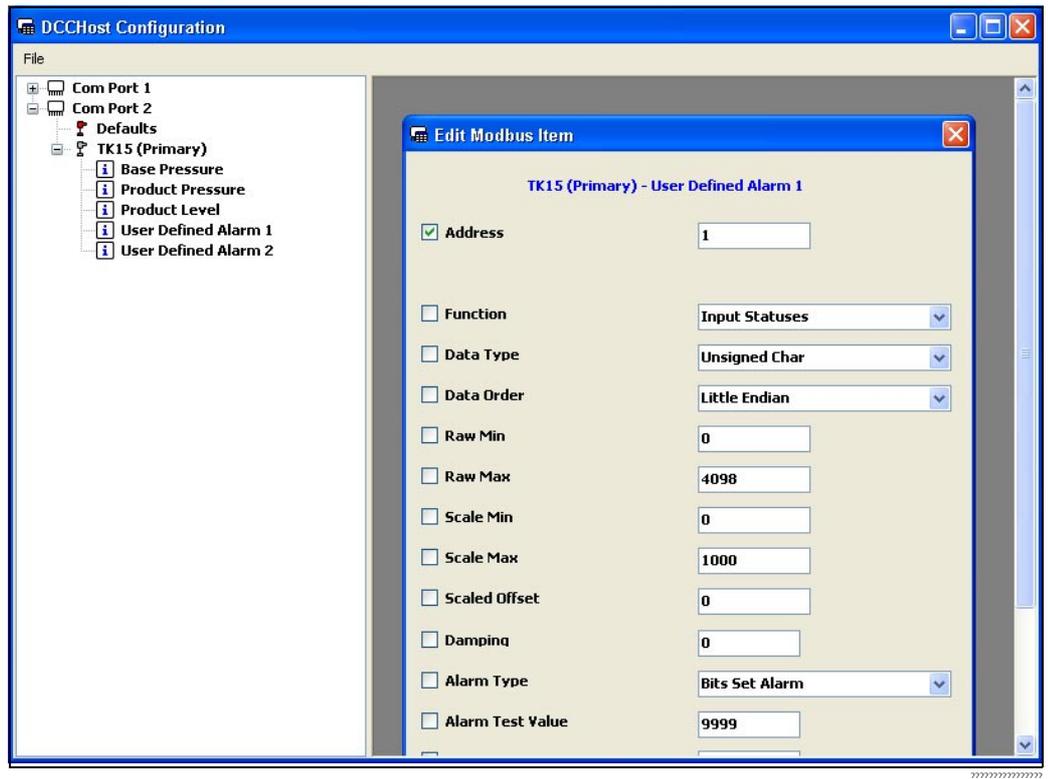
- User Defined Alarm 32

The alarms are available in the database for all gauge types; however whether they can be used depends on the physical gauge type. In fact only the Modbus TG, ModbusIO block, OPCTG and OPCIO block gauge type running on the direct interface currently support the use of User Defined Alarms.

To configure a Modbus TG gauge to poll for one of the eight User Defined Alarms, select the menu option **Configuration** → **DCC Host Configuration** and add the point to the required gauge.



These may be configured to read individual coils.



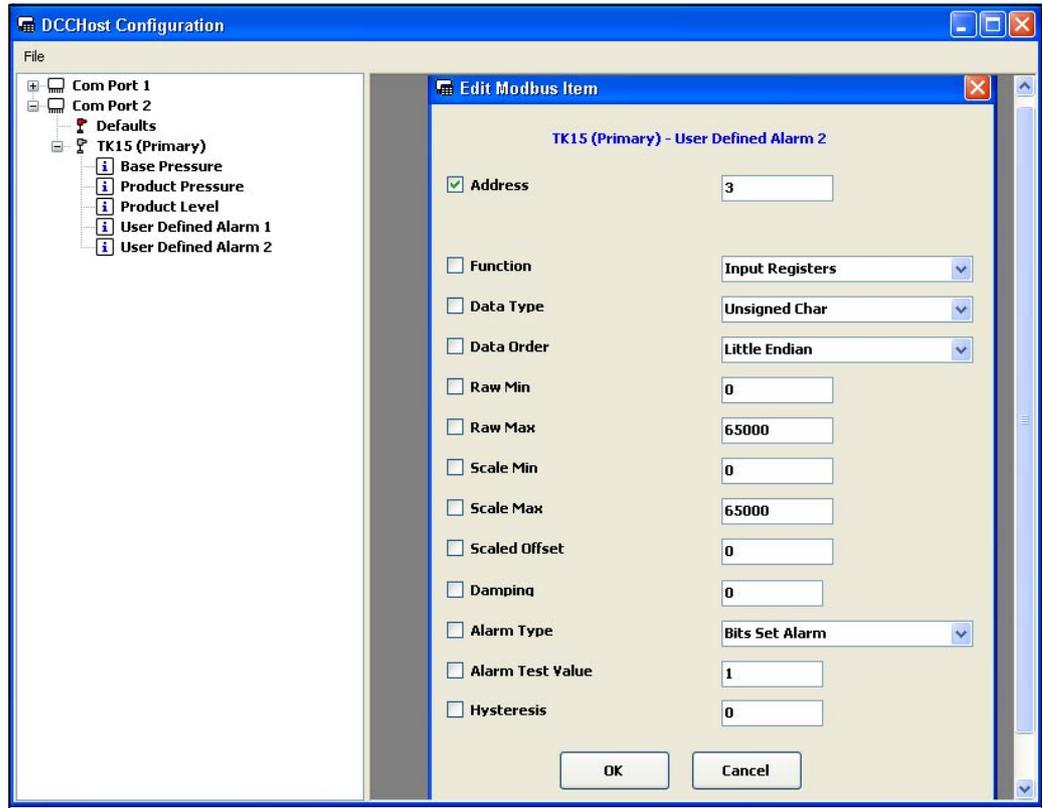
In the above configuration **User Defined Alarm 1** will be generated from an Input Status coil at address 1.

The **Alarm Type** field is used to set the activation of the alarm.

The **Alarm Test Value** will be used to test the incoming data to see whether the alarm is active.

Therefore in this case **Alarm Type** is set to **Bits Set Alarm** and the test value is 1. So if a value of 1 is received then the alarm will be activated, any other value will clear the alarm. The incoming value is a coil therefore it will be either 1 or 0.

User defined alarms may also be configured to read individual bits within a bit-mapped register. In that case the **Data Type** would be set to **Unsigned Short**, and the **Status Type** will define the bits to be tested.



In the above configuration **User Defined Alarm 2** will be generated from Bit 1 of an Input Register at address 3.

## 6 DCC Host Runtime

### 6.1 DCC Host

The DCC Host module is responsible for collecting data from the field, calculating Tank Inventory, originating alarms and events, and processing command requests from the client.

The DCC Host can communicate to a wide range of field gauges via a Datacon device or directly to some gauge types using the manufacturer's protocols.

Current protocols supported include:

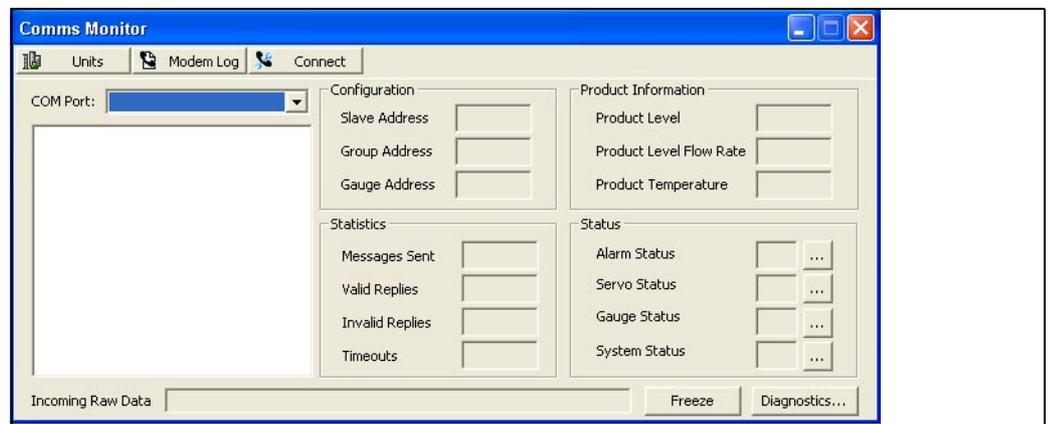
- Datacon protocol
- Enraf CIU
- Whessoe 550
- Whessoe 500
- Generic Modbus

The connection between the DCC Host module and the Datacon/gauges is normally by means of a physical hard wired serial interface connection, but may also be via a dial-up modem.

The DCC Host has the responsibility of determining the status of all gauge and system alarms. Furthermore the DCC Host module writes to the Events database.

### 6.2 Comms Monitor

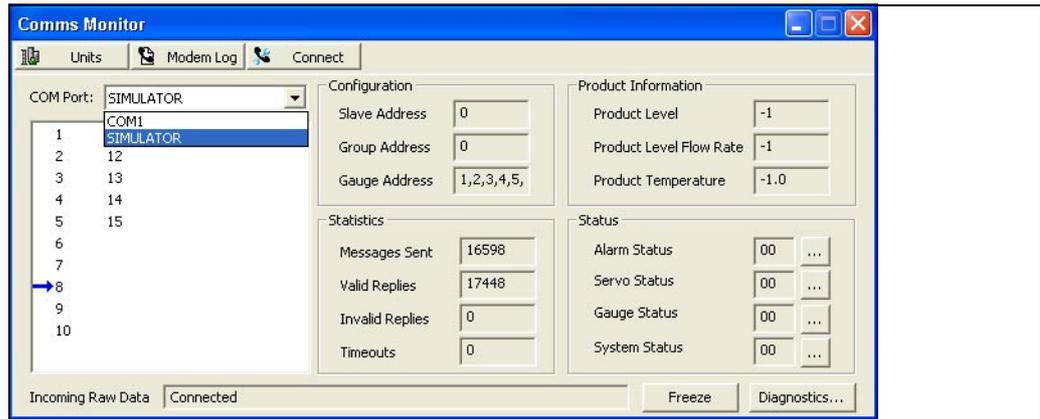
To view the DCC Host Comms Monitor either select **All Programs** → **Tankvision Multi Scan** → **DCC Monitor** from the windows start menu or click the **Comms Monitor** shortcut on the Windows desktop. This will bring up the following screen:



The DCC Host module allows the user to view the data communications that occurs between the PC and any connected devices such as Datacons or directly connected gauges.

From the drop down list select the **COM port** you chose for communication, e.g. COM 1. A list of gauges being polled on that COM port will be displayed in the box below.

One of the COM port selections available is the Simulator, this is a feature which simulates gauges within DCC Host and is treated exactly the same as a normal COM port.



NXA83\_Comms-Monitor\_COM-Port-Selection

The blue arrow indicates the gauge currently being polled. A red arrow will be displayed if a gauge is being polled on fast scan.

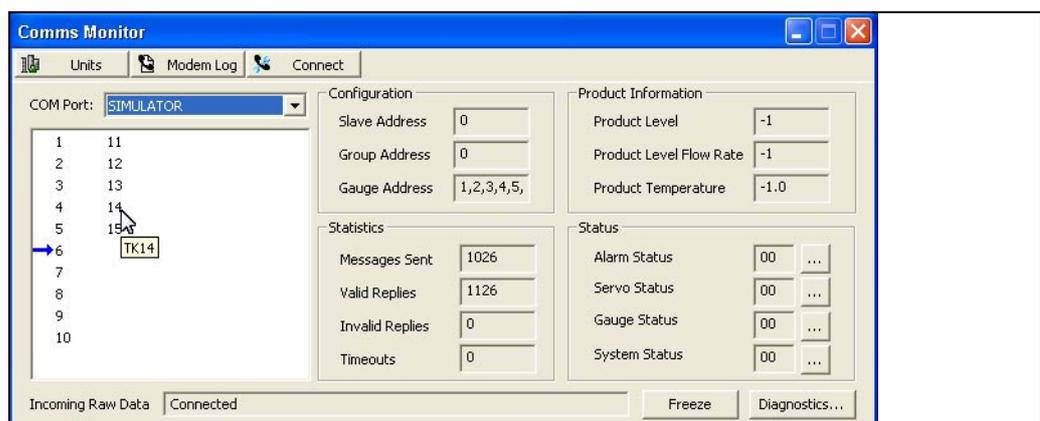
The **Statistics** section shows the current number of messages and responses since the interface was started. The **Product Information**, **Status** and **Incoming Raw Data** sections will only be updated if a gauge is selected to be watched, see next section for details.

If no gauge is selected to be watched then the **Incoming Raw Data** section will display the connection status of interface, this will be one of the following:

- Disconnected
- Connecting
- Connected
- Disconnecting

If an interface is not in the connected state then no messages will be sent to any gauges on that interface.

The list of gauges actually displays the Tank Address of each gauge entered in the **Gauge Configuration** screens. To display the name of the gauge in the list, hover the mouse over the Tank Address of the gauge.



NXA83\_Windows-Desktop\_Comms-Monitor-Window\_Gauge-Name

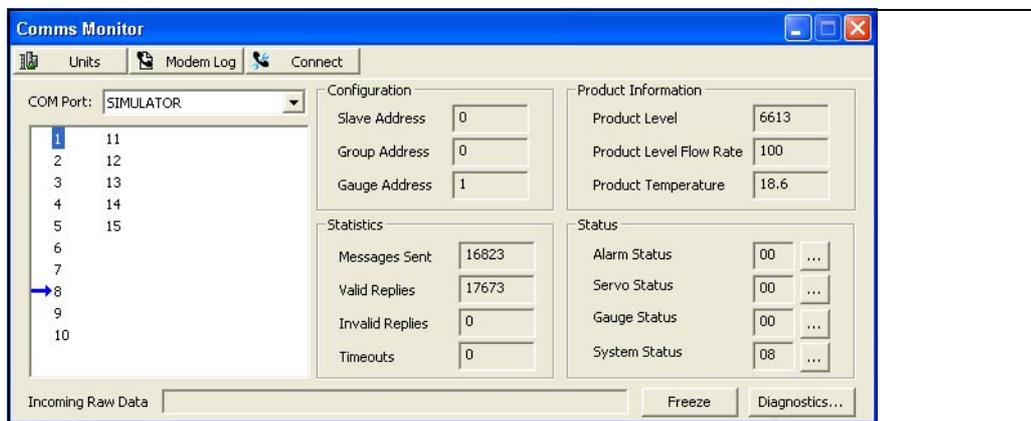
Viewing data in the DCC Host Comms Monitor is only really useful for maintenance and diagnostic reasons.

For general operations we do not recommend using the DCC Host Comms Monitor.

### 6.3 Watching a Tank

Sometimes it is useful to watch a specific tanks data.

Click on the **Tank number** to be viewed, e.g. Tank 1:

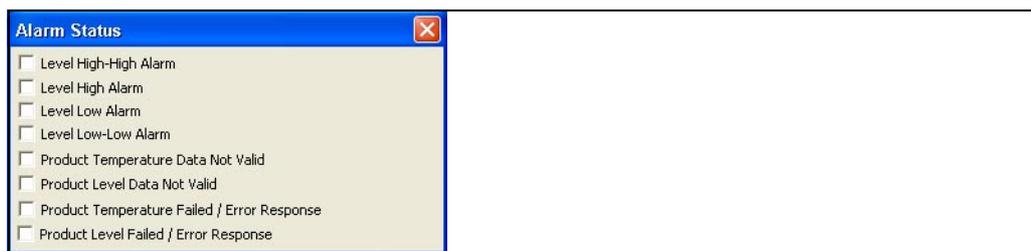


Each time the DCC Host polls that gauge (either directly or via a Datacon) the response and data will be captured and displayed.

The **Incoming Raw Data** field will display the raw data from the device, and to aid analysis can be frozen by clicking on the **Freeze** button.

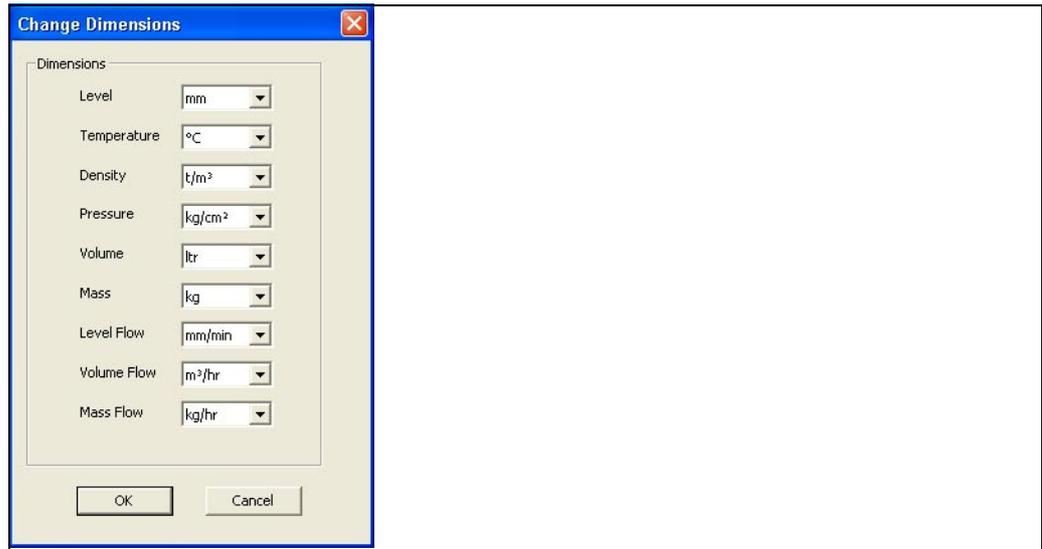
This does not stop the devices from being polled, but they will not be displayed on the screen. The transmitted data from the Datacon or gauge can then be analysed. This will require a detailed knowledge of the protocol being used.

The status fields returned by the device can be viewed in more detail by clicking on the button to the right of the required status.



Any status fields that are currently active (i.e. set) will be ticked.

The units of data displayed in the Comms monitor may be changed by clicking on the **Units** button at the top left of the screen.



NXA83\_Comms-Monitor\_Change-Dimensions

These units only apply to data displayed in the Comms Monitor, they do not affect data saved in the database or displayed in the standard screens.

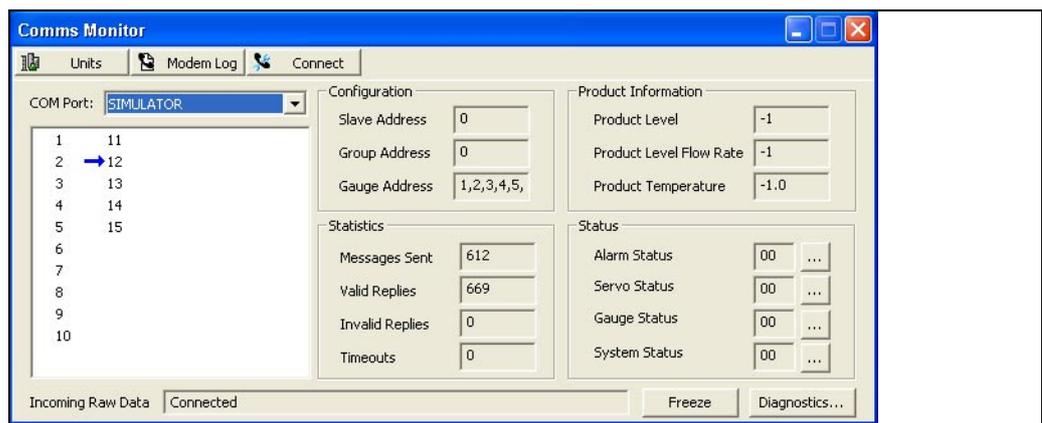
## 6.4 Viewing Modem Log

DCC Host maintains a log of the connection status of all interfaces and it is sometimes useful to view this log, especially for interfaces using a modem and scheduled connection.



Although it is referred to as "Modem Log" it actually applies to ALL interfaces within DCC Host, including those using serial ports on the local PC.

To view the log of all connections since DCC Host was started, click on the button **Modem Log** at the top of the Comms Monitor screen.



NXA83\_Windows-Desktop\_Comms-Monitor-Window\_Simulator

The file will then be automatically displayed using Notepad.

```

DCCHostConnection_Dump.log - Notepad
File Edit Format View Help
2012-10-25 02:10:42 DCCHost Starting
2012-10-25 02:10:42 Waiting for Hermes
2012-10-25 02:10:46 Hermes initialised
2012-10-25 02:10:46 Created RTDB readers and writers
2012-10-25 02:10:47 DCCHost Started
2012-10-25 02:10:47 COM 1 : Line Connected
2012-10-25 02:10:48 COM 1 : Polling Running
2012-10-25 02:10:48 COM 100 : Polling Running
NXA83_Comms-Monitor_Modem-Log

```

The file contains a date/time stamped log of any changes in connection status for all interfaces in DCC Host. It is especially useful for interfaces using Modem dial-up to connect to remote sites, as it details all the modem state changes and any line failures etc.

Any time that an interface is connected or disconnected it will be logged here, whether it be from a scheduled connection or from a manual user command.



All entries for COM 100 refer to the simulator, built in to DCC Host, which is treated the same as any other interface.

## 6.5 Connection Status

Gauges will only be polled when an interface is connected to the hardware it uses to communicate. This hardware could be an RS232 serial port on the PC or it could be a modem used to dial up a remote site.

The connection type of an interface is automatically set during the configuration of DCC Host (using DCC Config) and may be **Always** connected, **Manually** connected or connected at **Scheduled** times. However the user may manually override the automatic settings and command an interface to connect or disconnect at any time.

The connection status of a tank is normally displayed on the main home page, but this may vary depending on what the tankgauging system is.

If the tank displays any other status than **Connected** then the data is NOT live for that tank, but will be the values polled when the interface was last connected.

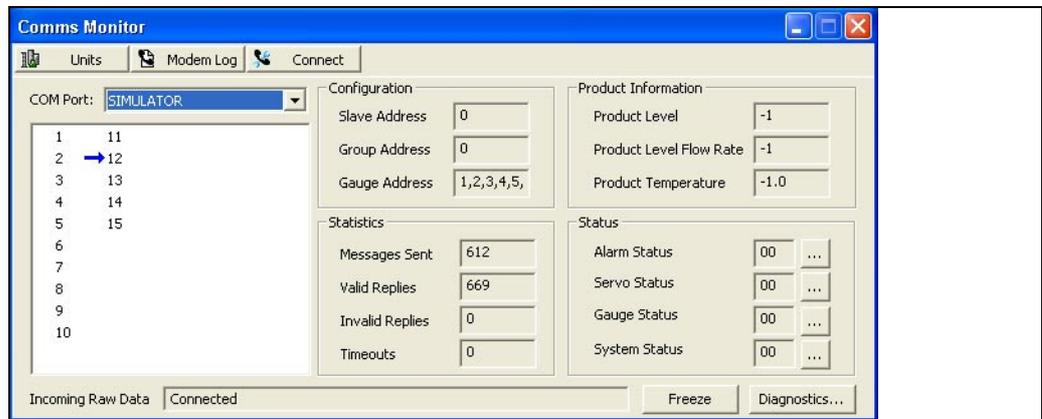
The connection status of an interface may be manually changed by the user by right clicking on any tank on that interface, and selecting either **Connect**, **Disconnect** or **Connect Once** from the **Gauge Commands** menu.

This command will apply to the interface as a whole and thus will affect ALL gauges on that interface. If the **Disconnect** option is selected a confirmation dialog will be displayed warning the user that this will result in the gauges no longer being polled and data on screen will be frozen for those tanks.

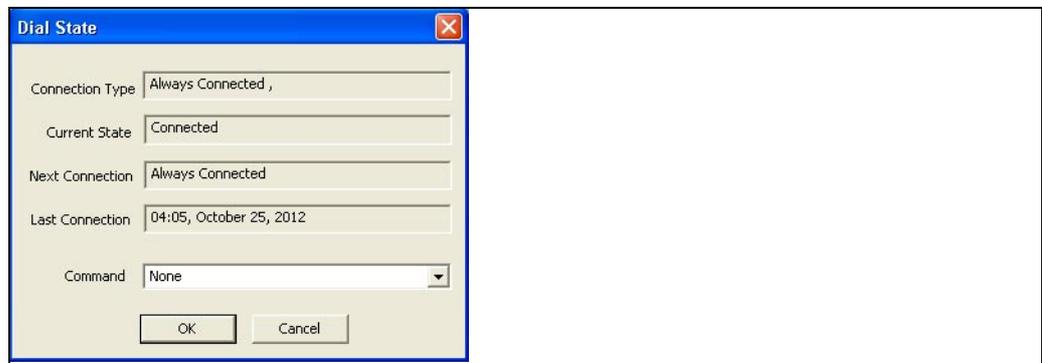
The difference between **Connect** and **Connect Once** is as follows:

- **Connect** – this will connect and stay connected permanently or until commanded to disconnect by the user
- **Connect Once** – this will connect and then automatically disconnect after it has retrieved all data from the gauges

The connection status of an interface in DCC Host may also be viewed and modified via the Comms Monitor. To view the connection status, select the COM port of the interface from drop down then click on the **Connect** button at the top of the dialog.



NXA83\_Windows-Desktop\_Comms-Monitor-Window\_Simulator



NXA83\_Comms-Monitor\_Dial-State

The connection details for the interface will be shown, including the **Connection Type**, the **Current State**, and the time of the **Last Connection** and time when the **Next Connection** will take place. This last field will only be meaningful if the interface has been configured for scheduled connections.

The **Command** field allows the connection state to be changed. To change, select the new state from the drop down list.



NXA83\_Comms-Monitor\_Dial-State\_Command

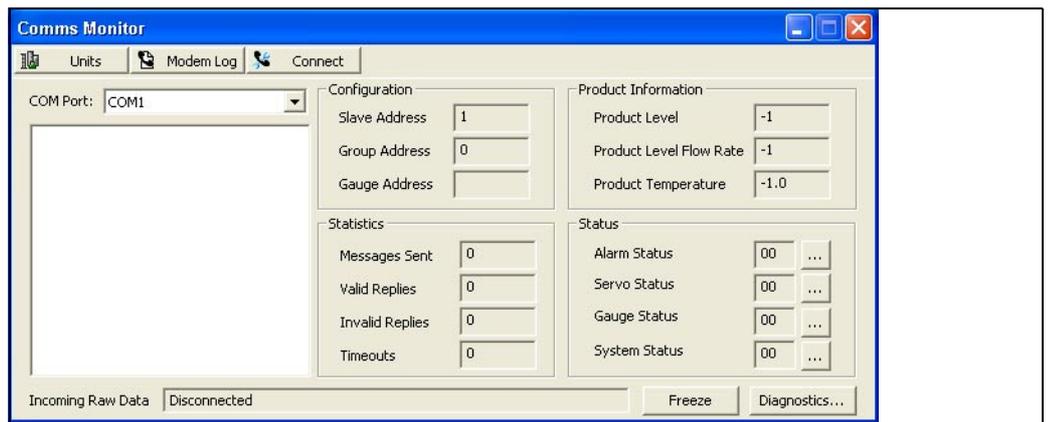
The states selectable are:

- **Connect** – this will connect and stay connected permanently or until commanded to disconnect by the user
- **Disconnect** – this will disconnect the interface

- **Scheduled Once** – this will connect and then automatically disconnect after it has retrieved all data from the gauges
- **None** – no change in connection state

Once the command has been selected click on **OK** to activate new state. If the command is **Disconnect** then a confirmation dialog will be displayed.

Once an interface has been disconnected then NO gauges will be polled. If the port is selected in the Comms Monitor, there will be no communication activity shown and **Disconnected** will be displayed in the **Incoming Raw Data** field.



## 6.6 Enhanced Raw Data

To view the raw data bytes transmitted and received on a serial port, click on the **Diagnostics** button. This will show an extended Comms Monitor dialog.



This will display a text box showing raw communication data and status. This is a live screen that will scroll the new lines of data as they occur.

If tank is set to watch (i.e. a gauge is selected in the gauge list box) then only raw data for that tank will be displayed, and that gauge id will be shown in the title for the raw data text box. If no gauge is selected then all raw data for all gauges will be shown, and the title box will display just the COM port number.

To close the raw data text box use the **Close Enhanced Raw Data** button. DO NOT use the **Close** button on the raw data screen.

To freeze the display so that it may be viewed, click on the **Freeze** button. Click on the **Resume** button to carry on updating the text box.



Freezing the raw display does not stop the actual communications to the gauges.

To log the raw data output to a file, click on the button **Raw Data Log To File**. The logged data will be saved to a file called "DCCHost\_Comms.txt". This will be located in the same directory as the DCCHost executable module. This location depends on the system and installation type, but is normally:

**"C:\program files\common files\tank farm automation\server"**

Or for embedded systems:

**"C:\program files\tanks"**

Click on the button again to stop logging to file.

## 7 DCC Slave Configuration

### 7.1 DCC Slave

The DCC Slave sub-system is related to the DCC Host and shares many of its control features. But whereas DCC Host is responsible for getting tank data into the system, DCC Slave is used to get data out of the system and into a Scada, DCS or similar system.

The DCC Slave is a standard sub-system within the application, running as a system service which will be automatically started when the PC starts. It is responsible for running multiple slave serial interfaces that allow other master systems (i.e. SCADA or DCS) to retrieve data from the systems real-time database (RTDB).

Multiple interfaces may be configured to communicate using one of the protocols currently available within DCC Slave, these being:

- Modbus (see note below)
- Enraf Host
- Datacon protocol
- Extended Modbus
- Alarm Relay Output (Modbus and internal relay card)
- TankMaster

The DCC Slave service is capable of servicing master systems on multiple COM ports, modems and network connections simultaneously, but before it can do so they must be configured within the service.



When the Modbus protocol is selected for a network port, then the variant protocol known as ModbusTCP will actually be used.

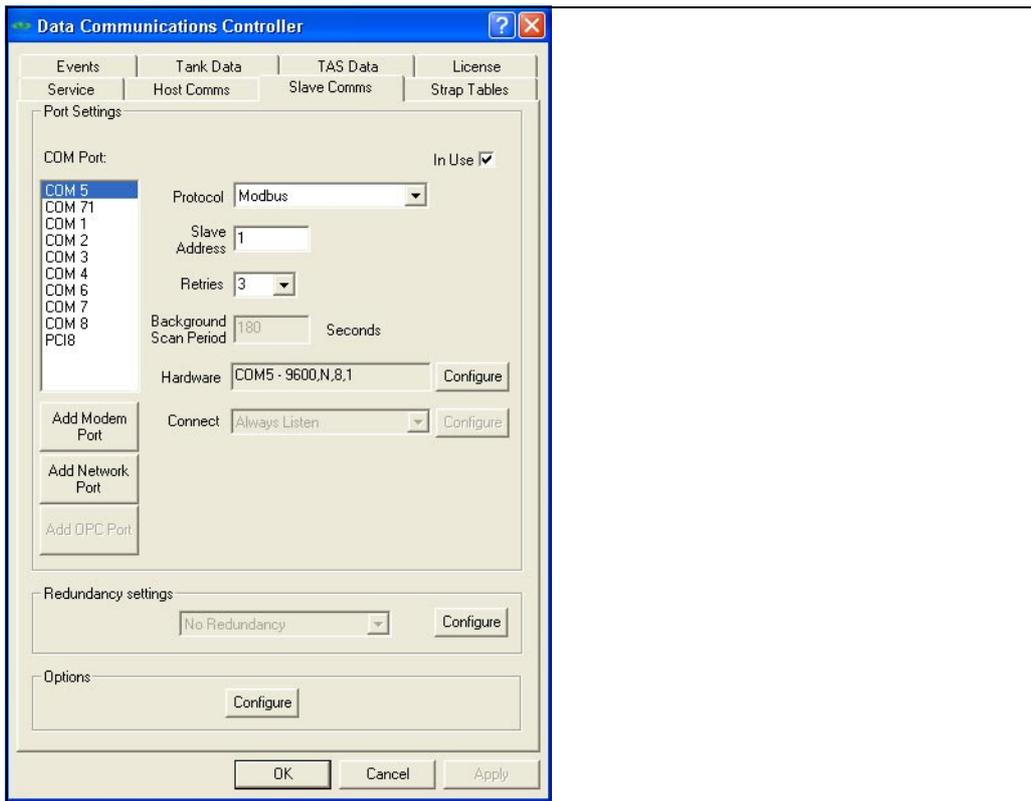
### 7.2 Configuring Slave COM Ports

The DCC Slave service requires the selection and configuration of the communication ports that are to be used.

Multiple serial ports, modems and network connections may be configured, up to the maximum licensed.

To configure the ports, run the DCC Configuration tool from the menu item:  
**Windows Start → All Programs → Tankvision Multi Scan → DCC Configuration.**

Click on the **Slave Comms** tab.



NXA83\_DCC-Config\_Slave-Comms

A list of all physical COM ports (RS232/485/422 etc.) will be displayed together with any logical Modem and Network ports configured (they will have an offset of 50 for modems and 70 for networks i.e. COM 51 etc.). See the later sections on Modem and Network Ports for details on how to add and configure these ports.

If one or more Arcom PCI8 relay cards have been installed in the PC then item PCI8 will also be displayed in the COM port list.

Select the COM port to be configured and make sure the **In Use** box is ticked. The protocol and communications parameters can now be edited.

The **Protocol** field determines what is to be connected to the COM port. The current selection options are:

- Modbus
- Enraf Host Protocol
- Enraf CIU Slave
- Datacon
- Extended Modbus
- Modbus Alarm Output
- Redundant Link Control
- ASCII Alarm Output



When the Modbus protocol is selected for a network port, then the variant protocol known as ModbusTCP will actually be used.

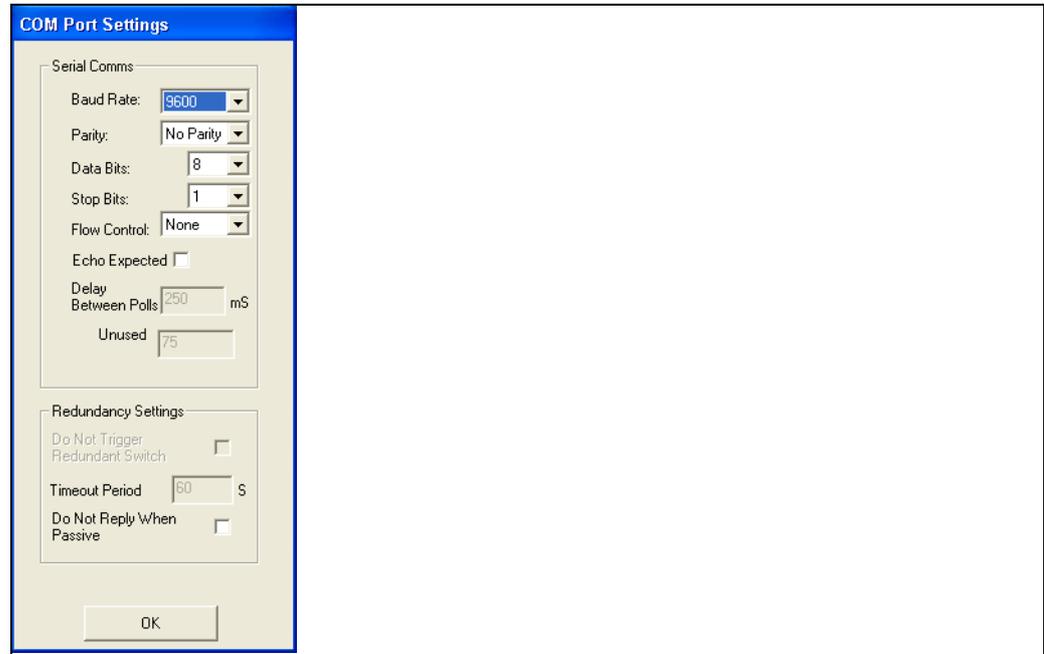


Redundant Link Control is only ever used if the **Switch By System** redundancy mode is selected.

The Slave Address must be set for all Modbus based protocols and also for the Datacon protocol and is the device address that the interface will reply to.

The **Retries** field does not apply to most protocols at the present time.

The communication parameters will be displayed in the Hardware section. For a serial COM port the baud rate, data bits, parity and stop bits will be displayed. These should be set to the same settings of that of master system that will poll this slave interface. They may be altered by clicking on the **Configure** button to the right, and editing the parameters in the dialog screen.



NXA83\_DCC-Config\_Slave-Comms\_COM-Port-Settings

If the serial communications hardware produces an echo of sent characters (i.e. some RS485 or 20mA current loop devices), tick the **Echo Expected** field to tell the driver to handle the echoed characters.

Modem ports will display only the word "Modem" in the **Hardware** section. The modem may be configured via the **Configure** button. See the later section on Modem Ports for more details.

Slave interfaces are **Always** connected to the serial COM port or modem they use to communicate, therefore the **Connect** field will be fixed in this mode.

A warning message may be displayed if a new interface is added by selecting the **In Use** box and the number of licensed host interfaces has been exceeded.



NXA83\_Data-Communications-Controller\_Host-Comms\_Interfaces-exceeded

When all slave interfaces have been configured or modified, click on the **Apply** or **OK** buttons to save any changes. There may be a momentary delay whilst the details are saved and the DCC Slave service automatically re-configures itself with the new configuration.

### 7.3 Configuring Modem Ports

Ports in DCC Slave can be configured to automatically accept connections from remote clients using locally connected (or internal) modems. They will then reply to all requests from remote master until it hangs up the line.

Modem ports are to DCC Slave as software COM ports denoted by an offset of 50, thus the first modem COM port will be COM 51.

They are treated as if they were physical COM ports on the PC, with only one slave interface accessing one physical modem device.

**i** Once a modem has been assigned to a slave interface it can ONLY be used by that interface. No other interfaces in DCC Slave or DCC Host can use that modem. The reason for this is that when a slave interface starts it takes control of the modem and does answer any incoming calls on that device.

Configuring modem ports in DCC Slave is very similar to configuring modem port in DCC Host. Please refer to the previous section in manual for details. The only differences are that no **Phone Number** need be entered and the **Connect** field will be fixed as **Always**.

### 7.4 Configuring Network Ports

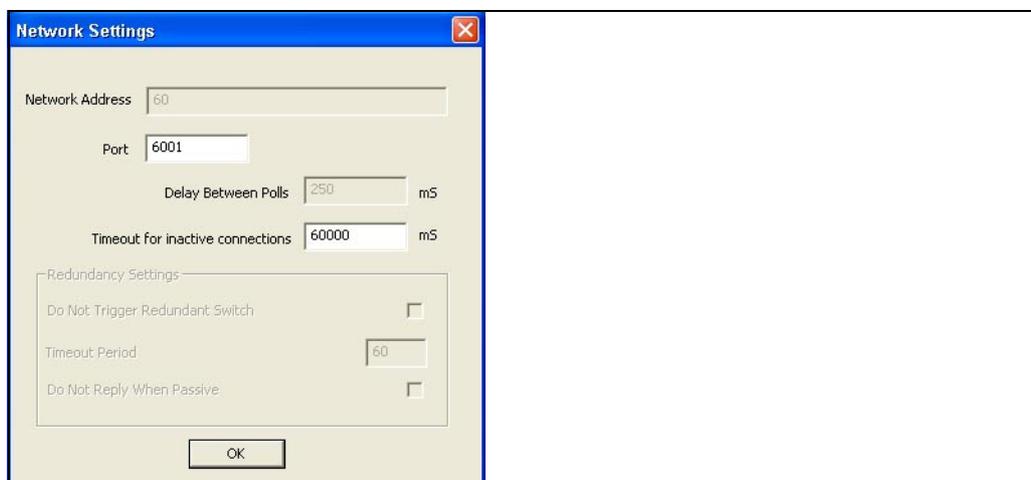
Ports in DCC Slave can be configured to automatically accept connections from remote clients using network socket connections. They will then reply to all requests from remote client until it disconnects the network socket.

Modem ports are to DCC Slave as software COM ports denoted by an offset of 70, thus the first modem COM port will be COM 71.

Software COM ports are treated the same as real COM ports with regards to licensing.

A network port makes no difference to the protocol selected to run on that port, except for the Modbus protocol. If that protocol is selected then it will use the Modbus variant called ModbusTCP.

Configuring network ports in DCC Slave is very similar to configuring network ports in DCC Host. Please refer to the previous section in manual for details. However there are a number of differences when configuring the network settings.



mmmmmm

The only parameters that require configuring are the TCP/IP **Port** that the slave interface will listen on, and the **Timeout for inactive connections**.

If no requests have been received in the slave for longer than the timeout value, then the socket will be automatically disconnected. It should not be reduced too low, as this may affect the stability of the connection. Generally keep this value above at least 10 seconds and the recommended value is 60 (60000 ms).

Ensure that the Port value does not clash with any ports already in use with other programs or services on the PC. The range for the port field is 1 to 65 000, although normally it is set less than 10000, and it is recommended that it should be above 1000 to avoid a range of addresses commonly used by system processes.

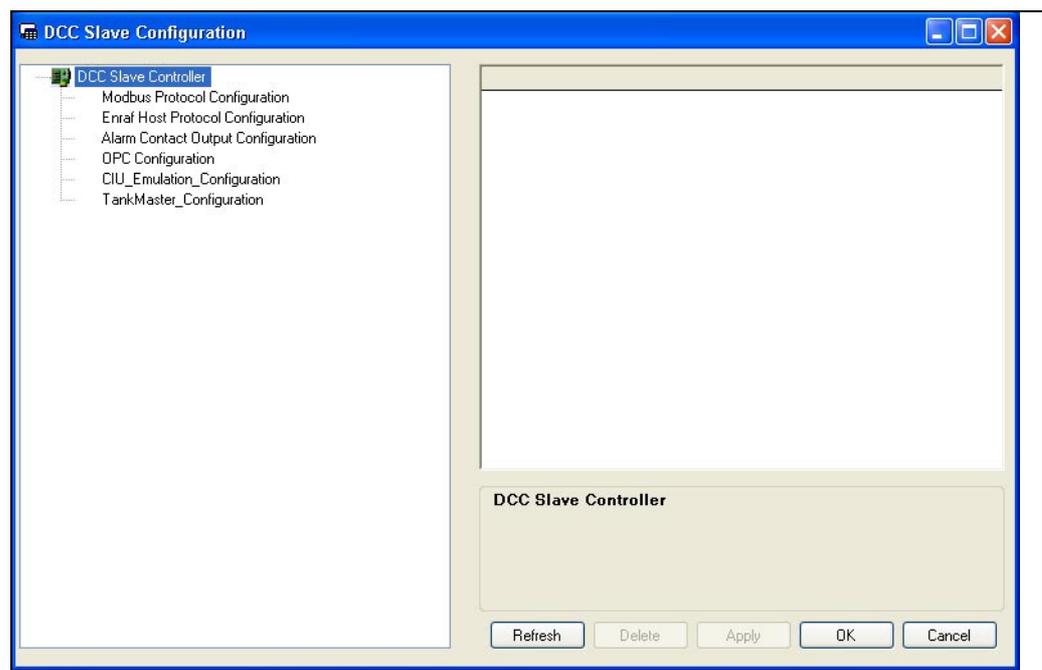
If in doubt then start the command prompt and type in **Netstat-a**.

The second column will show all ports currently in use.

## 7.5 Protocol Configuration

Most communication protocols in DCC Slave require some configuration data to be entered, which is specific to that protocol (i.e. Modbus requires a register/coil map to be set up).

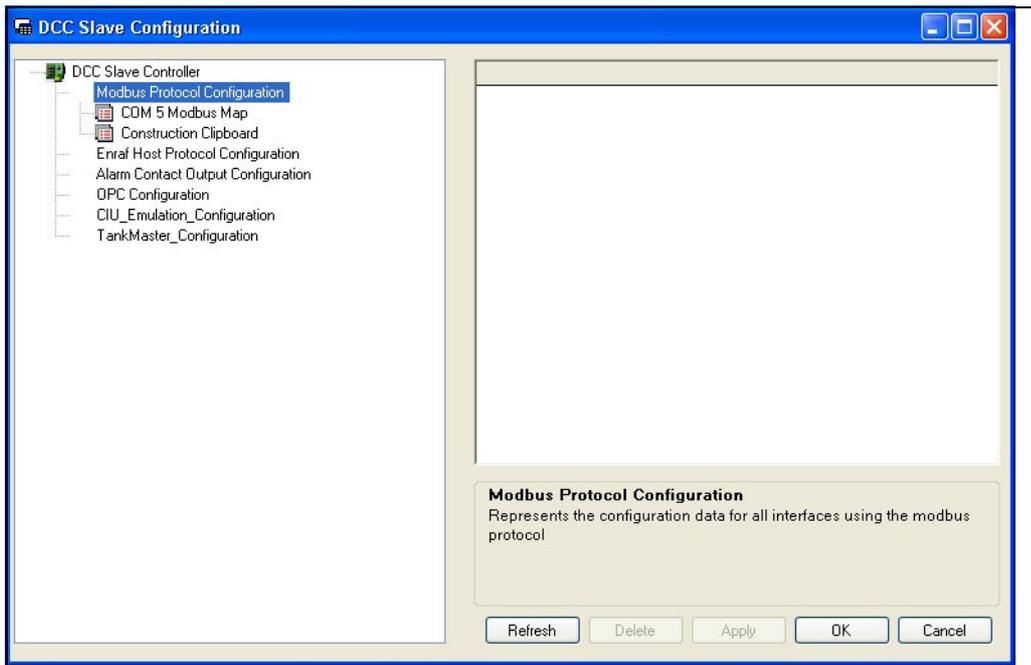
To configure any protocol specific data, select **DCC Slave Configuration** item from the **Configuration** menu.



NXA83\_DCC-Slave-Configuration

The display consists of a left hand panel with a tree list of all protocols supported by DCC Slave and a right hand panel where selected data may be edited. To navigate through the tree structure simply left click on the item to expand it.

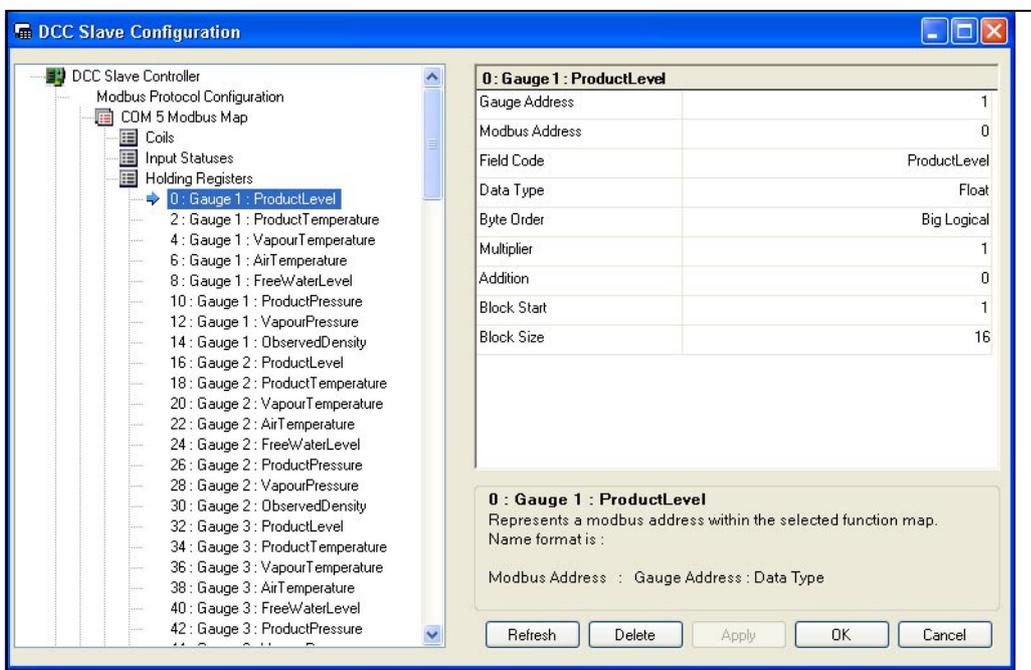
Expanding each protocol will display a list of all COM ports currently configured for that protocol.



NXA83\_DCC-Slave-Configuration\_Modbus-Protocol-Configuration

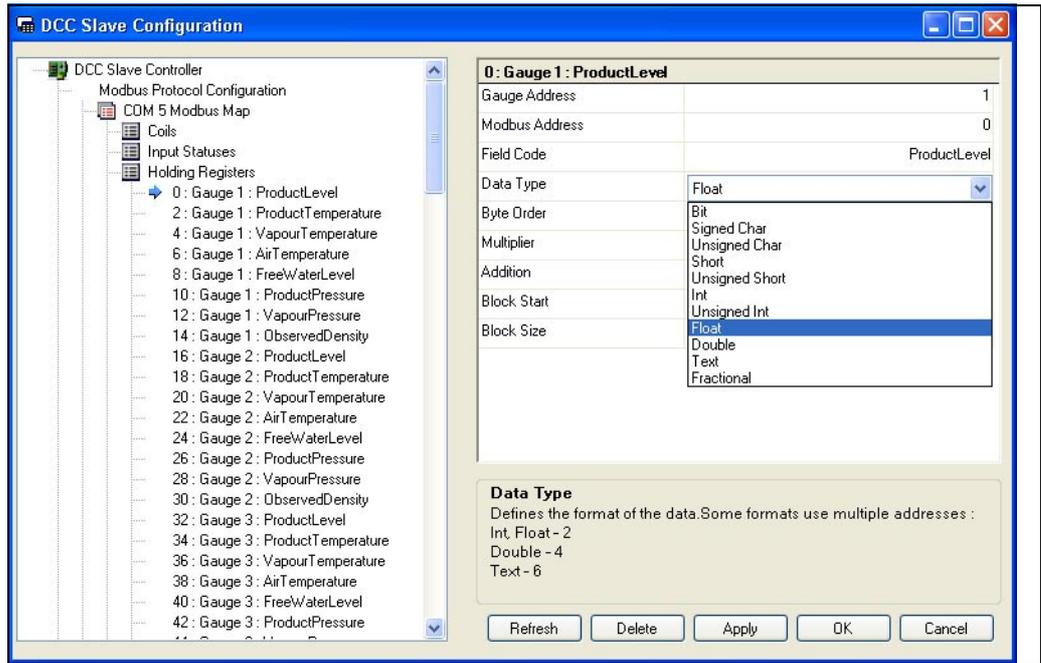
Under each COM port entry will be displayed the currently configured protocol specific data for the interface running on that port. The protocol specific data options are described in the relevant section on that protocol, further on in this document.

If there are any configurable items associated with a currently selected tree item then they will be displayed in the right hand panel.



NXA83\_DCC-Slave-Configuration\_Modbus-Protocol-Configuration\_Holding-Registers

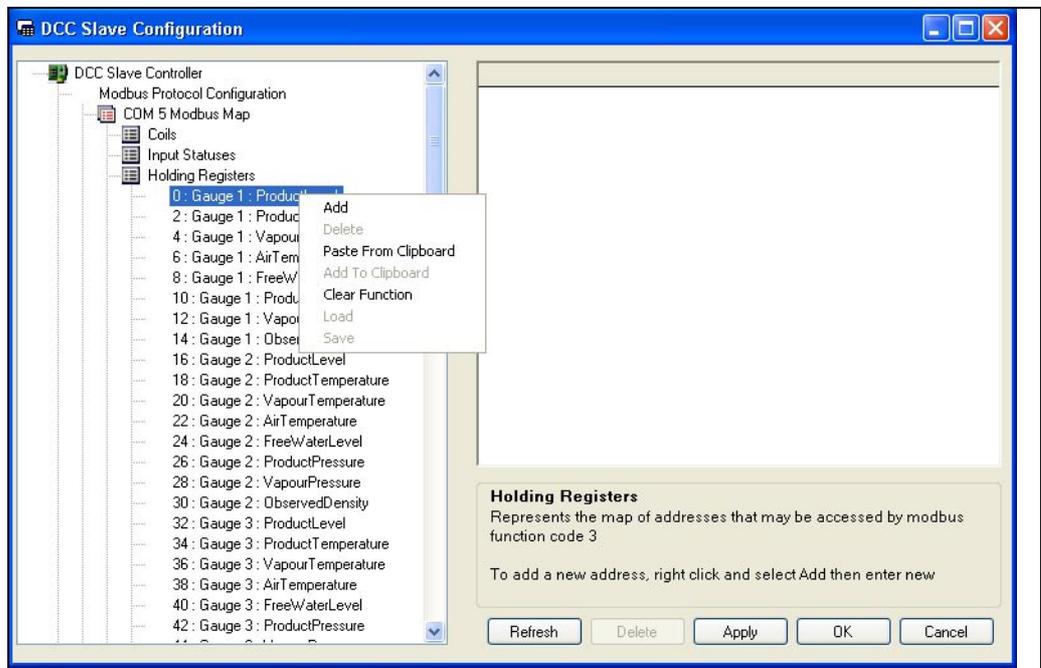
To edit configuration data displayed in the right hand panel simply click on that item and either edit the text or select from the drop down list.



NXA83\_DCC-Slave-Configuration\_Edit-DataType

To save any data to the database, click on the **Apply** button. To save data and refresh the left hand panel tree display, click on the **Refresh** button. This button will also update the display if a new interface has been added to DCC Slave using the control panel applet while in this program.

Right clicking on tree items will display a pop-up menu with any allowed actions available for that item. The actions depend on the protocol and are detailed later in this document.



NXA83\_DCC-Slave-Configuration\_Right-Click-Pop-Up

## 7.6 Modbus Protocol Configuration

The DCC Slave implementation of the Modbus protocol currently supports the following functions:

- 01 Read Coil Status
- 02 Read Input Status
- 03 Read Holding Register
- 04 Read Input Register
- 05 Force Single Coil
- 06 Preset Single Register
- 08 Diagnostics – sub-task 00 only
- 16 Preset Multiple Registers

The following sections offer a quick getting started guide to configuring interfaces in the DCC Slave that use the Modbus Datacon protocol.

They assume a degree of prior knowledge of the Modbus protocol.

All Modbus interfaces have to be configured to participate in any communication with a host system.

The COM port parameters are configured in the standard control panel applet common to all protocols. However the protocol specific data must be configured using the **DCCSlaveConfigGUI** application.

### 7.6.1 Planning

Successful dialogue between two computers using the Modbus protocol does require some pre-planning.

In particular, the following issues will need to be defined and agreed with the supplier of the Modbus Host:

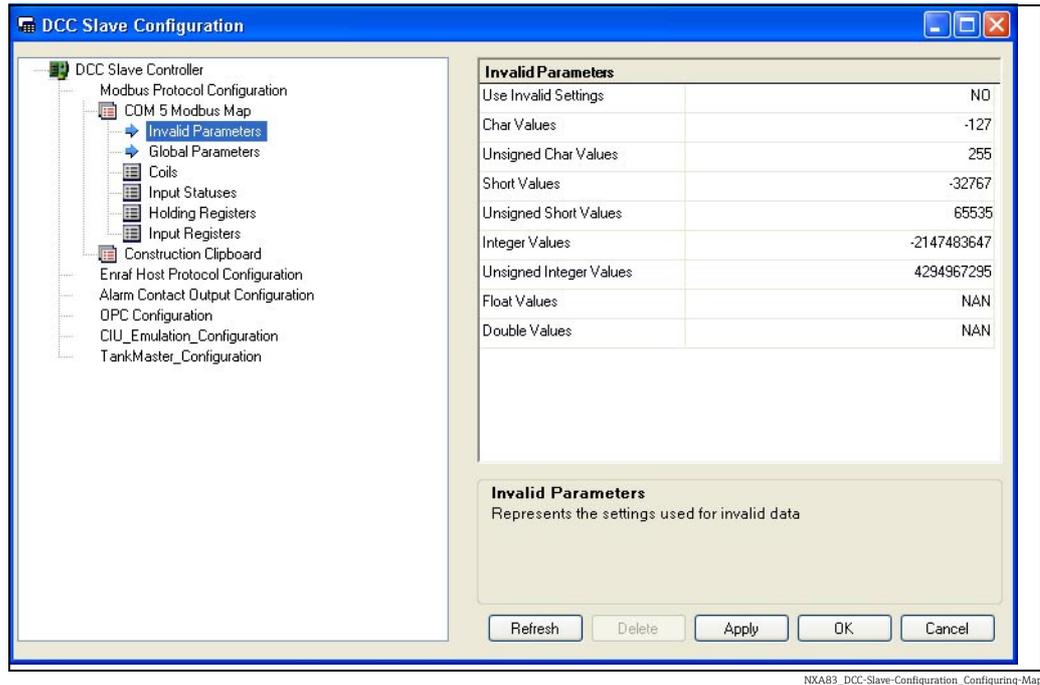
- Modbus link communication settings
- Modbus Slave Address
- Data Items to be transferred for each tank
- Engineering units for each data item
- Data type for each data item
- Register addresses for each data item

Although the Modbus Slave driver allows the user to have any data item in any order, it is often wise to order them in contiguous and consecutive addresses. Certainly in this case it is more efficient to process the data when they are ordered in ascending order of Modbus Address.

### 7.6.2 Configuring the Map

Start the **DCCSlaveConfigGUI.exe** module.

This module can be used to configure the map for any Modbus slave interface in DCC Slave.



NXA83\_DCC-Slave-Configuration\_Configuring-Map

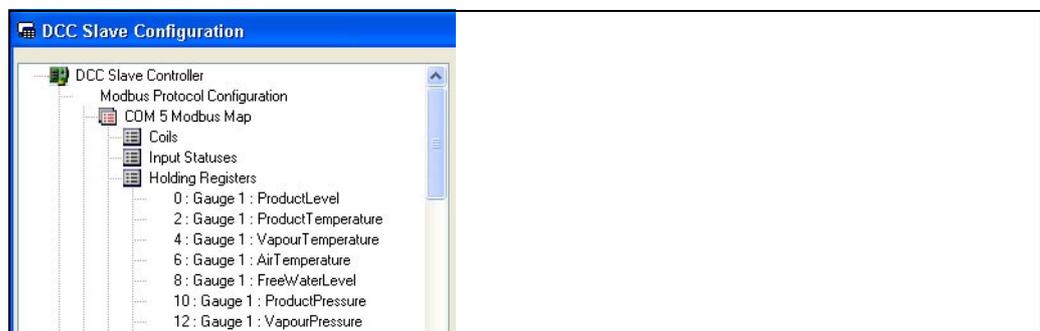
The DCC Slave configuration tool consists of a tree list of all protocols on the left and an edit section on the right. Included in the list will be the item **Modbus Protocol Configuration**, under which is a list of all COM ports using the Modbus protocol, plus the Construction Clipboard.

Each COM port item consists of a set Modbus map functions/addresses containing the actual map that the Modbus interface will use, split into the four Modbus function types:

- Coils
- Input Statuses
- Holding Registers
- Input Registers

The Construction Clipboard is a set of temporary maps that may be pasted to the actual Modbus map to speed up the configuration of large sites.

To display the map for a particular function type, click on it in the tree structure:



NXA83\_DCC-Slave-Configuration\_Tree-Structure

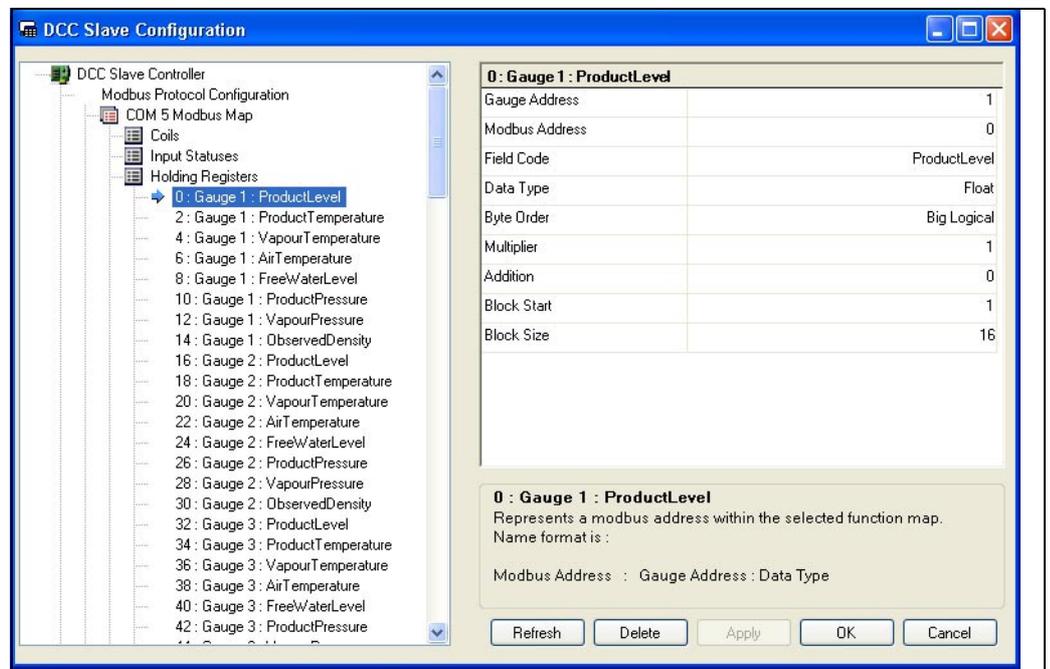
Each address will be displayed with the following naming convention:

- Address : Gauge Number : Data Type

The address is the Modbus address in the selected function map. The Gauge Number is the value assigned to a gauge for the field "Tank Address" in the database.

For Example, the item **0 : Gauge 6 : ProductLevel** in the Holding registers function map is the first holding register, which will return the Level for the gauge which has a "Tank Address" of 6.

Each address has a set of parameters used to format the data when sent to the host PC. To display the parameters for an address, click on that address in the left hand box and the data will be displayed in the right hand edit box.



NXA83\_DCC-Slave-Configuration\_Modbus-Protocol-Configuration\_Holding-Registers

The data will consist of:

- **Gauge Address:** A drop down list of all current gauges in the database, displayed with gauge address first, then the tank name.
- **Modbus Address:** This is a fixed field and cannot be modified.
- **Field Code:** A drop down list of all fields accessible for gauge. This list has a very large range of items and for ease of use may be filtered. See later for some notes on certain field types.
- **Data Type:** A drop down list of the data formats for the data. Note that some formats may take more than one address (i.e. Floats, Doubles etc.), thus care should be taken when adding more addresses to allow for this.
- **Byte Order:** A drop down list of the four basic byte ordering formats.
- **Multiplier**
- **Addition:** "Real" (decimal) number factors which are used to scale and offset values transferred over the Modbus network. Values read from the database are first multiplied by the Multiplier and then have the Addition factor added. The resulting value is then passed to the Modbus master device. Conversely, values sent from the Modbus master device have the Addition factor subtracted and then are divided by the Multiplier before being stored in the database.
- **Block Start**
- **Block Size:** These parameters are only required for block type fields such as profile data and skin temperatures. They should be ignored for all other types. They define the start position in the profile table (from 1 to 256) for the first point in the block and the number of consecutive points after that i.e. the size of the block.

Data may be modified then saved by clicking on the **Apply** button.

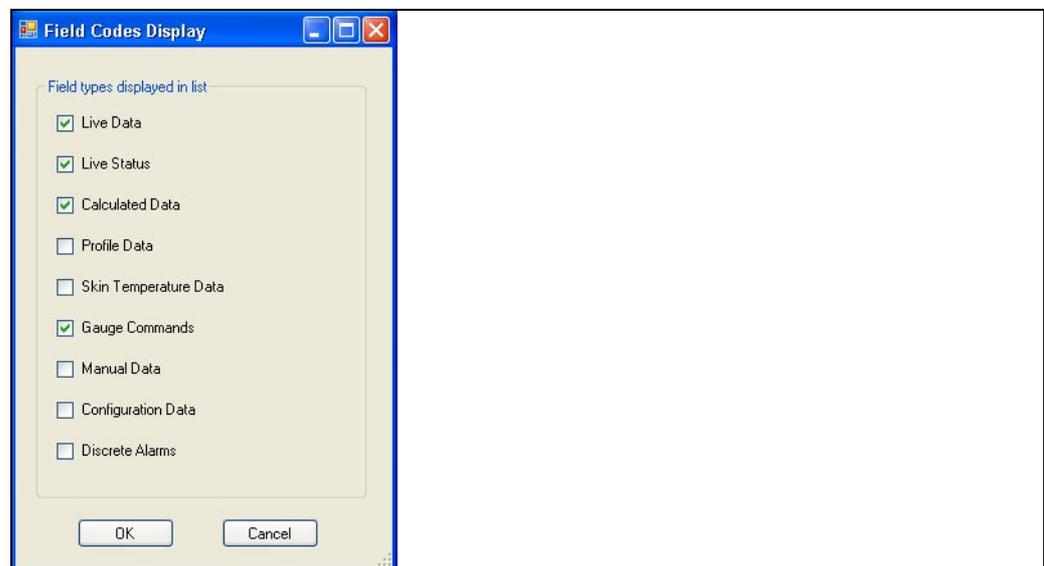
### 7.6.3 Filtering Field Code Selection

There are a large amount of options to select from the **Field Code** drop down menu.

To simplify this selection procedure, the drop down list may be filtered to display only certain groups of field codes types, defined below.

- **Live Data** – data generally returned from the gauges i.e. levels, temperatures densities etc.
- **Live Status** – status fields for the gauges
- **Calculated Data** – data calculated by the tank gauging system i.e. volumes etc.
- **Profile Data** – data from gauges that can carry out density/temperature profiles
- **Skin Temperature Data** – data from temperature multiplexors
- **Gauge Commands**
- **Manual Data**
- **Configuration Data**
- **Discrete Alarms** – live data alarms that may be accessed as individual bits

By default only **Live Data**, **Live Status**, **Calculated Data** and **Gauge Commands** will be displayed. To change the filter click on the **Field Code** line in the data parameter table.

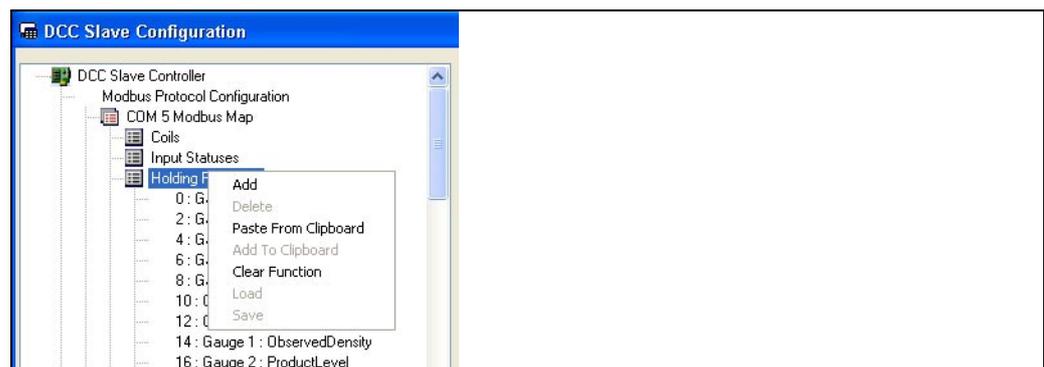


NXA83\_DCC-Slave-Configuration\_Field-Codes-Display

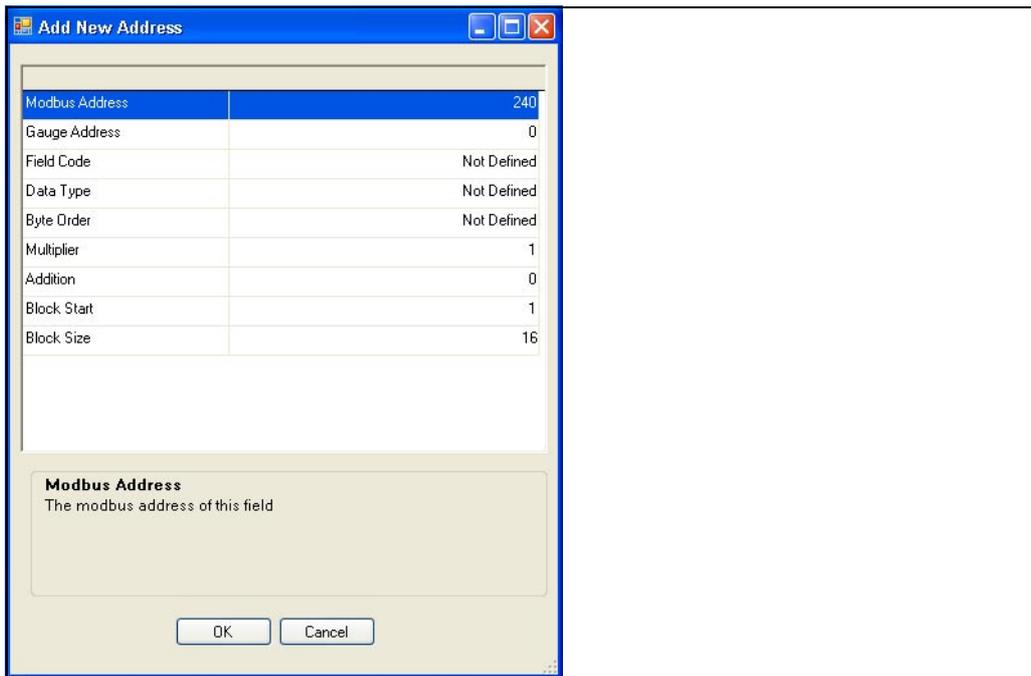
Make sure those groups to be displayed are ticked.

### 7.6.4 Adding and Deleting Entries

To add a new entry to the Modbus map click on the Modbus function for which the address is to be added. Right click to display the pop-up menu and select **Add**.



NXA83\_DCC-Slave-Configuration\_Add-Entry



NXA83\_DCC-Slave-Configuration\_Add-New-Address

The **Add New Address** dialog will be displayed, allowing the new address parameters to be entered.

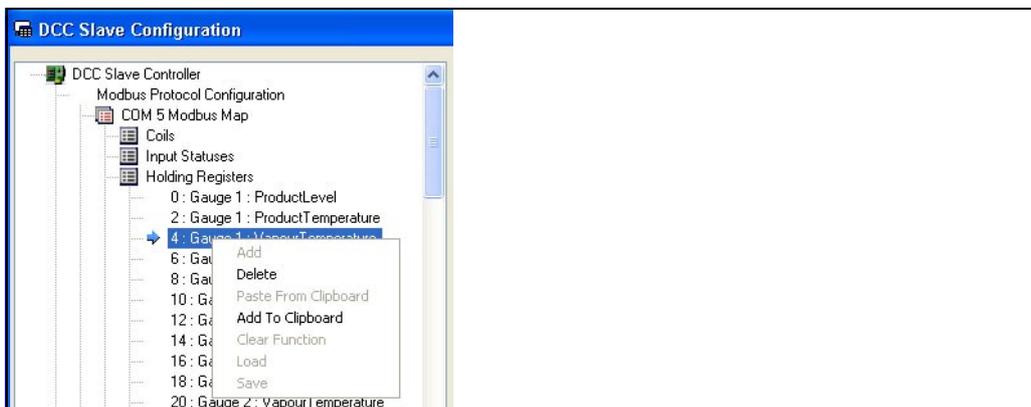
The data will initially be set to default values or the last address added, except for the **Modbus Address** field which will be set to the next free address in the function map.

After all parameters have been entered, click on **OK** to add the new address to the map.

If an illegal parameter has been entered then a warning message will be displayed. This applies particularly to the Modbus address which must be a unique value within the function map.

-  The address must be unique and if this data item is immediately following the previous one the next Modbus Address must take account of the size of the previous Data Type. For example, for an Integer the Modbus Address increments by 1, for a Float the Modbus Address increments by two etc.

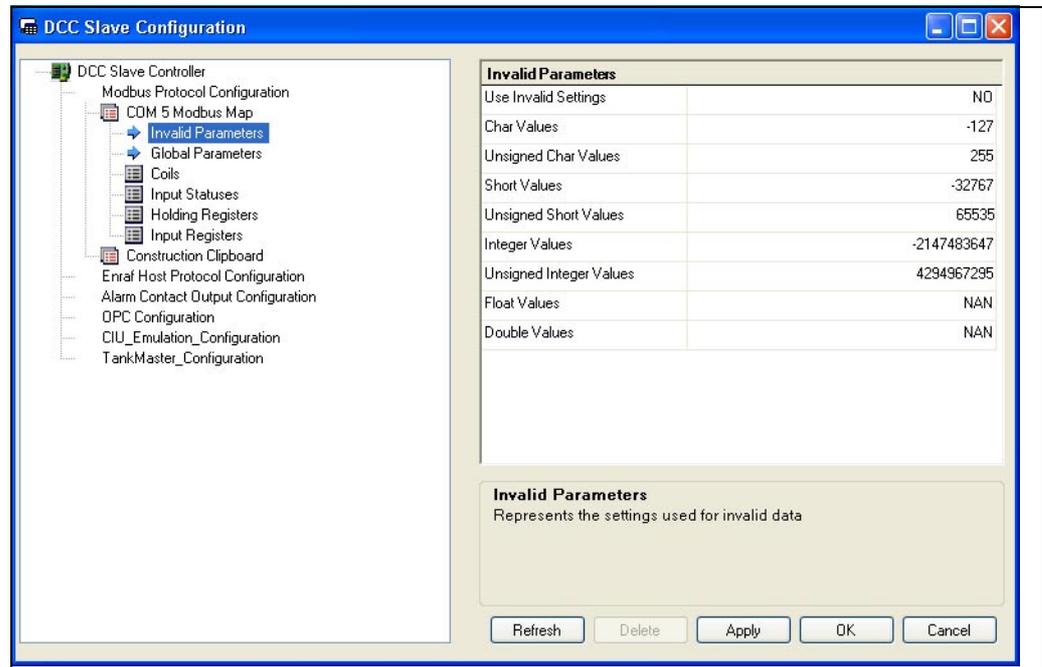
To delete an existing entry in the Modbus map, click on the address is to be deleted. Then right click to display the pop-up menu and select **Delete**.



NXA83\_DCC-Slave-Configuration\_Delete

### 7.6.5 Setting Invalid Parameter Values

Each COM port Modbus map has an entry where the invalid data settings can be modified.



NXA83\_DCC-Slave-Configuration\_Invalid-Parameters

These parameters are the values that data will be set to if they have an invalid status, depending on their configured Data Type. For instance if the **Char Values** field was set to -127 then any data configured as a char in the Modbus map would be set to -127 when its status is invalid.

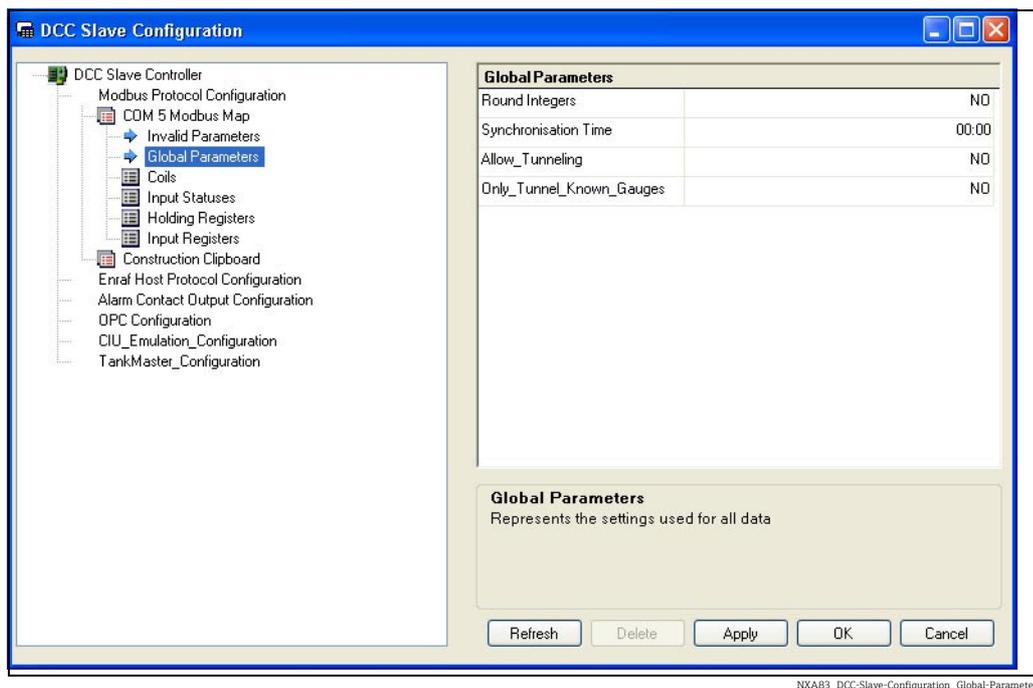
By default the **Use Invalid Settings** field will be set to NO, which means that the invalid data settings will not be used, and the data will retain its last good value. The status field for that data item would then have to be checked.

Changing the **Use Invalid Settings** field to YES will set all data fields to use invalid data values.

Float and Double values may be set to Zero, Nan (Not a Number) or Infinity. All these values are defined as part of the standard IEEE 754 definition of floating point numbers.

### 7.6.6 Setting Global Parameter Values

Each COM port Modbus map has an entry where the any global parameters settings can be modified. These are settings that apply to the interface as a whole.



The parameters are as follows:

- **Round Integers:** Defaults to NO, in which case any parameters configured as integers in the Modbus interface will be truncated when the value is extracted from the tank gauging database. If set to YES then values will be rounded.
- **Synchronisation Time:** This is the time that the system will be set to when the time synchronisation field is written to. See later section on Time Synchronisation for more details.
- **Allow Tunneling:** Defaults to NO, in which case any modbus requests for PLC addresses other than this slave interface will be ignored. If set to YES then any modbus requests for addresses other than this slave interface will be tunneled to all gauge interfaces that run the modbus protocol. The reply from the a gauge will be returned on this interface. This is mainly used with SAAB gauges.
- **Only Tunnel Known Gauges:** Defaults to NO. Tunneling must be enabled to use this parameter. If set to YES then tunnel requests will only be sent to any gauge that exists in the database. If set to NO then the tunnel request is broadcast to ALL gauge interfaces.

### 7.6.7 Construction Clipboard

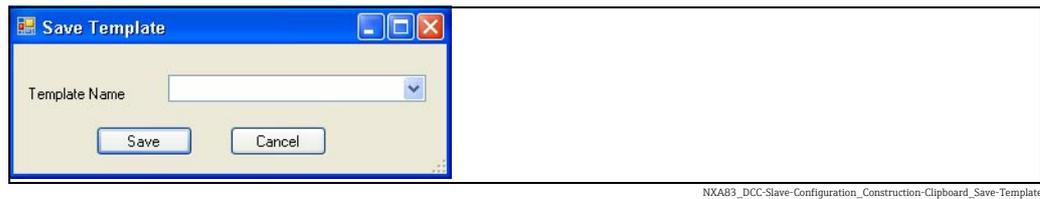
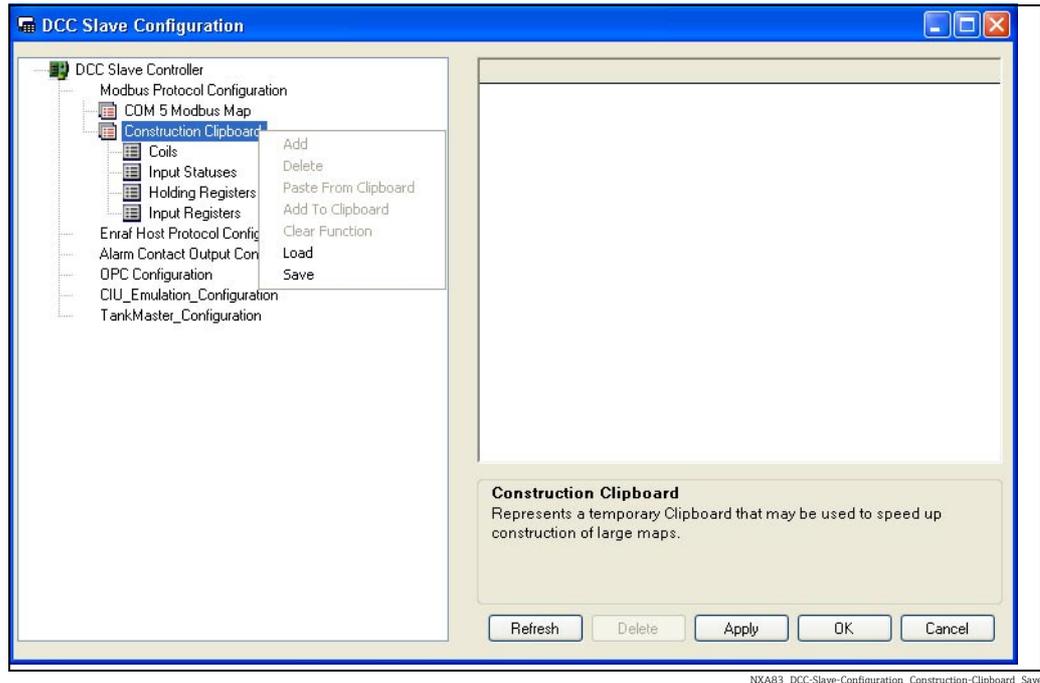
The Construction Clipboard allow common blocks of information to be pasted to the Modbus function maps repeatedly, thus reducing the time taken to create a large map.

There are a number of ways to add addresses to the clipboard:

- Hold down the CTRL key and click on each address to be added to the clipboard
- Right click on an address and select the **Add To Clipboard** option from the pop-up menu
- Select a Function in the Clipboard and add a new address in exactly the same manner as described previously in "Adding New Modbus Entries" (→ 64)

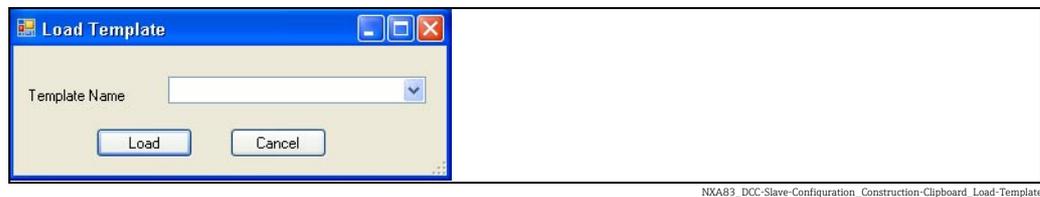
Any addresses added by the above methods will automatically be added to the same function type in the Clipboard as the source function. Thus if addresses from the Holding Registers function map are added to the Clipboard, they will be added to the Holding Registers function.

Once a set of blocks have been created for the Construction Clipboard, they may be saved to a Template for later use. To do this right click on the **Construction Clipboard** item and select **Save** from the pop-up menu.



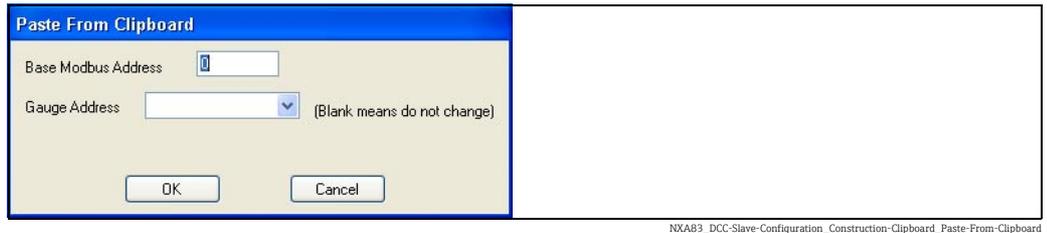
Enter the **Template Name** to save the Template under, or select an existing name from the drop down list. Then select **Save** to store this Template. Various different Templates may be saved.

Previous Templates may be loaded in the Construction Clipboard. To do this right click on the **Construction Clipboard** item and select **Load** from the pop-up menu.



Enter the **Template Name** of the Template to load or select from the list, then click **Load**. The existing clipboard entries will be cleared and new ones will be loaded from the Template.

To use a template to paste addresses to the Modbus map, right click on the function map to be added to, and select **Paste From Clipboard** from the pop-up menu. This will display the following dialog:



NXA83\_DCC-Slave-Configuration\_Construction-Clipboard\_Paste-From-Clipboard

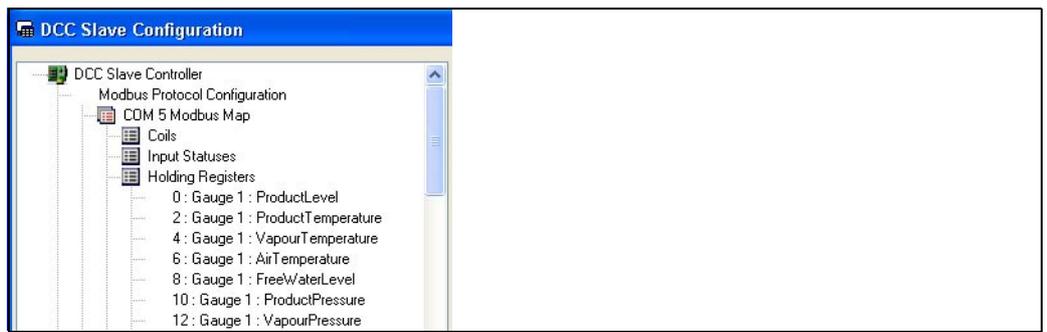
The **Base Modbus Address** is used to offset all the addresses in the clipboard when pasting to the destination.

If a Gauge Address is selected then all addresses pasted from the clipboard will have their Gauge Address field set to this value. Leaving this field blank will mean the gauge addresses will be as per the values in the Clipboard.

Click **OK** to add the addresses from the Clipboard to the map.

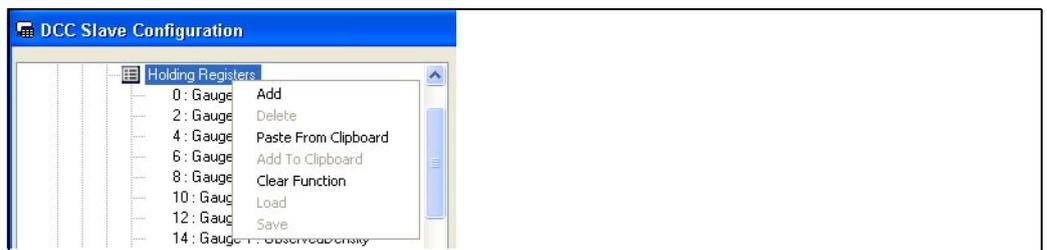
**EXAMPLE:**

If the Clipboard was setup for the Holding Registers as shown below:



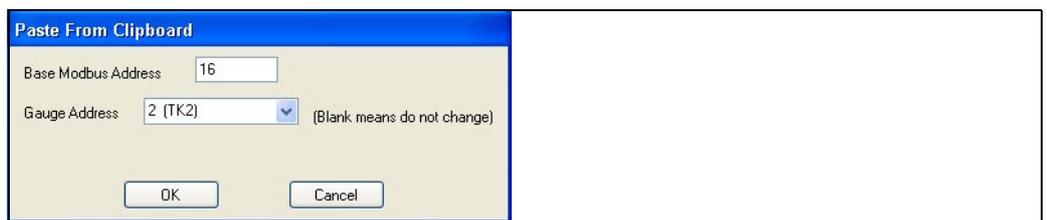
NXA83\_DCC-Slave-Configuration\_Construction-Clipboard\_Example\_Setup

To paste these registers to the actual Modbus map, select the Holding registers function map and right click to display the pop-up menu.



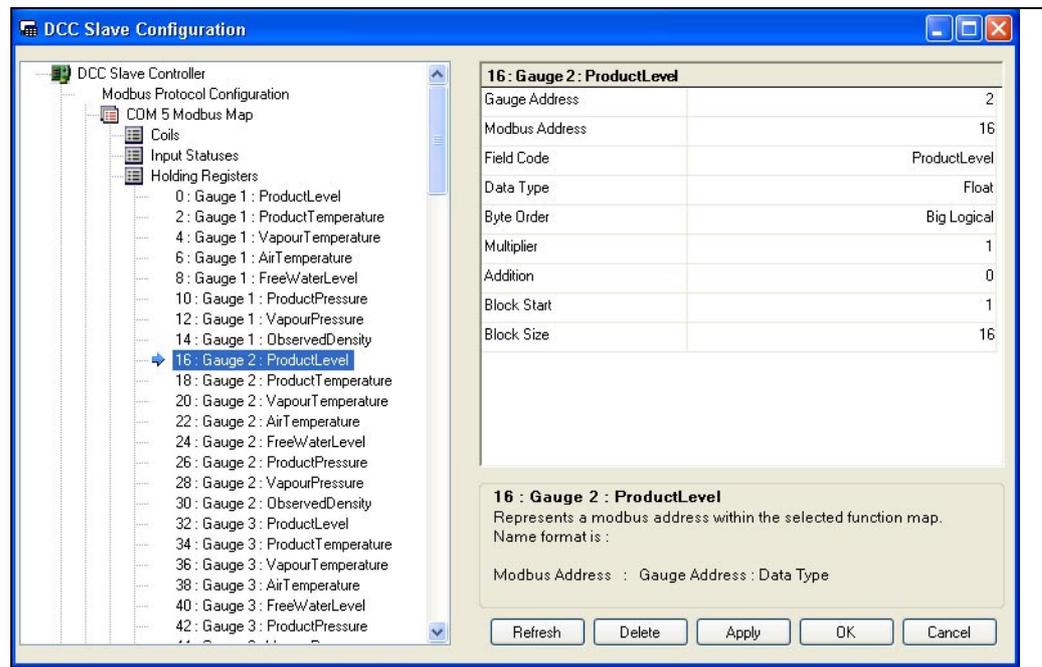
NXA83\_DCC-Slave-Configuration\_Construction-Clipboard\_Example\_Add

Selecting the **Paste From Clipboard** option displays the settings dialog.



NXA83\_DCC-Slave-Configuration\_Construction-Clipboard\_Example\_Paste

After entering the required data, in this example a **Base Modbus Address** of 16 and using gauge 2, then clicking on **OK** will add the registers as shown below:



All eight registers have been added, with the correct Modbus addresses starting from 16 and with a source Gauge address set to 2.

### 7.6.8 Making Changes

The configuration of the Modbus map can be modified at any time using the **DCCSlaveConfigGUI** tool.

The changes only take effect after the **OK** or **Apply** button has been pressed in the tool and the DCC Slave service stopped and re-started.

Clearly any changes made to the Modbus Map need to be made at the host end as well.

### 7.6.9 Field Codes

A field code must be defined for every address within the Modbus map, which determines what data item within a gauge is used as the source for the address. There are a large number of selections available and some knowledge of the database structure is required to determine which fields are required in the Modbus map.

This section will give an outline of some of the standard fields for such things as live data parameters returned from the gauge, calculated volumes, profiles and gauge commands.

#### Live Data Parameters

These parameters are the raw data returned from the gauge or may be manually entered.

Each parameter will have three field codes:

- Data Value: the actual value of the data
- Mode: 0 = data is live from gauge, 1 = data is manual entered
- Status: -1 = data is valid, 0 to 255 = Diagnostic code for invalid data

The name of the three fields will be **Param**, **ParamMode** and **ParamStatus**, where **Param** is the name of the parameter.

For example, the ProductLevel parameter has three field codes available, named ProductLevel, ProductLevelMode and ProductLevelStatus.

The full set of parameters available are shown below, but gauges usually supply only a subset of these. It is up to the configurator to determine which are required.

- ProductLevel
- ProductTemperature
- DisplacerPosition (no mode available)
- FreeWater
- ReferenceDensity
- ObservedDensity
- ObservedTemperature
- ProductPressure
- AirPressure
- VapourPressure
- VapourTemperature
- AirTemperature
- LngCurrentDensity
- OilDepth

### Watchdog

As part of the live parameters section there is a field called Watchdog. This is a special field which will be incremented every time it is read, indicating that the DCC communications sub-system is healthy.

If the DCC sub-system is not healthy (i.e. DCC Host or Hermes services fail) then this field will stop incrementing, in which case ALL data returned on the Modbus link should be regarded as suspect.

The watchdog field has a range 0 to 65 535 (i.e. two bytes, unsigned) and will wrap around to zero. When using this field in a Modbus link, the actual value is not important, as long as it is different to the last value read.

### Time Synchronisation

As part of the live parameters section there is a field that can be used to synchronise the time between the two systems connected by the Modbus interface.

The field is called either **Synchronise Time Command** or **TimeSync**, depending on the software version. As it is a global field, it only needs to be added once to the Modbus map (the tank assigned to it is irrelevant).

When this field has a value written to it (it does not matter what value), the time on the Modbus slave system will be set to the time configured in the modbus setting:

**Global Parameters → Synchronisation Time.**

Thus the field should only ever be written to once per day at this time.

The field can be configured as a coil or a register in the Modbus map.

### Live Status Parameters

These parameters are the raw status data returned from the gauge, they include the field codes:

- AlarmStatus
- GaugeStatus
- ServoStatus
- SystemStatus
- ExtendedAlarmStatus

- SystemGaugeStatus
- ServoAlarmStatus
- FCULevelStatus
- FCUTemperatureStatus
- FCUDiagnostic
- ScanMode
- Stratification Status
- Stratification Severity

They are all bit mapped values with the following meanings:

bit	Alarm Status Description
0	Level Hi Hi Alarm
1	Level Hi Alarm
2	Level Low Alarm
3	Level Low Low Alarm
4	Roof Alarm (Software)
5	Float Alarm (Software)
6	Difference Alarm (Software)
7	Unauthorised Movement (Software)
8	Level Hi Hi Alarm (Software)
9	Level Hi Alarm (Software)
10	Level Low Alarm (Software)
11	Level Low Low Alarm (Software)
12	Hi Temperature Alarm (Software)
13	Lo Temperature Alarm (Software)
14	Hi Flow Alarm (Software)
15	Lo Flow Alarm (Software)

bit	Gauge Status Description
0	Config Command Data Ready
1	ConfigCommand Data Executing
2	Gauge Command Data Ready
3	Gauge Command Executing
4	Gauge Initialising
5	Fast Scan
6	Manual Scan Complete
7	Offline/Offscan

bit	Servo Status Description
0	Servo up
1	Servo down
2	Servo Top Limit
3	Servo Bottom Limit
4	Off-Level
5	Water/Interface Mode

bit	Servo Status Description
6	Testing/Stowing/Calibrating
7	Blocked/Frozen

bit	System Status Description
0	Servo Check Passed
1	Servo Check Failed
2	Config Command Failed
3	Gauge Command Failed
4	Gauge Fault
5	-
6	-
7	General Fault

bit	Extended Alarm Status Description
0	Low Density (Gauge)
1	High Density (Gauge)
2	Low Temperature (Gauge)
3	High Temperature (Gauge)
4	High Density (Software)
5	Low Density (Software)
6	Density Deviation (Software)
7	Temperature Deviation (Software)
8	High Water (Software)
9	Low Water (Software)
10	Density Deviation (Gauge)
11	Temperature Deviation (Gauge)
12	Time To Variable (Software)
13	No Reply from gauge (Software)
14	Theft Detected (Software)
15	LNG Rollover (Software)

bit	Servo Alarm Status Description
0	Level Hi Hi Alarm
1	Level Hi Alarm
2	Level Low Alarm
3	Level Low Low Alarm
4	Product Level No Response
5	Product Level Data Not Valid
6	Product Temperature No Response
7	Product Temperature Data Not Valid
8	Servo up

bit	Servo Alarm Status Description
9	Servo down
10	Servo Top Limit
11	Servo Bottom Limit
12	Off-Level
13	Water/Interface Mode
14	Testing/Stowing/Calibrating
15	Blocked/Frozen

bit	System Gauge Status Description
0	Config Command Data Ready
1	ConfigCommand Data Executing
2	Gauge Command Data Ready
3	Gauge Command Executing
4	Gauge Initialising
5	Fast Scan
6	Manual Scan Complete
7	Offline/Offscan
8	Servo Check Passed
9	Servo Check Failed
10	Config Command Failed
11	Gauge Command Failed
12	Gauge Fault
13	-
14	-
15	General Fault

bit	FCU Level Status Description
0	-
1	-
2	-
3	-
4	-
5	-
6	-
7	Product Level Valid

bit	FCU Temperature Status Description
0	Element 1 Invalid
1	Element 2 Invalid
2	Element 3 Invalid
3	Element 4 Invalid

bit	FCU Temperature Status Description
4	Element 5 Invalid
5	Element 6 Invalid
6	Element 7 Invalid
7	Element 8 Invalid
8	Element 9 Invalid
9	Element 10 Invalid
10	Element 11 Invalid
11	Element 12 Invalid
12	Element 13 Invalid
13	Element 14 Invalid
14	–
15	Product Temperature Valid

bit	Scan Mode Description
0	–
1	Fast Scan Mode
2	–
3	–
4	–
5	–
6	–
7	Offscan mode
8	Connected
9	Connecting
10	Disconnecting
11	Not Connected
12	–
13	–
14	–
15	–

bit	FCU Diagnostic Description
0	–
1	–
2	–
3	Watchdog Tripped
4	–
5	–
6	–
7	–
8	–
9	–

bit	FCU Diagnostic Description
10	-
11	-
12	Passive System is Active
13	-
14	-
15	-

In addition to the standard status data there are a number of status fields that have no fixed mapping, but rather have a "free form" bit allocation dependent on the type of gauge being polled gauge.

These codes include:

- GaugeMode
- ExtendedGaugeStatus

At present most gauges do not implement these fields. Those that do are defined below.

- Scientific Instruments 6280 and 6290 LTD gauges

bit	Extended Gauge Status Description
0	Bottom Reference
1	Lower Sensor
2	Upper Sensor
3	Interlock
4	Master Enable
5	Metric, 1 = metric, 0 = Imperial. Note 6290 only
6	Drive Disable. Note 6290 only
7	Disable Reel Alarm
8	Reel Alarm
9	Probe Uncalibrated
10	System Change Enabled
11	Interval Timer
12	Probe At Liquid Level
13	Unit Not Responding
14	Bit test/Alarm bit test mode. Note 6290 only
15	Simulator mode. Note 6290 only

bit	Gauge Mode Description
0	Manual Mode
1	Calibrate Mode
2	Auto Mode (Follow Level)
3	Program Mode (Profile)
4	Drive Up
5	Stopped

bit	Gauge Mode Description
6	Drive Down
7	Topscan Mode (Profile)

■ ITG 60/70 gauges

bit	Extended Gauge Status Description
0	Density Sensor
1	Temperature Sensor
2	BSW Sensor
3	Datum Sensor
4	–
5	–
6	–
7	–
8	1 Minute Warning
9	Configuration Warning
10	Liquid State, 1 = Homogeneous, 0 = Stratified
11	Liquid State unknown
12	Intelligent Sensing Head Fitted
13	Novram Error
14	–
15	–

bit	Gauge Mode Description
0	Seeking Level
1	Doing Profile
2	Doing Dip
3	Finding BSW
4	Following BSW
5	Finding Datum
6	Following Level
7	–

**Stratification Status parameters**

These parameters indicate the status of the product layering within the tank, and are calculated after a profile has been carried out. They are only relevant if the gauge has the ability to carry out density and temperature profiles.

Stratification Status:

- 0 = unknown
- 1 = Homogenous
- 2 = Stratified

Stratification Severity:

- NOT SUPPORTED AT THE MOMENT

### Calculated Volume/Mass Parameters

These parameters are calculated from the raw data parameters returned from the gauge. They will be set to -1 if not valid, otherwise they will contain an analogue value and are thus best set as float or double data types.

- DensityCorrectionFactor
- FreeWaterVolume
- GaugeVolume
- GrossMass
- GrossObservedVolume
- GrossStandardVolume
- GrossWeight
- NetMAss
- NetObservedVolume
- NetStandardVolume
- NetWeight
- StandardUllageVolume
- StandardUsableVolume
- TotalCalculatedVolume
- TotalGrossStandardVolume
- TotalObservedVolume
- UllageLevel
- UllageMass
- UllageWeight
- UllageVolume
- UsableMass
- UsableWeight
- UsableVolume
- VapourMass
- VapourStandardVolume
- VCF

### Profile Data Parameters

Some gauges can carry out profiles of the temperature and density in a tank. These are saved in the database as a set of tables for level, temperature and density. The profile data for the last profile command is available via the Modbus slave and may be added to the Modbus map in user defined blocks.

The field codes are:

- DensityProfileBlock1
- DensityProfileStatusBlock1
- LevelProfileBlock1
- LevelProfileStatusBlock1
- TempProfileBlock1
- TempProfileStatusBlock1

Each block requires certain parameters to be defined:

- Block Start: The index within the profile table (1 to 256 range) for the first position in the block
- Block Size: The number of profile points after the start position to be included in the block, up to a maximum of 256

Thus if the gauge supports 10 profile points (i.e. an Enraf 854 gauge) then the Block Start should be set to 1 and the Block Size to 10.

The status blocks are coded as per Live Data status i.e. -1 = valid, 0 to 255 = invalid.

When adding profile blocks to the map it must be taken into account that multiple ADDRESSES will actually be added depending on the Block Size field. i.e. if

LevelProfileBlock1 is added at address 0, with a Block Size of 10 and its data type is float then the next free address is 20 (float = 2 addresses · 10 profile points in block).

Profile blocks may be added in the normal way, being assigned for each gauge in the database, but this may take up a very large amount of addresses. Thus there is a second method of using the profiles in the Modbus map, using the **ProfileSelector** field.

To enable this mode add the **ProfileSelector** field code to the map ONCE only, the gauge address does not matter. Then the Profile blocks should be added to the map ONCE only, again which gauge address is selected for the map does not matter.

The profile blocks get their data from the gauge whose address is currently stored in the ProfileSelector point. Thus to read the profile data for a gauge, the master should first write the gauges address to the ProfileSelector point, then read the profile blocks.

### Skin Temperature Data

Some gauges can return a set of skin temperatures from around the surface of the tank, up to maximum of 250 individual points. These are updated in DCCHost on the background scan cycle, defaulting to every three minutes.

The skin temperature data is available via the Modbus slave and may be added to the Modbus map in user defined blocks.

- SkinTempBlock1
- SkinTempStatusBlock1

Each block requires certain parameters to be defined:

- Block Start: The index within the skin temperature table (1 to 256 range) for the first position in the block
- Block Size: The number of temperature points after the start position to be included in the block, up to a maximum of 256

The status blocks are coded as per Live Data status i.e. 1 = valid, 0 to 255 = invalid.

Skin Temperature blocks are treated in the same way as Profile blocks in the Modbus map. Therefore refer to the section on Profile Data for more details on the formatting of blocks. This also includes the **ProfileSelector** method. If this is used then it also applies to Skin Temperature blocks.

### Multi-Element Temperature Parameters

Gauges that have multi-element or multi-spot temperature transmitters fitted can return a table of up to 16 temperatures. This data is available via the Modbus slave and may be added to the Modbus map in a single block of sixteen addresses.

The field codes are:

- TempElementBlock
- TempElementStatusBlock

The TempElementBlock contains the 16 temperature points in the table, with the status block having the 16 corresponding status fields coded as per Live Data status i.e. -1 = valid, 0 to 255 = invalid.

When adding a multi-element temperature block to the map it must be taken into account that 16 ADDRESSES will actually be added i.e. if TempElementBlock is added at address 0 and its data type is float then the next free address is 32 (float = 2 addresses · 16 profile points in block.)

### Gauge Commands

There are a number of fields used to issue commands to control a gauge and are currently the only fields that may be written to as well as read.

The fields are:

- GaugeCommand
- Arg0
- Arg1
- Arg2
- Arg3

A command will be sent to the gauge when the **GaugeCommand** field is written to.

Args 0 to 3 contain optional arguments required for certain commands and should be set BEFORE writing to the **GaugeCommand** field.

The **Gauge Command** field consists of an ASCII character which may be one of the following:

ASCII character	Meaning
A	Alternate (Fast) Scan
B	Basic (Normal) Scan
O	Single Scan (for Off-Scan gauge)
X	Exclude from Scan/Off-Scan
T	Test Gauge (Servo Check)
C	Cancel Test
F	Freeze
N	Un-freeze
S	Stow
U	Un-Stow
R	Raise
L	Lower
Q	Calibrate
1	Interface 1 (air – product)
2	Interface 2 (tank bottom)
3	Interface 3 (product – water)
M	Store current Level, Temperature and Alarm Status
E	Update Element Temperatures
I	Update Tank/Interface Profile Levels
K	Update Tank/Interface Profile Temperatures
P	Update Tank/Interface Profile Densities
D	Dip mode (level/temperature dip)
V	Profile Scan (temperature/density/water/datum, etc.)
W	Water Dip
Y	Datum Dip (zero level)
Z	Initialise Gauge (Initialise Gauge Comms – does not reset the gauge itself)
-	Cancel current command
(sp)	No command active (status indication - not actually sent as a command)

The Argument fields 0 to 3 may be required for the following commands:

S → Stow/Lock Test Gauge	Meaning
Arg0	0 = Stow Gauge to Top Limit Cut-out
	1 = Lock Test to Level
	2 = Stow Gauge to Top Limit Cut-out then return to Product Level
	3 = Lock Test to Level then return to Product Level
Arg1 → Lock Test Level (mm)	0 = Lock Test to Top Limit Cut-out

T → Test Gauge (Servo Check)	
Arg0 → Servo Check/Test Distance (mm)	0 = use default 300 mm
Arg1 → Servo Check/Test Tolerance (mm)	0 = use default ± 5 mm
Arg2 → Servo Check/Test Timeout (seconds)	0 = use default 90 seconds

V → Profile Scan (Temperature/Density/Water)	
Arg0 → bit 0	0 = Tank Profile
	1 = Interface Profile
Arg0 → bit 1	0 = Downwards Scan
	1 = Upwards Scan
Arg0 → bit 2	0 = Exclude Water Level Dip
	1 = Include Water Level Dip
Arg0 → bit 3	0 = Exclude Datum Level Dip
	1 = Include Datum Level Dip
Arg0 → bit 4	0 = Include Temperature Profile Scan
	1 = Exclude Temperature Profile Scan
Arg0 → bit 5	0 = Include Density Profile Scan
	1 = Exclude Density Profile Scan
Arg0 → bit 6	not used: set to 0
Arg0 → bit 7	0 = Profile Scan Limit Levels Use Absolute Positions
	1 = Profile Scan Limit Levels Use Relative Positions
Arg1 → Profile Lower Limit Level (mm)	0 = use value pre-configured in gauge
Arg2 → Profile Upper Limit Level (mm)	0 = use value pre-configured in gauge
Arg3	≥ 1 and ≤64 → Number of Profile Points
	≥ 65 Profile Interval (mm)
	0 = usevalue pre-configured in gauge

W → Water Dip	
Arg0	Water Interface Level Offset

### Alarm Configuration Parameters

Programmable alarm configuration parameters may be accessed via the Modbus interface.

These are listed below, and all may be written to (using Modbus tasks 5 and 16) as well as read:

- FlowAlarmEnabled
- FlowHigh
- FlowLow
- FlowHysteresis
  
- LeakAlarmEnabled
- Leak Threshold
  
- DifferenceAlarmEnabled
- DifferenceAllowableTolerance
- DifferenceHysteresis
  
- LevelAlarmEnabled
- LevelHiHi
- LevelHi
- LevelLo
- LevelLoLo
- LevelHysteresis
  
- TemperatureAlarmEnabled
- TemperatureHigh
- TemperatureLowTemperatureHysteresis
  
- FloatAlarmEnabled
- FloatLevel
- FloatHysteresis
- FloatDirection
  
- RoofAlarmEnabled
- RoofEarlyWarning
- RoofHysteresis
- FloatDirection
  
- DensityAlarmEnabled
- DensityHigh
- DensityLow
- DensityHysteresis
  
- WaterAlarmEnabled
- WaterHigh
- WaterLow
- WaterHysteresis
  
- ProfileDeviationAlarmEnabled
- TemperatureProfileDeviation
- DensityProfileDeviation

All the Fields containing the "Enabled" key word are coded as follows:

- 0 = Alarm Disabled
- 1 = Alarm Enabled

All other fields are setpoint values, and will use the default units for the type of alarm, i.e. Level setpoints are in mm, density setpoints in kg/l, etc.

### 7.6.10 Byte Ordering

Within a data item which has a length of more than one byte, the bytes which make up that data item may be placed into registers in various sequences for transfer over the Modbus network. Such detail is not specified in the Modbus Protocol Reference Guide. The Modbus drivers provide a selection of different byte ordering methods so as to cater for Modbus devices which store data internally in different ways. Always consult the documentation of the Modbus device in order to determine which byte ordering the device employs.

In the following description, the "first" register is that with the numerically lowest address used for the data item and the "last" register is that with the numerically highest address. When registers are transferred over the Modbus network, they are sent in ascending numerical order within the messages.

In the examples of the encoding of a 32 bit number. **A** represents the address of the first register used to hold the data item value and **A+1** represents the next, i.e. last register so used. The bytes of the data item value are represented by MSB (Most Significant Byte), NMSB (Next Most Significant Byte), NLSB (Next Least Significant Byte) and LSB (Least Significant Byte). The Most Significant Byte of a data value is that byte which contains the part of the number which would be leftmost if the number were in written form (e.g. in the number 1234, the "1" is the most significant digit).

The byte ordering within a data item may be set to one of the following:

#### Little Endian

- Least significant byte of data value in low byte of first register
- Most significant byte of data value in high byte of last register

Example:

Register	Byte	Content
Register A	high byte	contains data value NLSB
Register A	low byte	contains data value LSB
Register A+1	high byte	contains data value MSB
Register A+1	low byte	contains data value NMSB

#### Big Endian

- Most significant byte of data value in low byte of first register
- Least significant byte of data value in high byte of last register

Example:

Register	Byte	Content
Register A	high byte	contains data value NMSB
Register A	low byte	contains data value MSB
Register A+1	high byte	contains data value LSB
Register A+1	low byte	contains data value NLSMB

#### Little Logical

- Least significant byte of data value in high byte of first register
- Most significant byte of data value in low byte of last register

Example:

Register	Byte	Content
Register A	high byte	contains data value LSB
Register A	low byte	contains data value NLSB
Register A+1	high byte	contains data value NMSB
Register A+1	low byte	contains data value MSB

### Big Logical

- Most significant byte of data value in high byte of first register
- Least significant byte of data value in low byte of last register

Example:

Register	Byte	Content
Register A	high byte	contains data value MSB
Register A	low byte	contains data value NMSB
Register A+1	high byte	contains data value NLSB
Register A+1	low byte	contains data value LSB

The encoding of Time/Date values conforms to ODBC standards, where the time and date is represented by an eight byte floating point number. The whole number part of this represents the number of days since 1st January 1900. The fractional part represents the time as a fraction of 24 hours.

Floating point numbers are encoded in accordance with the IEEE754 specification. It should however be noted that there are numerous other methods of encoding floating point numbers but these are relatively uncommon. Some of these alternative methods are significantly different from the IEEE754 specification such that values so encoded would be immediately obvious as being wrong. There are however, a small number of encoding methods which are similar to IEEE754 and which may only manifest erroneous values in subtle ways. Always consult the documentation for the Modbus device to verify the encoding method used.

### 7.6.11 Data Types

The following data type identifiers are currently supported in the Modbus interface:

Type	Range	Number of Register used	Notes
Bit	0-1	1 Coil	Should only be used for Coils or Input Status maps
Signed Char	-127 to +127	1	-
Unsigned Char	0 to 255	1	-
Short	-32768 to 32767	1	-
Unsigned Short	0 to 65535	1	-
Int	-2147483648 to 2147483647	2	-
Unsigned Int	0 to 4294967295	2	-
Float	3.4E ± 38 (7 digits)	2	-
Double	1.7E ± 308 (15 digits)	4	-
Text	12 ASCII characters	6	Each register is packed with 2 ASCII characters

Type	Range	Number of Register used	Notes
Fractional	0.0000 – 65535.9999	2	Data is converted to 4 byte value. HiWord (top 2 bytes) is integer value = 0-65535 LoWord (lower 2 bytes) is decimal places = 0-9999 Number of decimal places sent is defined in the Block Size field, up to a maximum of 4 places (default is 3) <b>For Example:</b> If using 4dp then a Volume value of 40052.7148 m3 will be converted to Hi Word = 40052 Lo Word = 7148 The bytes sent (in hex notation) on the Modbus interface will depend on the Byte Order selected: Little Endian = 9C 74 1B EC Big Endian = EC 1B 74 9C Little Logical = 74 9C EC 1B Big Logical = 1B EC 9C 74

## 7.7 Enraf Protocol Configuration

DCC Slave implements the Enraf Host Communications protocol as defined in the Enraf document "Host Communications Manual ENTIS+, Version 5.23x" (file name HO-523x).

This defines an ASCII protocol for communication between a Master system and the slave interface running on the tank gauging system.

The following sections offer a quick getting started guide to configuring interfaces in the DCC Slave that use the Enraf Host protocol. They assume a degree of prior knowledge of the protocol and it is recommended that a copy of the definition document is available when configuring the interface.

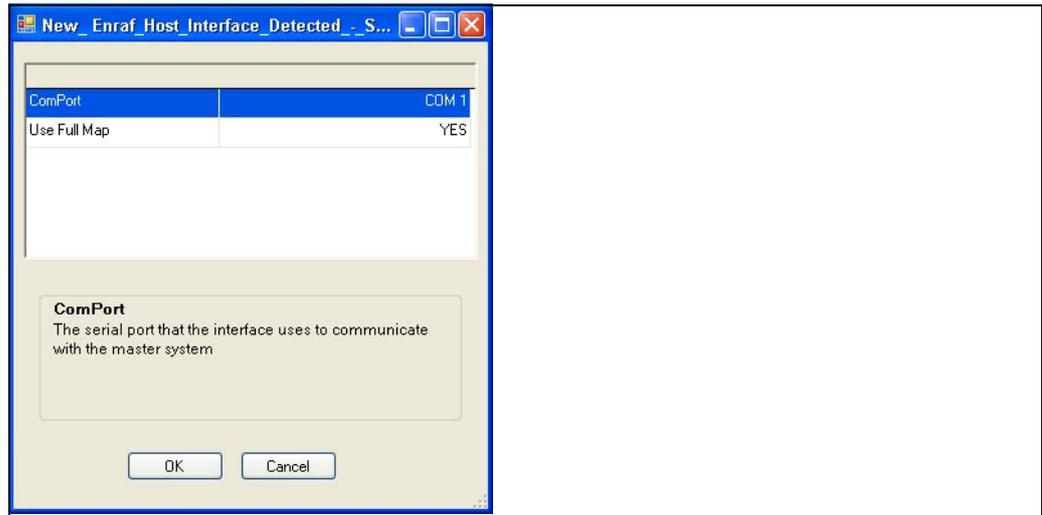
DCC Slave supports a sub-set of the full protocol, this includes:

- Full support of all Communication Parameter configuration options
- Tank Data configuration formats for versions 3b , 4.0q and 5.2.1
- All Tank data items supported, except Inlet and Outlet valve data
- Packet types supported:
  - Packet type 1 (Entity Request)
  - Packet type 3 (Download Command)
  - Packet type 7 (Entity Command)

### 7.7.1 Configuring Communication and Data Format

Start the DCCSlaveConfigGUI.exe module.

If one or more Enraf Hosts have been newly added to the DCC Slave and currently have no configuration details then the following screen will automatically be displayed, prompting the user to select the map type.

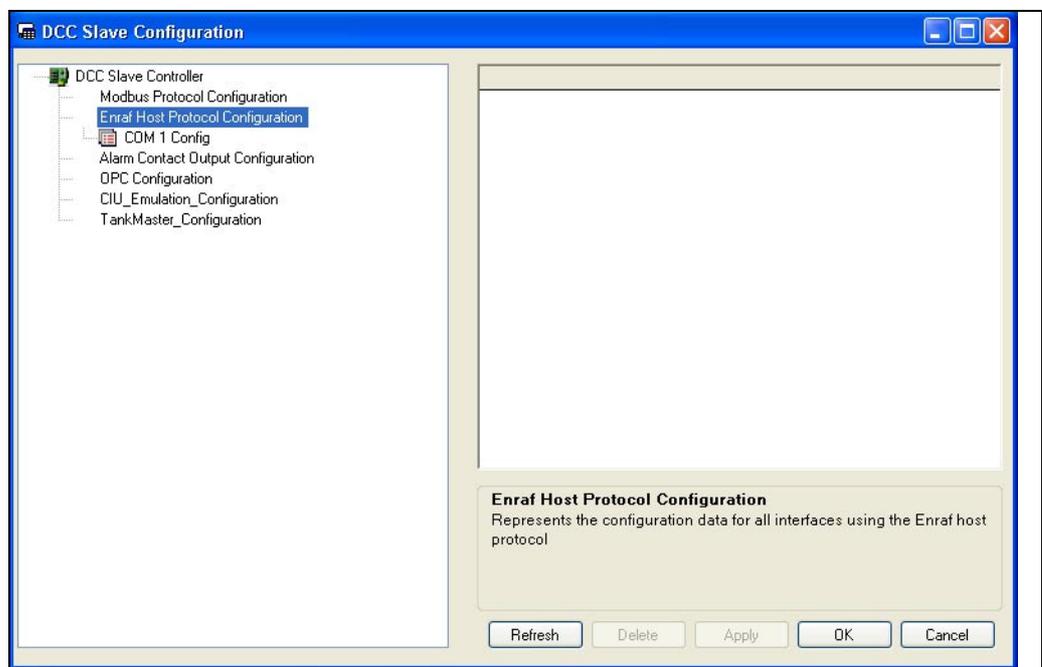


NXA83\_DCCSlaveConfigGUI\_Select-Map-Type

The COM port is fixed and cannot be changed but the **Use Full Map** fields may be set to **YES** or **NO**.

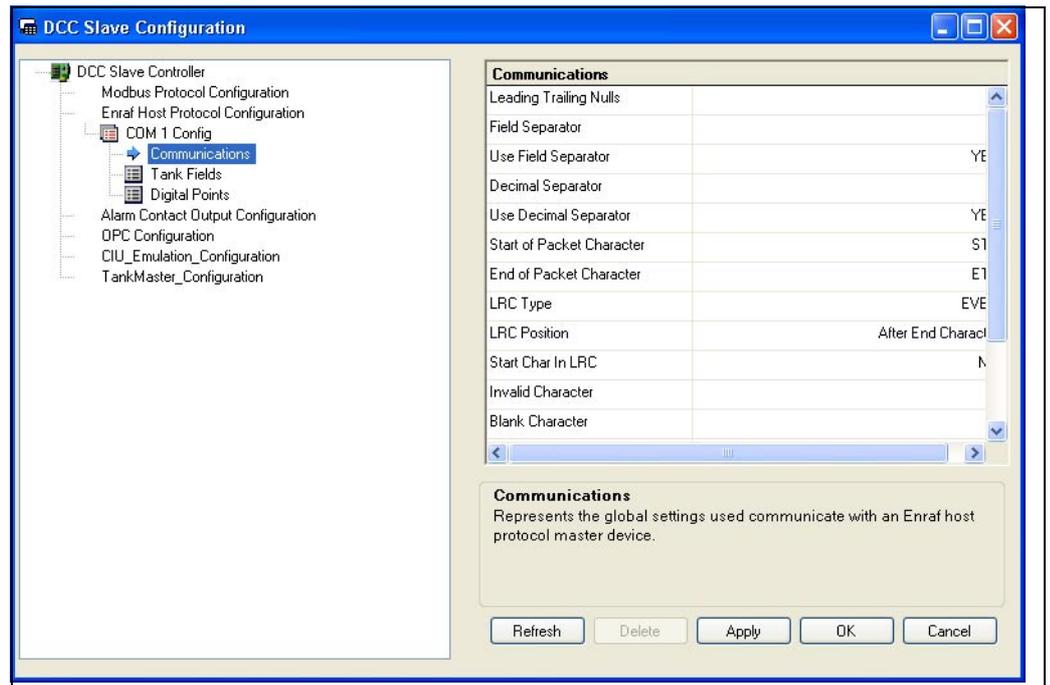
If set to **YES** then the tank configuration map will be initialized to that defined for version 5.2.1 of the protocol, if **NO** then to version 3b/4.0q (see section 3.3 of the Enraf definition document HO-523x).

Once all new Enraf interfaces have been initialized then the main configuration screen will be displayed. Expand the **Enraf Host Protocol Configuration** item to display all COM ports currently assigned to use this protocol.



NXA83\_DCCSlaveConfigGUI\_Enraf-Host-Protocol-Configuration

To edit the configuration of an interface, click on the COM port item for the interface and select the **Communications** item.



NXA83\_DCCSlaveConfigGUI\_Communications

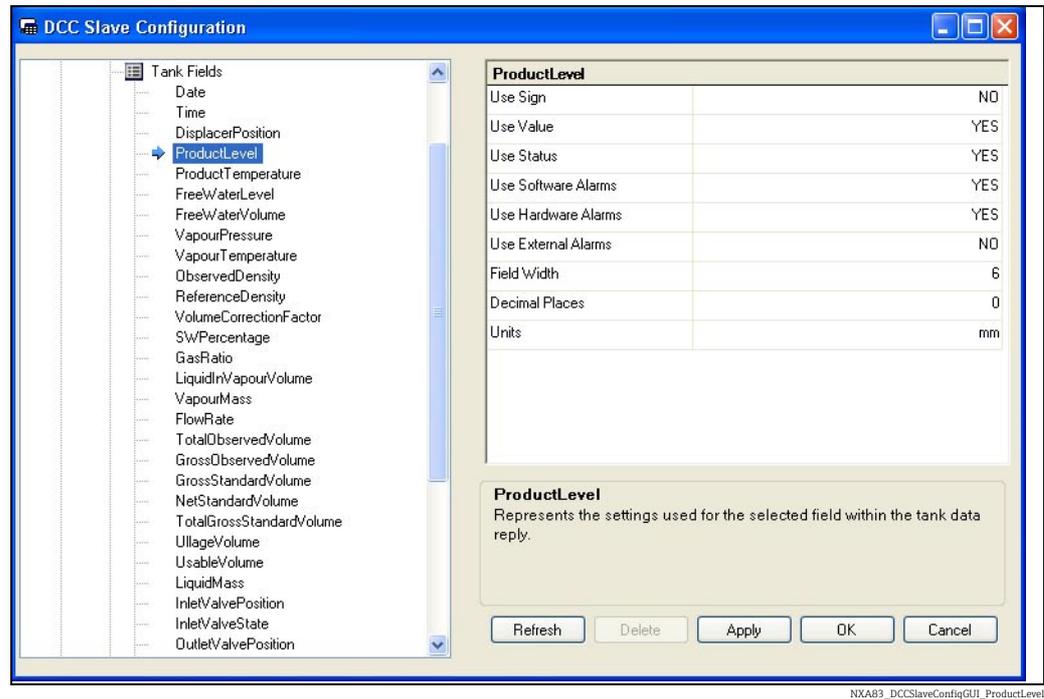
This allows all the Enraf specific communications fields to be edited in the right hand edit panel.

- **Leading Training Nulls:** Number of null characters to be sent before and after the reply message
- **Field Separator:** ASCII character to send in between each tank data field in the reply message
- **Use Field Separator:** Enables or disables the field separator
- **Decimal Separator:** ASCII character to use a decimal point in data fields that have decimal places
- **Use Decimal Separator:** Enables or disables the decimal separator
- **Start of Packet Character:** ASCII character that signifies a start of a packet
- **End of Packet Character:** ASCII character that signifies a end of a packet
- **LRC Type:** Defines the type of Longitudinal Redundancy Check for error detection
- **LRC Position:** The LRC character can be located before or after the end character in the packet
- **Start Char in LRC:** Include the packet start character in the LRC calculations
- **Invalid Character:** Any data field that undefined or invalid such as when a tank entity requested is not found in the database
- **Blank Character:** All data fields are right-justified and the remaining spaces are filled with the blank character
- **Profile Not Avail Field:** Not Supported at the moment
- **Profile Data Separator:** Not Supported at the moment
- **Name Length:** The character length of the entity names in a transmitted packet i.e. Tank Names in the system. Names are left-justified.
- **Mass in Air:** If set to yes then all mass fields in the interface will be refer to weights (i.e. mass in air), otherwise mass will be used (i.e. mass in Vacuum)

For a fuller description of these fields see section 3.2 of the Enraf definition document HO-523x.

After configuring the communications parameters, the tank data format has to be defined. This is displayed as a list of all possible data that can be returned per tank and can be edited by clicking on the **Tank Fields** item.

For each data field there is a number of formatting parameters that can be configured by clicking on the field and editing the data in the left hand panel.



The order of the data fields in the list is the order that they are sent in a tank reply. Each data field can have optional elements enabled:

- Sign
- Value
- Status
- Software Alarms
- Hardware Alarms
- External Alarms

If all elements in a data field are set to NO (not set to be used), no data from that field is transmitted.

Each data field also has three parameters that define the format of the data Value if it is enabled:

- Field Width
- Decimal Places
- Units

All other elements of the field (Sign, Status, Sw, Hw, Ex) are one character in length.

For a fuller description of these fields see section 3.3 of the Enraf definition document HO-523x.



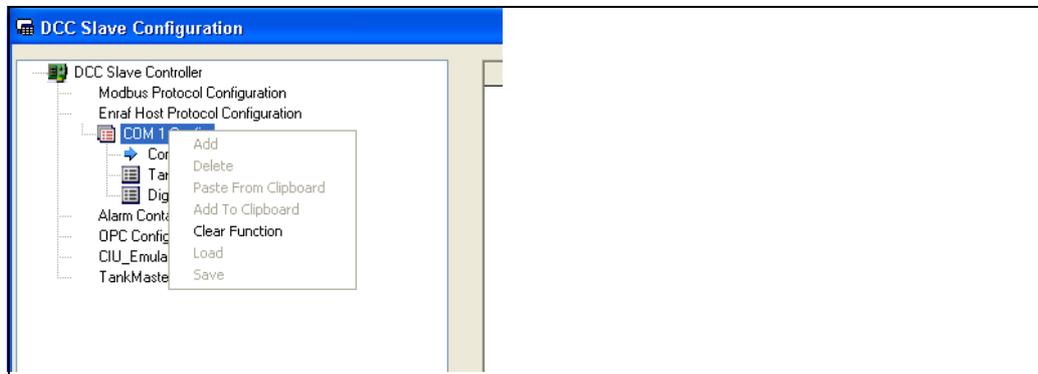
External alarms are not supported and will always return the "-" character if enabled for a field.

Click on the **Apply** button to save any changes to the database.

If DCC Slave is running when any changes are saved, it must be re-started before those changes will be used.

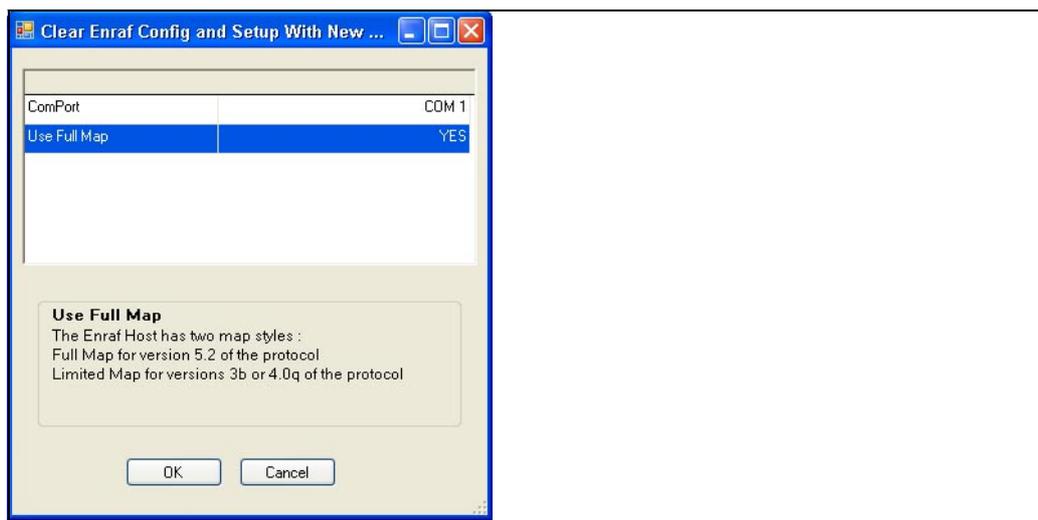
### 7.7.2 Resetting the Configuration

If for any reason the configuration data for an Enraf interface needs to be reset back to defaults then this can be carried out in DCCSlaveConfigGUI by right clicking on the interface in the **Enraf Host Protocol Configuration** section and selecting **Clear Function** from the pop-up menu.



NXA83\_DCCSlaveConfigGUI\_Clear-Function

This will display a dialog where the new map type may be selected.



NXA83\_DCCSlaveConfigGUI\_Clear-Enraf-Config

Select the map type then click **OK**. This will set all communications parameters and Tank Fields back to the standard default values.

As always any changes will only become permanent when the **Apply** is clicked.

### 7.7.3 Tank Field Codes

The following table defines the connections between the tank field codes defined in an Enraf Host interface and the database:

Enraf Host Tank Field	Database Field
Date, Time	No connection to database. Generated when reply message created
Product Level	Product Level
Displacer Level)	Displacer Position
Product Temperature	Product Temperature
Water Level	Water Level
Water Volume	Water Volume
Vapour Pressure	Vapour Pressure
Vapour Temperature	Vapour Temperature
Observed Density	Observed Density
Ref Density (or Density at ref)	Reference Density
Volume Correction factor	Volume Correction factor (VCF)
S&W percentage	Not Supported
Gas Ratio	Not Supported
Liquid in Vapour Volume	Vapour (Gross) Standard Volume
Vapour Mass	Vapour (Gross) Standard Volume or Not Supported if MassInAir setting = yes
Flow Rate	Volume Flow rate (m3FlowRate)
Total Observed Volume	Total Observed Volume
Gross Observed Volume	Gross Observed Volume
Gross Standard Volume	Gross Standard Volume
Net Standard Volume	Net Standard Volume
Total Gross Standard Volume	Total Gross Standard Volume
Room Available	Ullage Volume
Available Volume	Usable volume
Liquid Mass	Gross Mass - Vapour (Gross) Standard Volume or Gross weight if MassInAir setting = yes
Inlet Valve Position	Not Supported
Inlet Valve State	Not Supported
Outlet Valve Position	Not Supported
Outlet Valve State	Not Supported
Product Name	Not Supported
Product Code	Volume Correction Control See Note Below
Ref Temperature	Volume Correction Control See Note Below
Thermal expansion Coeff	Not Supported in Interface
Mass calculation	"V" or "A" if MassInAir setting = yes
Total Mass	Gross Mass or Gross Weight if MassInAir setting = yes
Ullage/Innage Level Type	Bit 1 of Inventory Control Settings

Product Code is supported for the following values only:

Value	Product Code
A	API table A
B	API table B
C	API table C
D	API table D
M	manually entered VCF
X	no volume correction, VCF = 1

Ref Temperature will be determined from the selected Volume Correction Method, except for tables with a variable reference temperature which will not be supported.

Only the following characters will be supported for the Status of tank field data items:

Character	Status of Tank Field Data items
-	All fields
&	Levels, Temperatures, Pressures and Densities only
B	Level only
C	Level only
F	All fields
S	Level only
T	Level only
W	Level only

Only the following characters will be supported for the Software Alarm (Sw) of tank field data items:

Character	Software Alarm (Sw) of Tank Field Data items
-	All fields
F	All fields
H	Level only
H	Level, Temperature, Flow rate, Water Level, Ref and Obs Density only
L	Level, Temperature, Flow rate, Water Level, Ref and Obs Density only
L	Level only
N	Level, Temperature, Flow rate, Water Level, Ref and Obs Density only

Only the following characters will be supported for the Hardware Alarm (Hw) of tank field data items:

Character	Hardware Alarm (Hw) of Tank Field Data items
-	All fields
F	All fields
H	Level only
H	Level, Temperature, Ref and Obs Density only

Character	Hardware Alarm (Hw) of Tank Field Data items
L	Level, Temperature, Ref and Obs Density only
L	Level only

The External Alarm (Ex) of tank field data items will not be supported for any fields.

#### 7.7.4 Downloading to Tank Fields

The following tank fields may be written to be the master system using Packet type 3 (Download Command):

- Product Code
- Reference Density\*
- Thermal Expansion Coeff.
- Water Volume\*
- VCF\*
- Product Level\*
- Product Temperature\*
- Vapour Pressure\*
- Vapour Temperature\*
- Product Name
- Water Level\*

Fields MUST be enabled in the map before they can be downloaded to using Packet type 3.

Those fields marked with an asterisk in the above list must be set to manual mode in the database, otherwise the downloaded data will be discarded.

### 7.8 Extended Modbus Configuration

DCC Slave implements an extended version of the Modbus protocol as defined in the Enraf document "Specification Extended Modbus Communication for ENTIS+". It is configured using a special tool called "EMBConfiguration.exe" located in the standard system installation directory.

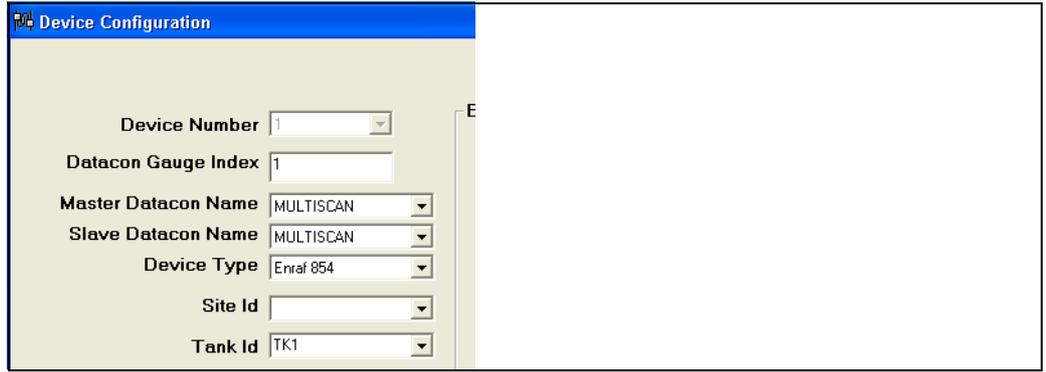
### 7.9 Datacon Slave Configuration

DCC Slave can be configured to respond on one or more serial ports/modems as if it were a slave Datacon device.

There is no configuration required other than:

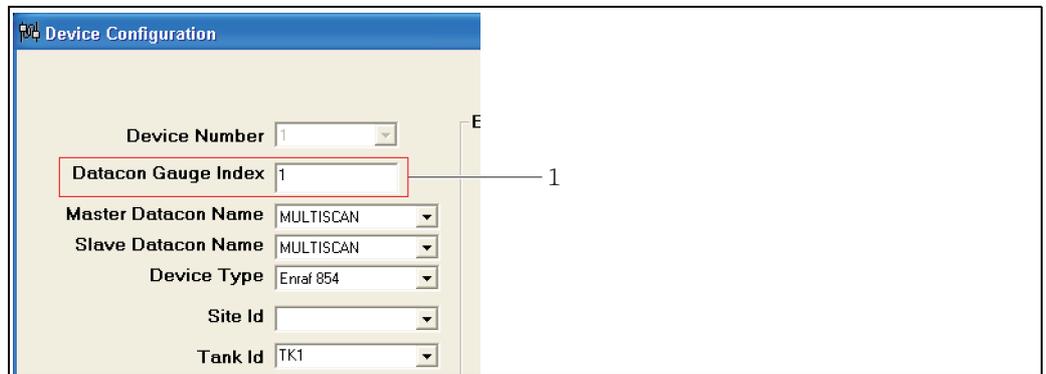
- Select Datacon for the protocol type
- Set the slave address to that requested by the master system.

Gauges are addressed using the **Device Number**, as defined in the **Gauge Configuration** screen.



NXA83\_Device-Configuration\_Device-Number

This option can often be useful if one Tank Gauging system requires tank data from a second remote Tank Gauging system, especially so if there is no network connection between the two systems. The master system would configure gauges with the **Datacon Gauge Index** set to the same value as the **Device Number** in the receiver system acting as a Datacon slave.



NXA83\_Device-Configuration\_Datacon-Gauge-Index

1 Configuring gauges with the Datacon Gauge Index

This option is also the standard method of connecting Tank gauging software running on a PC with a Multi Scan NXA83B.

## 7.10 Relay Contact Output Configuration

DCC Slave sub-system currently supports relay contact outputs using the following types of hardware:

- **External Relay units** connected via a serial port on the PC and using the Modbus protocol to communicate as slave devices. Multiple devices may be connected on the same port if RS485/RS422 hardware is used. Multiple serial ports are also supported.

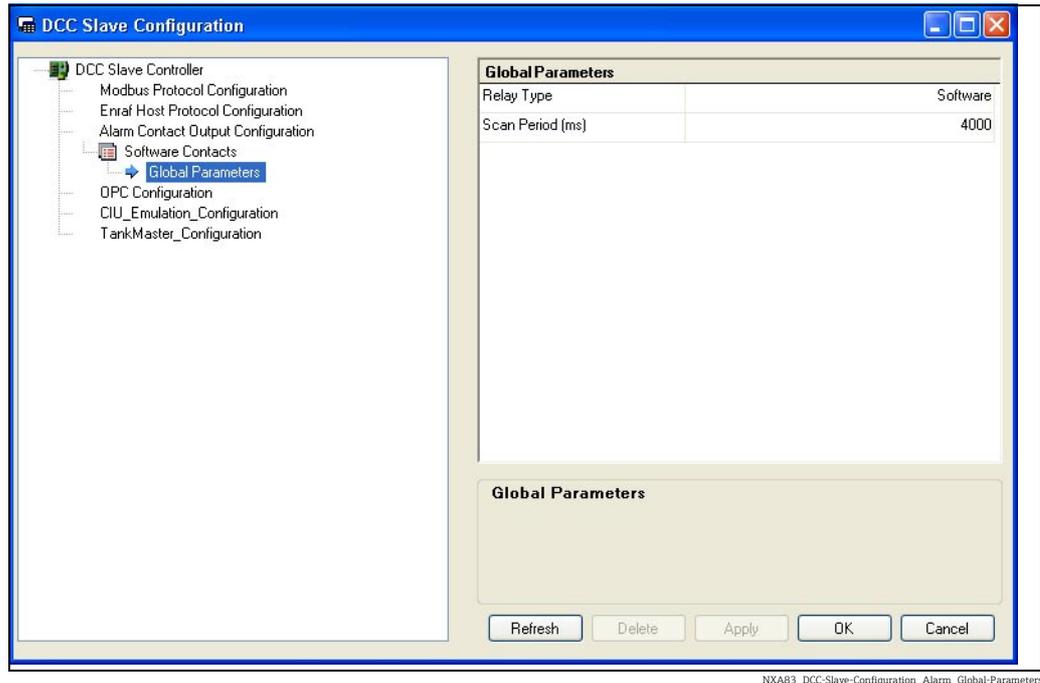
The relays may then be configured to be activated by any gauge alarm, software alarm or certain other tank status signals.



These devices should not be used in safety critical operations.

### 7.10.1 Configuring Relay Boards

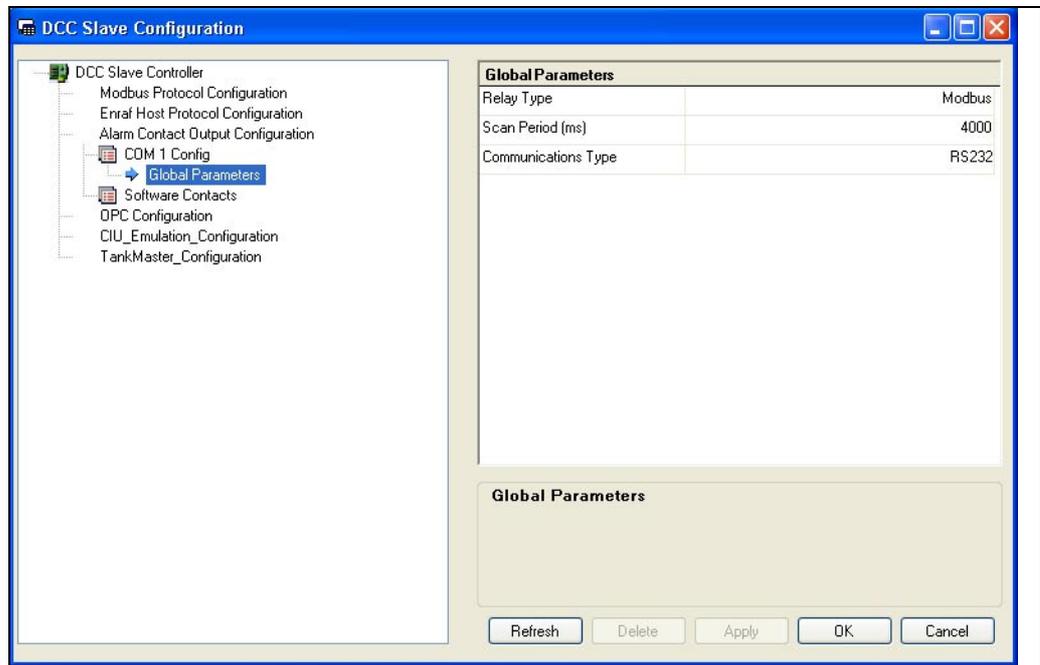
Relay contact outputs may be controlled by any combination of gauge/tank alarm states, and are configured using the **DCCSlaveConfigGUI.exe** utility.



NXA83\_DCC-Slave-Configuration\_Alarm\_Global-Parameters

## 7.10.2 Configuring Modbus Devices

Relay contact outputs may be controlled by any combination of gauge/tank alarm states, and are configured using the **DCCSlaveConfigGUI.exe** utility.



NXA83\_DCC-Slave-Configuration\_Modbus\_Global-Parameters

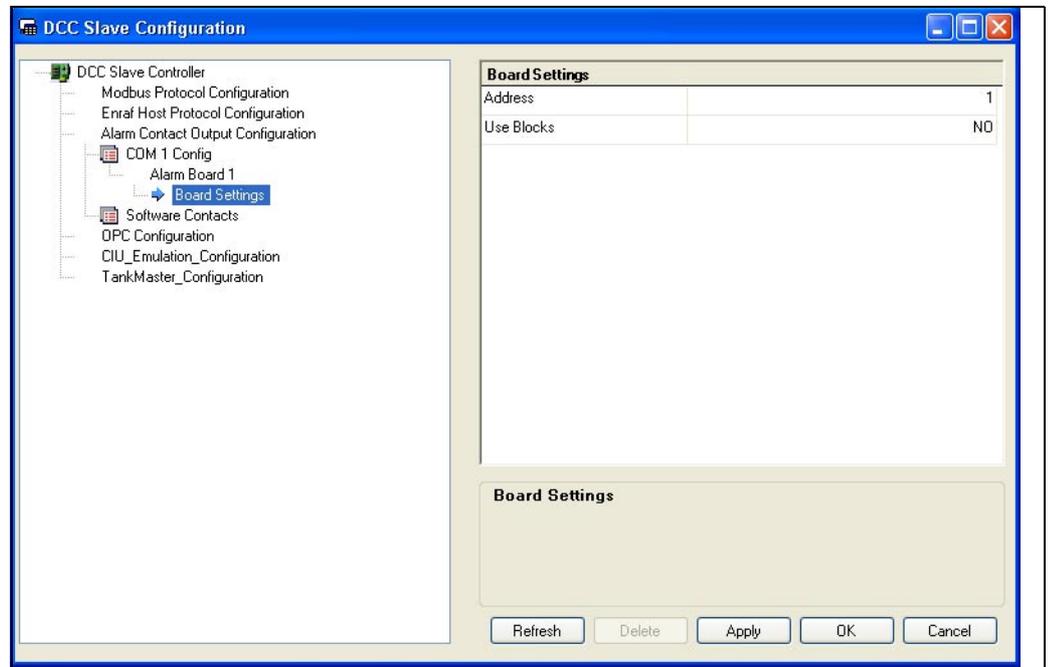
All EXTERNAL Modbus alarm device configuration is contained within the **Alarm Contact Output Configuration** section, and each COM port configured for Modbus Alarm Output devices will have an entry under this section. For example **COM 6 Config**.

Under each COM port config section there are a set of **Global Parameters** that apply to all Modbus devices connected on that port.

- **Relay Type:** The will be fixed as Modbus

- **Scan Period:** The scan time in milliseconds for checking the relay states, defaulting to 4 000 milliseconds
- **Communications Type:** Set to RS485 if an echo of requests is expected, otherwise set to RS232

Alarm Boards must be added to the DCC Slave sub-system before they can be configured. To do this, right click on the required COM port section and select **Add** from the pop-up menu.



NXA83\_DCC-Slave-Configuration\_Modbus\_Board-Settings

Each Alarm Board represents a Modbus device connected on the serial link.

The Board Settings are as follows:

- **Address:** Defines the Modbus RTU address of the device
- **Use Blocks:** Determines if individual Modbus COILS are to be used to control relay outputs OR outputs may be set in blocks using REGISTERS

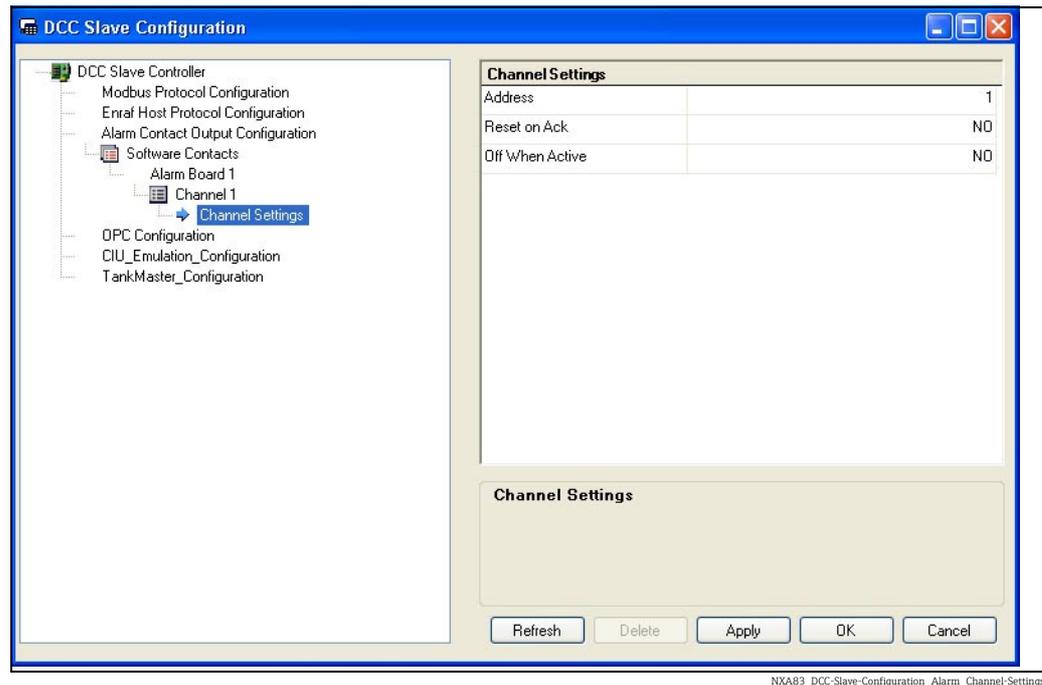
Multiple boards may be added.

Boards may be deleted by right clicking on the board and selecting **Delete** from the pop-up menu.

Each relay on a board is referred to as a Channel and these must be added and configured. If **Use Blocks** is set to NO then each Channel is an individual Modbus coil, if set to YES then a Channel is a block of 16 bits within a Modbus register.

### 7.10.3 Configuring Channels

Each relay contact on a board is defined as a Channel. To add a new channel to a board, right click on the **Alarm Board** where the relay is located and select **Add** from the pop-up menu.



The Channel Settings define the operation of the relay as follows:

- **Address:** For EXTERNAL Modbus devices this is the address of the coil or register.
- **Reset on Ack:**
  - NO = relay is set to inactive state when all source fields (i.e. alarms) are cleared
  - YES = relay is set to inactive state when all source fields (alarms) have been acknowledged, even if the alarm is still active
- **Off When Active:**
  - NO = relay is energised when in active state
  - YES = relay is de-energised when in active state

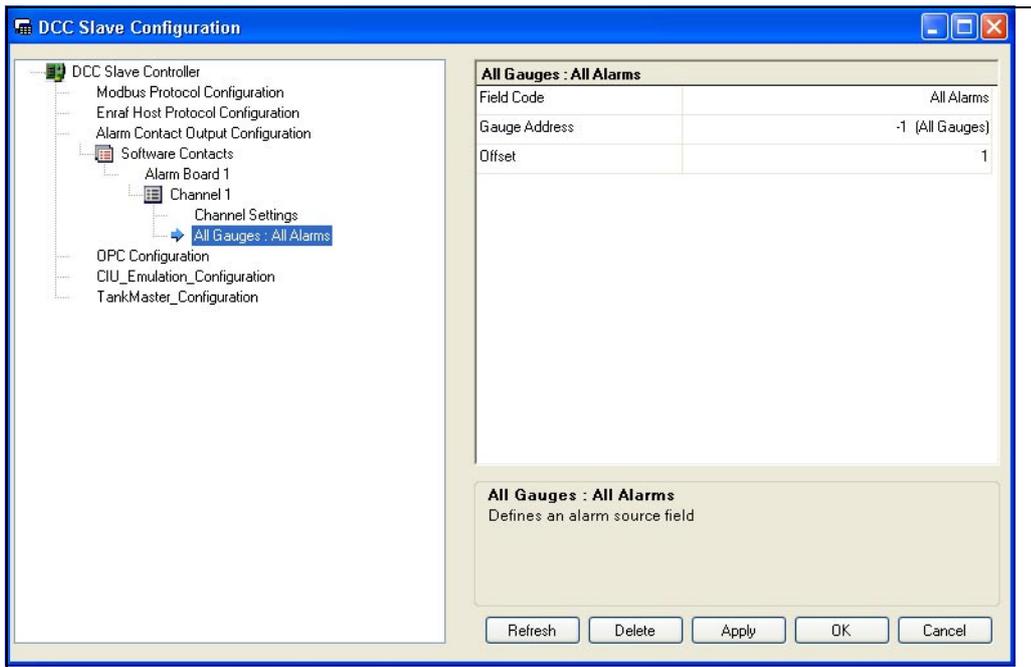
All channels must be configured for the alarms or status values that will trigger/control it. These are referred to as Fields, and each channel may have multiple fields controlling it.

#### 7.10.4 Configuring Fields

Each relay channel must have at least one source Field defined. Various different fields may be assigned to a channel.

Each Field defines an alarm or status point from a tank/gauge that will be used to control the channel.

To add a Field to a channel, right click on the **Channel** and select **Add** from the pop-up menu.



NXA83\_DCC-Slave-Configuration\_Alarm\_All-Gauges

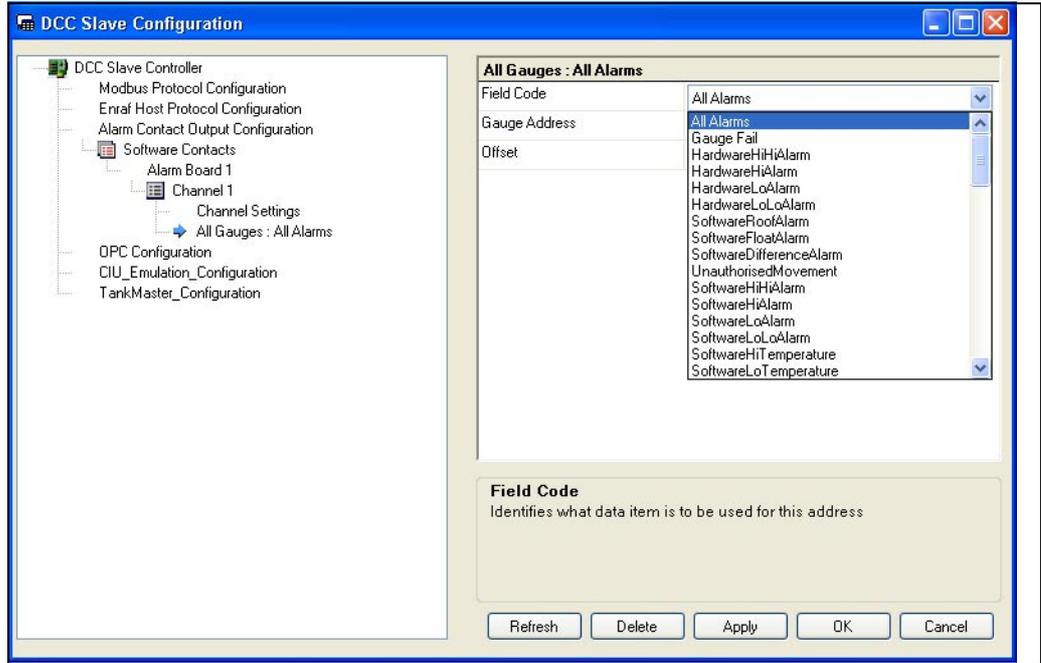
Each Field will be displayed as the source tank, followed by the source alarm.

The parameters for each Field are defined as:

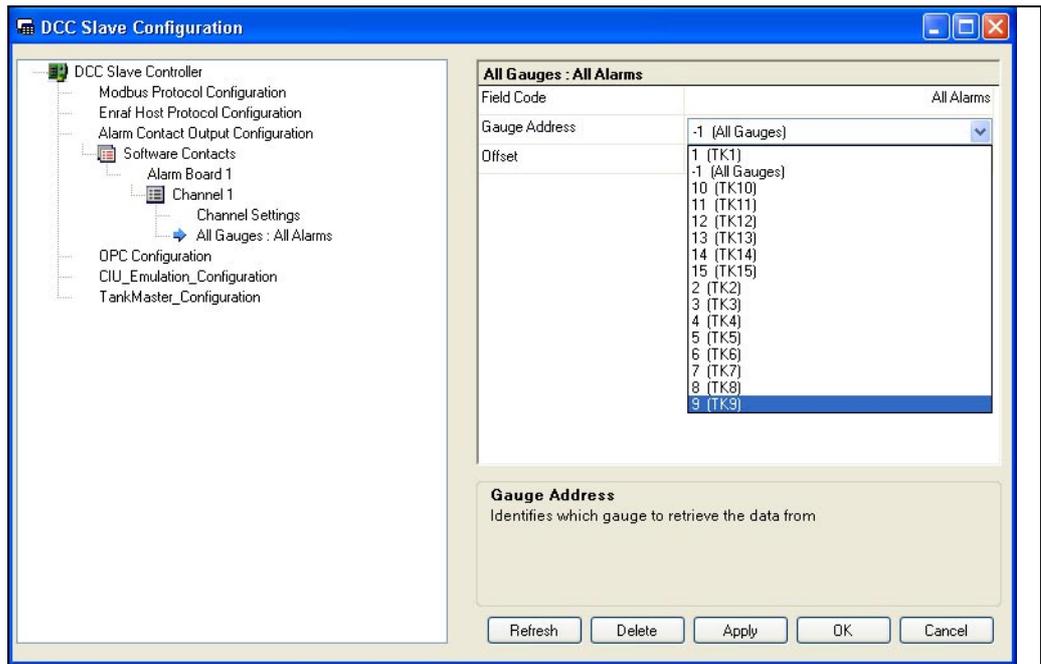
- **Field Code:** The name of the source alarm to test for. A value of "All Alarms" means all alarms will be checked.
- **Gauge Address:** The tank number to check for the alarm. This should be the same as the Gauge Number from the Gauge Configuration screens. A value of -1 means ALL Gauges will be checked.
- **Offset:** This is the bit number (1 to 16) within a block of 16 bits. It is only used if the Alarm Board settings have Use Blocks set to YES. If blocks are not used then set this always to 1. For Modbus devices using blocks this means that this is the bit number within the register.

When adding a new Field it will default to All Gauges and All Alarms as the source. This may then be modified to select individual alarms and/or gauges as required. A number of fields may be added to the same channel, with each field defining an individual alarm and gauge.

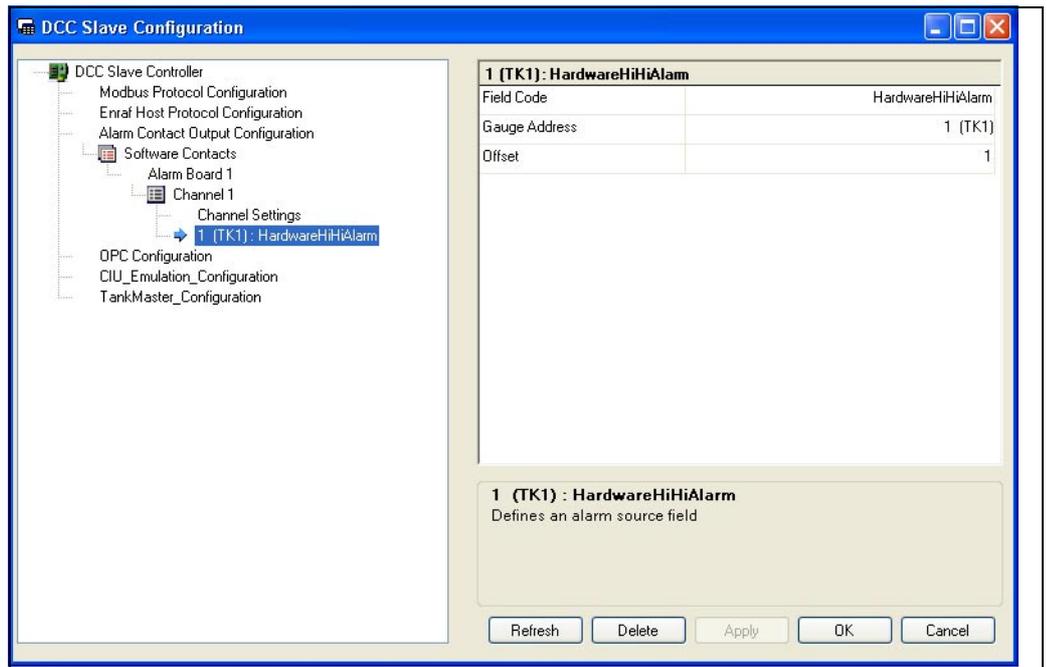
For instance if the relay channel was required to activate on the HI HI Gauge alarm for tank TK001 only then set the following:



NXA83\_DCC-Slave-Configuration\_Alarm\_All-Gauges\_Field-Code



NXA83\_DCC-Slave-Configuration\_Alarm\_All-Gauges\_Gauge-Address



NXA83\_DCC-Slave-Configuration\_Alarm\_All-Gauges\_HarwareHiHiAlarm

## 7.11 TankMaster Configuration

The TankMaster interface in DCC Slave is based on the Modbus protocol, with a pre-configured map. It supports up to 256 tanks, each with its own Tank Index position (range 0 to 255) in the modbus register map.

The interface is defined in the document "SP0142-12 - ICD Iss 1-0-0". Please refer to this document for a full definition of the functionality supported.

The only configuration required in DCC for the TankMaster interface is to define how the tanks in the Tankgauging database are mapped to the Tank Index positions in the predefined TankMaster map.

### 7.11.1 Default Tank Index mapping

By default the Index position in the TankMaster map is defined in the field **Datacon Gauge Index** in the **Device Configuration** screen.

The screenshot shows the 'Device Configuration' window. The 'Datacon Gauge Index' field is highlighted with a red box and contains the value '0'. Other fields include 'Device Number' (1), 'Master Datacon Name' (BLANK), 'Slave Datacon Name' (BLANK), 'Device Type' (Enraf 854), 'Site Id', 'Tank Id' (TK1), 'Port Address' (1), 'Comm Port' (99), 'CIU Address' (1), 'Gauge Address' (1), 'Type Of Instrument' (B), 'Type of Record' (B), and several checkboxes for 'Multi Gauges', 'Backup CIU Available', 'Average Temp. Fitted', 'Skin Temp. Fitted', and 'Hybrid Density'. A 'Background Scan' section lists Auxiliary 1 through Auxiliary 16. A 'Change Dimensions' section has dropdowns for Level, Temp, Density, Pressure, Volume, and Mass, all set to 'AUTO'. 'Flow Rate Damping' is 15 and 'Flow Rate Integration' is 10. Buttons for Edit, Copy, Add, Delete, Cancel, Save, Download, and Exit are visible.

1 Device configuration screen: Datacon Gauge Index

This should start at zero for the first index position in the TankMaster map.

 The **Datacon Gauge Index** field is only displayed if a **Master Datacon Name** is selected. If no datacon is used, then add a dummy datacon (such as the preconfigured BLANK datacon device in some systems). The Configuration screen will also require the **Port Address** to be entered, in this case just enter a value of 1 for this.

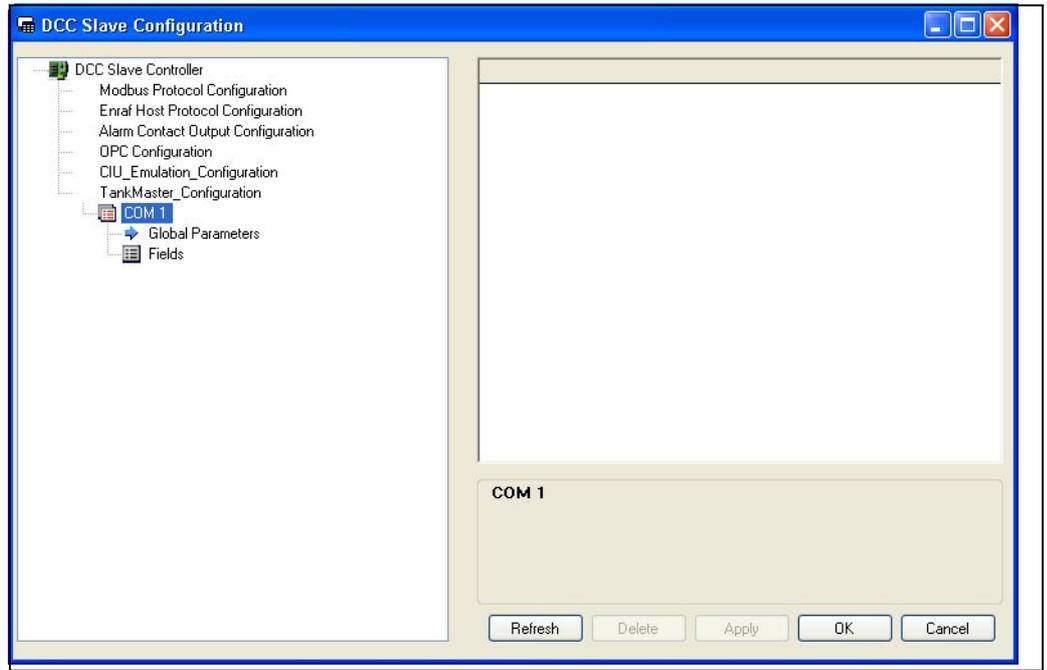
Each gauge must have a unique index position. Therefore, when the TankMaster interface is used, every gauge in the database should have a value entered in the **Datacon Gauge Index** field.

All Tankmaster interfaces configured in DCC Slave will use the same Tank Indexing as described above, unless individual interfaces are configured using the advanced mapping.

### 7.11.2 Advanced Tank Index mapping

Individual TankMaster interfaces may have their own Tank Index allocation via the **DCC Slave Configuration** screen.

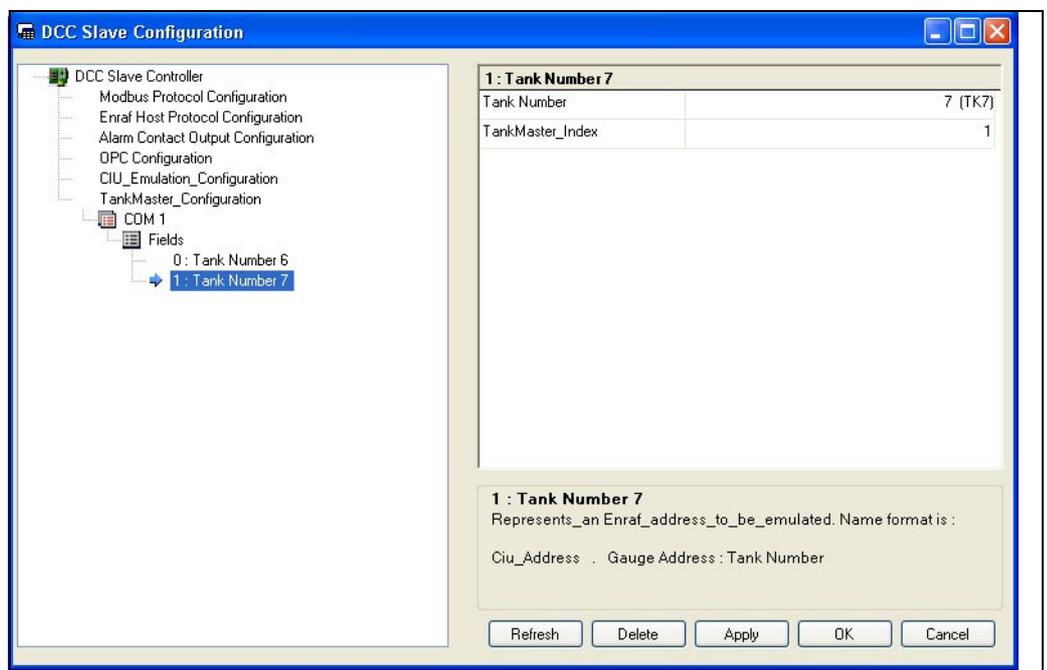
To configure a Tank Index map for a port, select the COM port under the TankMaster item.



NXAB3\_DCC-Slave-Configuration\_TankMaster-Configuration

Add all the Indexes required in the Fields section. Each entry consists of a pair of items:

- Tank Number in the database
- TankMaster Index position



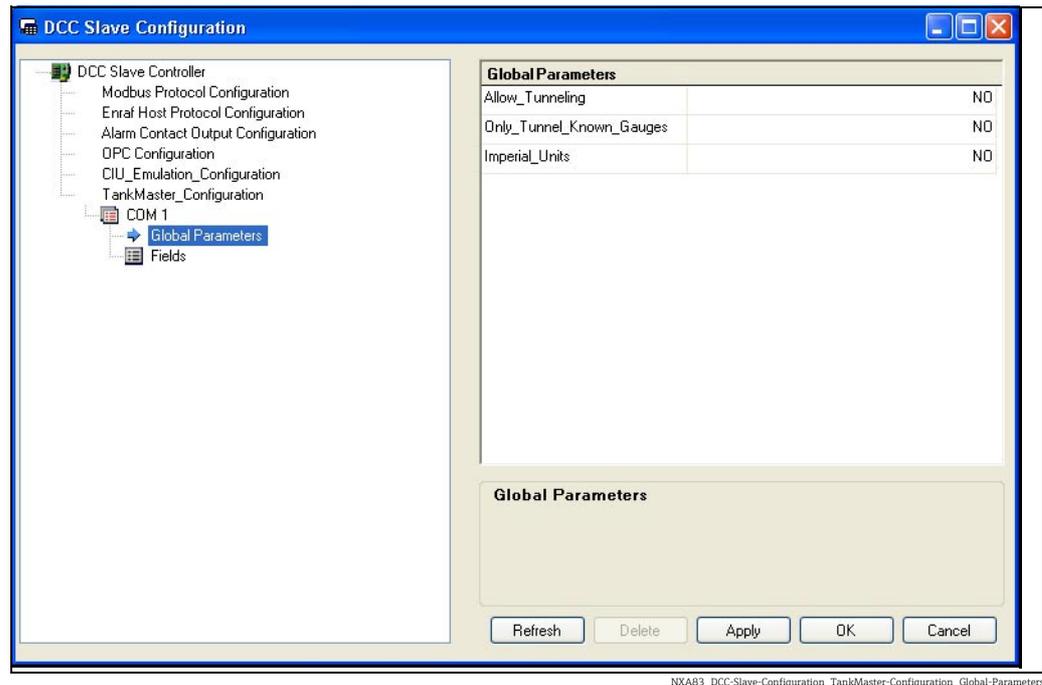
NXAB3\_DCC-Slave-Configuration\_TankMaster-Configuration\_Fields

The TankMaster interface running on that COM port will now use the map defined in the **DCC Slave Configuration** tool, and will NOT use any Datacon Gauge Index values.

### 7.11.3 Other Advanced Configuration

The **DCC Slave Configuration** screen can also be used to configure a number of other parameters for individual ports.

Click on the **Global Parameters** section for the port.



NXA83\_DCC-Slave-Configuration\_TankMaster-Configuration\_Global-Parameters

The following items may be configured:

- **Allow Tunneling:** If set to YES, requests for Modbus Slave IDs different to the one for the TankMaster interface will be tunneled to all Modbus protocol master interfaces. The reply from any gauge will be returned.
- **Only Tunnel Known Gauges:** If set to YES, only Modbus IDs for gauges configured in the database will be tunneled.
- **Imperial Units:** By default a TankMaster interface will format all data in metric units. Set this field to YES to format in Imperial/English units.

# Index

## C

- Comms Monitor ..... 45
- Configuring a Host or Slave port ..... 9–10
- Configuring Host COM Ports ..... 18
- Configuring Host Modem Ports ..... 21
- Configuring Host Network Ports ..... 24
- Configuring Modem Ports ..... 57
- Configuring Network Ports ..... 57
- Configuring OPC Network Ports ..... 27
- Configuring Scheduled Connections ..... 29
- Configuring Slave COM Ports ..... 54
- Connection Status ..... 49

## D

- Data Communications Controller ..... 7
- Datacon Slave Configuration ..... 92
- DCC Config Tool ..... 8
- DCC Host ..... 18, 45
- DCC Host Configuration ..... 18
- DCC Host Runtime ..... 45
- DCC Options ..... 33
- DCC Services ..... 8
  - Starting ..... 8
  - Stopping ..... 8
- DCC Slave ..... 54
- DCC Slave Configuration ..... 54
- Designated use ..... 5

## E

- Enhanced Raw Data ..... 51
- Enraf Protocol Configuration ..... 85
  - Configuring Communication and Data Format ..... 85
  - Downloading to Tank Fields ..... 92
  - Resetting the Configuration ..... 89
  - Tank Field Codes ..... 89
- Events ..... 11
- Extended Modbus Configuration ..... 92

## F

- Field Definitions ..... 40

## H

- Host COM Ports ..... 18
- Host Modem Ports ..... 21
- Host Network Ports ..... 24

## I

- Icons ..... 3

## L

- License details ..... 16

## M

- Modbus Protocol Configuration ..... 61
  - Adding and Deleting Entries ..... 64
  - Byte Ordering ..... 83
  - Configuring the Map ..... 61

- Construction Clipboard ..... 67
- Data Types ..... 84
- Field Codes ..... 70
- Filtering Field Code Selection ..... 64
- Making Changes ..... 70
- Planning ..... 61
- Setting Global Parameter Values ..... 66
- Setting Invalid Parameter Values ..... 66
- Modbus TG Advanced Map ..... 35
- Modbus TG Basic Map ..... 34
- Modbus TG protocol configuration ..... 33
  - Field Definitions ..... 40
  - Modbus TG Advanced Map ..... 35
  - Modbus TG Basic Map ..... 34
  - User Defined Alarms ..... 41
- Modem Log ..... 48
- Modem Ports ..... 57

## N

- Network Ports ..... 57

## O

- OPC Network Ports ..... 27

## P

- Protocol Configuration ..... 58

## R

- Redundancy ..... 31
- Relay Contact Output Configuration ..... 93
  - Configuring Channels ..... 95
  - Configuring Fields ..... 96
  - Configuring Modbus Devices ..... 94
  - Configuring Relay Boards ..... 93

## S

- Safety icons ..... 3
- Scheduled Connections ..... 29
- Slave COM Ports ..... 54
- Strap Tables ..... 10
- Symbols ..... 3

## T

- Tank data ..... 12
- TankMaster Configuration ..... 99
  - Advanced Tank Index mapping ..... 100
  - Default Tank Index mapping ..... 99
  - Other Advanced Configuration ..... 101
- TAS Data ..... 14

## U

- User Defined Alarms ..... 41

## V

- Viewing Modem Log ..... 48
- Viewing the license details ..... 16

**W**

Watching a Tank ..... 47





71303675

[www.addresses.endress.com](http://www.addresses.endress.com)

---