System Integrator's Guide
MultiSmart Pump Station Manager

Contents

1 Introduction to MultiSmart 3

2 Purpose of this document 4

3 Integration to any SCADA 4

4 Tag database 4

5 Which tags should be integrated? 5

5.1 Examples of tags or points not used in an installation 5

5.2 Active Faults and Unacknowledged Faults 5

5.3 Integration approach 5

5.4 Basic 2-pump application (1 well) – 18 tags 6

5.5 Intermediate 2-pump application – 54 tags 8

6 Enabling software modules in MultiSmart 14

7 DNP3 configuration 14

7.1 DNP security 14

7.2 Common things to look for in profile document of DNP3 master 14

7.3 Event buffering (SOE vs Most Recent) 15

7.4 Planning for communications outages 16

7.5 Multiple DNP slaves 16

7.6 Peer to peer / “Store and forward” 17

7.7 IO and deadbands 17

7.8 Remote configuration 17

8 Modbus configuration 17

9 Export points list to csv 17

10 Media 18

11 Channels and ports 18

12 Comms redundancy 18

13 Comms troubleshooting 19

13.1 Basic 19

13.2 Advanced - How to turn on DNP3 logging 20

1 Introduction to MultiSmart

MultiSmart is a pump station manager - a new category of product - which combines the best of pump controller functionality, PLC flexibility, an RTU, and some control panel components.

The main functional blocks of MultiSmart include:

- Pump controller –extensive functionality to cover single pump through to 9 pump applications, including multiple groups of pumps
- Flow module – handles calculated flow or flow meters with common tags
- Supply protection – directly measures 3-phase supply and provides monitoring, datalogging and voltage protection
- Energy, power and motor protection module – measures 3-phase currents for each pump from CTs, calculates power, power factor, apparent power, energy, apparent energy, and provides motor protection functions; measures insulation resistance via 1000v megger test
- RTU – Modbus master/slave, DNP3 master/slave, allowing multiple masters and slaves (DNP security will be implemented in v2.0, available in Q3 2008)
- Custom logic – via “mini-logic engine”
- PLC functionality (all 5 languages) to IEC61131-3 standard (available from v2.0 of the product)

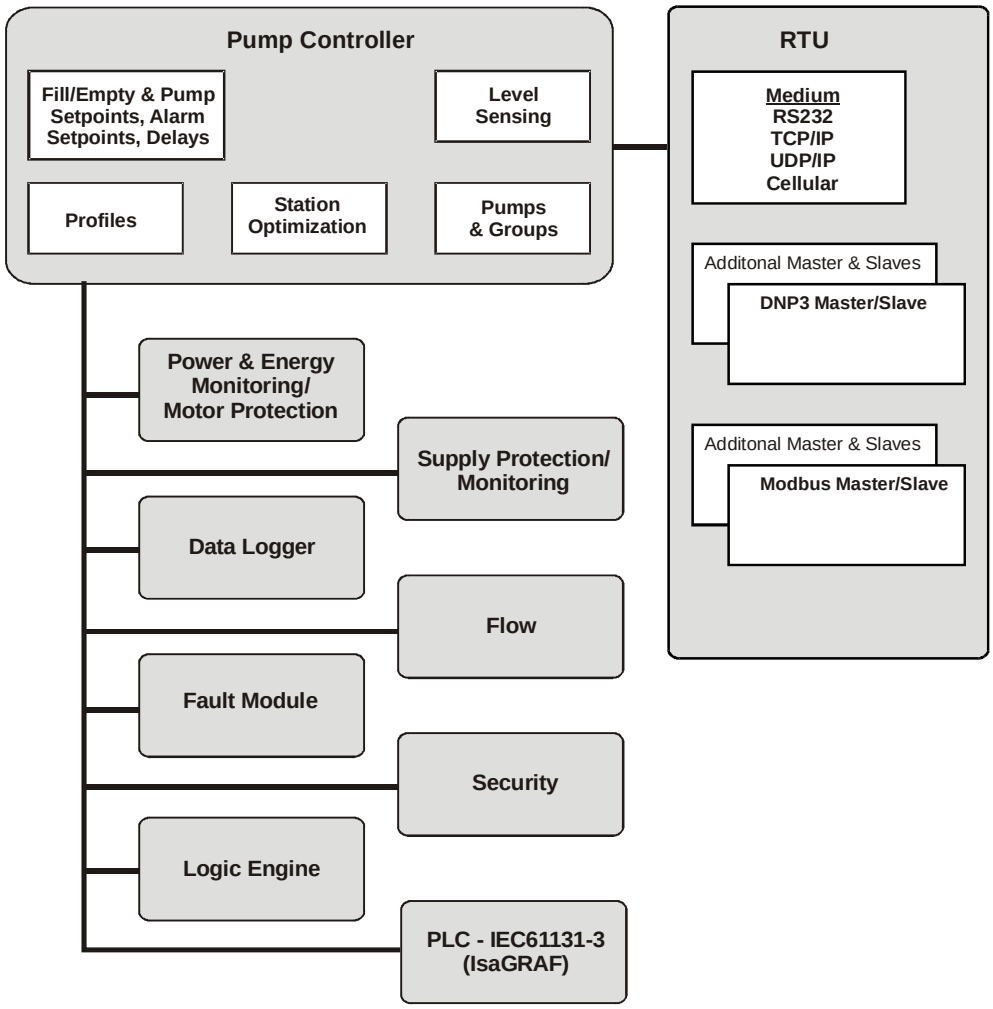

Figure 1 – Functional Blocks available in MultiSmart

2 Purpose of this document

This System Integrator's Guide is intended as a companion document to the Installation and Operation Manual (I&O manual) for MultiSmart. It is specifically written to enable System Integrator's to integrate MultiSmart rapidly into SCADA systems.

For electrical connection, commissioning, navigating the operator interface, and understanding the **Settings** menus in the product, see the I&O manual.

The I&O manual also has a communications chapter (Section 22) which should be read before starting with this Integrator's Guide.

3 Integration to any SCADA

The MultiSmart integrates to any modern SCADA – via Modbus or DNP3, and over any media.

There are also 3rd party tools available to speed up integration, such as the Software Toolbox MultiSmart TOPServer (www.softwaretoolbox.com/multismart). The MultiSmart TOPServer provides an OPC server for the MultiSmart tag list, greatly speeding up integration into SCADA platforms.

MultiSmart has successfully been integrated with:

- Wonderware InTouch
- Citect 6
- RSView
- Intellution iFix
- Iconics Genesis32
- ClearSCADA
- Software Toolbox TOPServer ("middleware" providing DNP3-OPC connectivity, not a SCADA front end)

4 Tag database

At the heart of the MultiSmart product is a tag database with 1000s of tags. These tags provide extensive functionality.

For example, if "Level" is being measured from AIN1, there will be an IO tag (AIN1) as well as a "Level" tag. The level tag is what is used in the pump control engine.

If the flow module is used, the "Pump 1 Flow rate" tag is used regardless of whether the Flow rate is a calculated value from the draw down test, or from an AIN via a flow meter.

5 Which tags should be integrated?

The tags available via Modbus or DN3 are listed in the profile documents, available from the MultiTrode website – www.multitrode.com.

The points list is extensive.

Also, note that additional tags can easily be added to the points list. See the notes in sections 5.4 and 5.5).

Because of the pre-configured nature of the product there are a large number of tags available for integration. There will often be many tags not used in a specific MultiSmart installation.

5.1 Examples of tags or points not used in an installation

The profile document includes DNP3 points (and Modbus points) for:

- "Pump 1 Seal fault" – this fault will only exist when a pump seal sensor is wired into MultiSmart and configured
- "Well 1 Backup level invalid" – this fault only occurs when a backup level device is configured
- "Well 2 Level" – this fault only occurs in 2 well applications

Therefore, there can be a large number of tags which will not be configured in any one installation.

5.2 Active Faults and Unacknowledged Faults

MultiSmart faults often have 3 states – active; unacknowledged and cleared

- Active is the typical fault state where the fault condition is present
- Unacknowledged is a very useful fault condition, indicating that the fault has occurred, the source of the fault has cleared, but the indication remains for operations that the fault was present

Therefore, 2 tags are often used for one fault, e.g. Under-voltage has 2 tags:

- Faults.Station.UnderVoltage.Status.Active
- Faults.Station.UnderVoltage.Status.Unacknowledged

The integration can include both tags, or simply the active tag.

Resetting the fault is now available when the fault is in the unacknowledged state. A number of fault reset tags exist to allow pump and station faults to be cleared. Some of these can be seen in the Intermediate 2-pump application (section 5.5).

5.3 Integration approach

We give 2 examples to make integration of a MultiSmart pump station as easy as possible. These are indicative only, to assist integrators new to the product.

To see an example of a more extensive implementation, with a few hundred tags per site, review the Outpost2 functional specification, available on www.multitrode.com

The definitive tag list is always the DNP3 and Modbus Slave profiles, available on the MultiTrode website.

5.4 Basic 2-pump application (1 well) – 18 tags

Description	Tag ID	DNP ID (default)	Modbus ID (default)	Comments
Pump 1 Running (called)	PumpControl._1.Running	Binary Input 70	Discrete Input 62	This means that the output control relay has been asserted. Feedback from the contactor can be brought back, details in section 5.5)
Pump 2 Running (called)	PumpControl._2.Running	Binary Input 130	Discrete Input 122	
Pump 1 Unavailable	PumpControl._1.FaultStatus.UnavailableActive	Binary Input 73	Discrete Input 65	The pump can be unavailable when supply faults are present as well as when pump faults exist. A pump fault does not have to make the pump unavailable, e.g. the default configuration for a seal fault. Any fault can be defined to be a display only fault, an auto reset fault (pump becomes available when fault condition clears), or a manual reset fault (pump only becomes available when manual or SCADA reset takes place)
Pump 2 Unavailable	PumpControl._2.FaultStatus.UnavailableActive	Binary Input 133	Discrete Input 125	
Pump 1 Active Fault	Fault.Pump._1.Active	*See note 1*	*See note 1*	
Pump 2 Active Fault	Fault.Pump._2.Active	*See note 1*	*See note 1*	
Level	PumpControl.Well._1.CurrentLevel or PumpControl.Well._1.ScaledLevel	Analog Input 10 or Analog Input 12	Input Register 38 or see Note 3 (section 5.5)	Currentlevel = level in %, ScaledLevel can be defined by the user so it displays on the MultiSmart screen, e.g. 0-5m, 0-10ft
High Level alarm	Faults.Well._1.HighLevel.Status.Active	Binary Input 32	Discrete Input 30	There are 2 high level alarms available. High and High-High. See Note 2.
Supply fault – Under-voltage Over-voltage Phase imbalance	Faults.Station.UnderVoltage.Status.Active Faults.Station.OverVoltage.Status.Active Faults.Station.VoltsPhaseImbalance.Status.Active	Binary Input 0 Binary Input 2 Binary Input 4	Discrete Input 0 Discrete Input 2 Discrete Input 4	
Supply, phase AB	IO.Unit._1.TopBoard.Vin._1.VoltsAb	Analog Input 47	Input Register 113	
Supply, phase BC	IO.Unit._1.TopBoard.Vin._1.VoltsBc	Analog Input 48	Input Register 114	
Supply, phase CA	IO.Unit._1.TopBoard.Vin._1.VoltsCa	Analog Input 49	Input Register 115	
Inflow	Flow.InflowRate	Analog Input 4	Input Register 21	
Pump 1 Flow rate	Flow.Pump._1.FlowRate	Analog Input 21	Input Register 63	
Pump 2 Flow rate	Flow.Pump._2.FlowRate	Analog Input 31	Input Register 93	
Station volume	Flow.VolumePumped	Counter 4	Input Register 22	

Note 1:

*Each pump has a number of different faults associated with it, depending on which faults are wired in/configured. Each one of these faults can be separately brought back and has a default tag associated with it. By default, there is no Any Fault tag for each pump. To add the DNP tag configure via the LCD, go to **Settings – More – Communications – DNP3 – DNP slave – Binary Inputs** and press **Add New**. This allows you to choose the Point Number (or accept the default). Navigate down to **Tag** and press **Select**. (To add the Modbus tag, follow the same process with Modbus instead).*

*You are now in the Tag Browser. Navigate through to **Faults-Pump-Pump 1-Active** and press **OK**. This has now configured a new DNP point to be Pump 1 Any Active Fault. Follow a similar process for Pump 2.*

Note 2:

***High** and **High-High** are 2 separate level alarms, with **High** level enabled by default. To bring back **High-High** separately, the default tag is Binary Input 30. To bring back **Any Level Alarm** as one tag, you need to configure a new point. Follow the process as with Note 1 above, and navigate through the tag browser to **Pump Control – Well - Well 1 – Level Alarm Summary – Any level alarm**.*

5.5 Intermediate 2-pump application – 54 tags

Description	Tag ID	DNP ID (default)	Modbus ID (default)	Comments
Pump 1				
Pump 1 Running (called)	PumpControl.Pump._1.Running	Binary Input 70	Discrete Input 62	
Pump 1 Mode (Auto/Off/Manual or Hand)	PumpControl.Pump._1.PumpMode	Analog Output 18	Holding Register 18	5 states: 0 – Auto, 1 – Manual, 2 – Semi-auto, 3 – Off, 4 - Decommissioned
Pump 1 Unavailable	PumpControl._1.FaultStatus.UnavailableActive	Binary Input 73	Discrete Input 65	The pump can be unavailable when supply faults are present as well as when pump faults exist. A pump fault does not have to make the pump unavailable, e.g. the default configuration for a seal fault. Any fault can be defined to be a display only fault, an auto reset fault (pump becomes available when fault condition clears), or a manual reset fault (pump only becomes available when manual or SCADA reset takes place)
Pump 1 Thermal overload fault active	Faults.Pump._1.ThermalOverload.Status.Active	Binary Input 76	Discrete Input 68	
Pump 1 Thermal C/B off trip fault active	Faults.Pump._1.CBOffTrip.Status.Active	Binary Input 104	Discrete Input 96	
Pump 1 Contactor Auxiliary fault active (pump failed to start)	Faults.Pump._1.ContactorAux.Status.Active	Binary Input 98	Discrete Input 90	
Pump 1 Motor Overtemp fault active OR if Flygt FLS used	Faults.Pump._1.MotorOvertemp.Status.Active Faults.Pump._1.FlsThermal.Status.Active	Binary Input 102 Binary Input 80	Discrete Input 94 Discrete Input 72	
Pump 1 Seal fault active OR if Flygt FLS	Faults.Pump._1.Seal.Status.Active Faults.Pump._1.FlsSeal.Status.Active	Binary Input 74 Binary Input 78	Discrete Input 66 Discrete Input 70	
Pump 1 Fault Reset	Faults.Pump._1.GroupManualReset	Binary Output 20	Coil 20	This will reset all unacknowledged faults for this pump
Pump 2				
Pump 2 Running (called)	PumpControl.Pump._2.Running	Binary Input 130	Discrete Input 122	
Pump 2 Mode (Auto/Off/Manual or Hand)	PumpControl.Pump._2.PumpMode	Analog Output 24	Holding Register 24	5 states: 0 – Auto, 1 – Manual, 2 – Semi-auto, 3 – Off, 4 – Decommissioned (only Auto, Semi-Auto, and Off can be written to, see Note 6)
Pump 2 Unavailable	PumpControl.Pump._2.FaultStatus.UnavailableActive	Binary Input 133	Discrete Input 125	

Pump 2 Thermal overload fault active	Faults.Pump._2.ThermalOverload.Status.Active	Binary Input 136	Discrete Input 128	
Pump 2 Thermal C/B off trip fault active	Faults.Pump._2.CBOffTrip.Status.Active	Binary Input 164	Discrete Input 156	
Pump 2 Contactor Auxiliary fault active	Faults.Pump._2.ContactorAux.Status.Active	Binary Input 158	Discrete Input 150	
Pump 2 Motor Overtemp fault active OR if Flygt FLS	Faults.Pump._2.MotorOvertemp.Status.Active Faults.Pump._2.FlsThermal.Status.Active	Binary Input 162 Binary Input 140	Discrete Input 154 Discrete Input 132	
Pump 2 Seal fault active OR if Flygt FLS used	Faults.Pump._2.Seal.Status.Active Faults.Pump._2.FlsSeal.Status.Active	Binary Input 134 Binary Input 138	Discrete Input 126 Discrete Input 130	
Pump 1 Fault Reset	Faults.Pump._2.GroupManualReset	Binary Output 22	Coil 20	This will reset all unacknowledged faults for this pump
Level				
Level (scaled)	PumpControl.Well._1.ScaledLevel	Analog Input 12	See Note 3	
High high level alarm	Faults.Well._1.HighHighLevel.Status.Active	Binary Input 30	Discrete Input 28	
High level alarm	Faults.Well._1.HighLevel.Status.Active	Binary Input 32	Discrete Input 30	
Low level alarm	Faults.Well._1.LowLevel.Status.Active	Binary Input 34	Discrete Input 32	
Low low level alarm	Faults.Well._1.LowLowLevel.Status.Active	Binary Input 36	Discrete Input 34	
Level alarm reset	Faults.Well._1.GroupManualReset	Binary Output 8	Coil 8	This resets all well faults, see note 5
Supply				
Undervoltage fault	Faults.Station.UnderVoltage.Status.Active	Binary Input 0	Discrete Input 0	
Overvoltage fault	Faults.Station.OverVoltage.Status.Active	Binary Input 2	Discrete Input 2	
Phase fail fault	Faults.Station.VoltsPhaseImbalance.Status.Active	Binary Input 4	Discrete Input 4	
Phase rotation fault	Faults.Station.VoltsPhaseRotation.Status.Active	Binary Input 6	Discrete Input 6	
Supply fault reset	Faults.Station.GroupManualReset	Binary Output 1	Coil 1	This resets all station faults, see note 6
Supply, phase AB	IO.Unit._1.TopBoard.Vin._1.VoltsAb	Analog Input 47	Input Register 113	

Supply, phase BC	IO.Unit._1.TopBoard.Vin._1.VoltsBc	Analog Input 48	Input Register 114	
Supply, phase CA	IO.Unit._1.TopBoard.Vin._1.VoltsCa	Analog Input 49	Input Register 115	
DC supply	IO.Unit._1.DcVolts.Input.Value	Analog Input 40	Input Register 106	
Flow				
Inflow	Flow.InflowRate	Analog Input 4	Input Register 21	Value in gals or litres /sec depending on units configured in the product
Pump 1 Flow rate	Flow.Pump._1.FlowRate	Analog Input 21	Input Register 63	
Pump 2 Flow rate	Flow.Pump._2.FlowRate	Analog Input 31	Input Register 93	
Station volume	Flow.VolumePumped	Counter 4	Input Register 22	
Overflow fault active	Faults.Station.Overflow.Status.Active	Binary Input 18	Discrete Input 18	
Setpoints				
Lead pump activation setpoint	PumpControl.Behaviour._1.ActSetPoint	Analog Output 19	Holding Register 19	In the tag database, the lead pump is known as “behaviour 1”, lag pump “behaviour 2” and so on. If alternation is off, i.e., fixed sequence, then Behaviour 1 is the first pump to run.
Lead pump de-activation setpoint	PumpControl.Behaviour._1.DeactSetPoint	Analog Output 20	Holding Register 20	
Lag pump activation setpoint	PumpControl.Behaviour._2.ActSetPoint	Analog Output 25	Holding Register 25	
Lag pump de-activation setpoint	PumpControl.Behaviour._2.DeactSetPoint	Analog Output 26	Holding Register 26	
Power, energy & efficiency				
P1 Pump efficiency, yesterday	MotorProt.Pump._1.EfficiencyYesterday	Frozen Counters 133	See Note 4	Gals or litres / kWHr.
P1 Power, KW	IO.Unit._1.BottomBoard.Power._1.Power	Analog Input 84	Input Register 156	Value in KW
P1 power factor	IO.Unit._1.BottomBoard.Power._1.PowerFactor	Analog Input 85	Input Register 157	
P1 Energy, KWHr	IO.Unit._1.BottomBoard.Power._1.EnergykWh	Analog Input 86	Input Register 158	To capture energy used yesterday (for daily analysis) use IO.Unit._1.BottomBoard.Power._1.EnergyYesterdaykWh, which is DNP Frozen Counters 129 (no default Modbus equivalent)

P2 Pump efficiency, yesterday	MotorProt.Pump._2.EfficiencyYesterday	Frozen Counters 138	See Note 4	Gals or litres / kWHr
P2 Power, KW	IO.Unit._1.BottomBoard.Power._2.Power	Analog Input 87	Input Register 159	
P2 power factor	IO.Unit._1.BottomBoard.Power._2.PowerFactor	Analog Input 88	Input Register 160	
P2 Energy, KWHr, yesterday	IO.Unit._1.BottomBoard.Power._2.EnergykWh	Analog Input 89	Input Register 161	To capture energy used yesterday (for daily analysis) use IO.Unit._1.BottomBoard.Power._2.EnergyYesterdaykWh, which is DNP Frozen Counters 134 (no default Modbus equivalent)
Insulation Resistance of Motor Windings				
P1 IRT	IO.Unit._1.BottomBoard.Irt._1.Value	Analog Input 165		Value in MOhms
P2 IRT	IO.Unit._1.BottomBoard.Irt._2.Value	Analog Input 166		Value in MOhms

Note 3:
The default Modbus tag list (as of v1.5.6) does not include scaled level. Level in % is tag: PumpControl.Well._1.CurrentLevel and Modbus Input Register 38.

*To add via the LCD, go to **Settings – More – Communications – Modbus – Modbus slave – Input Registers** and press **Add New**. This allows you to choose the Point Number (or accept the default). Navigate down to **Tag** and press **Select**.*

*You are now in the Tag Browser. Navigate through to **Pump Control – Well – Well 1 – Scaled Level** and press **OK**. This has now configured a new Modbus analog (input register) as Scaled Level.*

Scaled level has 4 decimal places by default. Therefore, in order to fit this within a 16 bit register (which is all that Modbus supports), this value needs to be scaled. For current level, we scale this by 10,000 in order to display from 0-100%. For scaled level, you may want to have some decimal places (e.g., 0.00 – 10.00 ft: In this case, the scale factor should be 100).

Note 4:
The default Modbus tag list (as of v1.5.6) does not include pump efficiency.

*To add via the LCD, go to **Settings – More – Communications – Modbus – Modbus slave – Input Registers** and press **Add New**. This allows you to choose the Point Number (or accept the default). Navigate down to **Tag** and press **Select**.*

*You are now in the Tag Browser. Navigate through to **Motor Protection – Pump – Pump 1 – Efficiency Yesterday** and press **OK**. This has now configured a new Modbus analog (input register) as pump efficiency. Use a similar process to add pump 2.*

Note 5:
Faults.Well._1.GroupManualReset – resets the following unacknowledged faults :

- *HighHighLevel*
- *HighLevel*
- *LowLevel*
- *LowLowLevel*
- *PrimaryLevelHighRange*
- *PrimaryLevelLowRange*
- *PrimaryLevelInvalid*
- *PrimaryLevelAinOverRange*
- *PrimaryLevelAinUnderRange*
- *BackupLevelInvalid*
- *BackupLevelAinOverRange*
- *BackupLevelAinUnderRange*
- *Well 1 PulseStart*
- *Well 1 PulseStop*

Note 6:
Faults.Station.GroupManualReset – resets the following unacknowledged station faults:

- *UnderVoltage*
- *OverVoltage*
- *VoltsPhaseImbalance*
- *VoltsPhaseRotation*
- *DCUnderVoltage*
- *DCOverVoltage*
- *MaxOnTime*
- *MaxOffTime*
- *PowerFail*
- *Overflow*
- *PulseStart*
- *PulseStop*

Note 7:
"Manual mode" on the MultiSmart main display is really "Semi-Auto" mode – which means that the pump will run in manual operation until the cut-out or deactivation point is reached, at which point the pump mode will revert to "Auto".

A pump can be placed in true manual mode only by holding the button down on the faceplate. Then, the Manual indication will flash on the LCD and the pump will be forced to run regardless of the level of the well as long as the button is held down.

A pump can be decommissioned via ***Settings – Commision/ Decom*** *screen.*

6 Enabling software modules in MultiSmart

MultiSmart is supplied in different configurations in different markets. To check which software modules are enabled on your unit, from the main screen press

Settings → More → More → Software Modules → Enable_Disable Modules

This lists which modules are enabled and disabled. To enable a module which is currently disabled, contact MultiTrode. Some modules have a price tag, others do not incur any cost.

7 DNP3 configuration

MultiSmart DNP3 Level 2 is fully compliant to the standard as can be seen by the audit done by Goanna Technologies in 2007. See www.multitrode.com for the compliance document.

Not all DNP3 masters (Master Telemetry Units or IO servers in the SCADA front end) support all objects or object types. Usually their profile document will describe any limitations.

MultiSmart has great flexibility in the slave configuration to match any limitations in the master.

For example, the current version of ClearSCADA doesn't support **binary input with status** – the MultiSmart slave can be re-configured to handle this:

Settings → Advanced → DNP Slave → Slave → Slave 01 → Session → Object 1 Default Variation

Change the value to **Single bit Bin**.

7.1 DNP security

DNP security is coming soon, in v2.0 of MultiSmart, which is expected in Q3 2008.

7.2 Common things to look for in profile document of DNP3 master

Check which object variations are supported. Our default list is:

- Object 1 (Binary inputs) – 2 (binary input with status)
- Object 2 (Binary input events) – 3 (binary input event with relative time)
- Object 10 (Binary outputs) – 2 (binary output with status)
- Object 11 (Binary output events) – 2 (binary output event with time)
- Object 20 (Counters) – 1 (32 bit counter with status)
- Object 21 (Frozen counters) – 1 (32 bit frozen counter with status)
- Object 22 (Counter events) – 5 (32 bit counter event with time)
- Object 23 (Frozen counter events) – 1 (32 bit frozen counter event without time)
- Object 30 (Analog inputs) – 1 (32 bit analog input)
- Object 32 (Analog input events) – 3 (32 bit analog event with time)
- Object 40 (Analog outputs) – 1 (32 bit analog output)
- Object 42 (Analog output events) – 1 (32 bit analog output event without time)

Also, check that the other device supports all of these object types. In particular, binary output events and analog output events may not be supported (many devices do not support these object types).

If this is the case, then it may be necessary to map these outputs to binary and analog input points as well so as to get indication of change of state for these points.

Many DNP masters do not support events with time stamps. Therefore, it may be necessary to modify these variations (for objects 2, 11, 22, and 32).

Not all devices support frozen counters either (including ClearSCADA).

7.3 Event buffering (SOE vs. Most Recent)

MultiSmart DNP Slave has the ability to buffer events between polls. The maximum number of events is defined for each object type, and can be reconfigured. However, using excessively large buffers could result in bandwidth issues (potential for very large messages being generated in response to an event poll). These buffer sizes can be reconfigured via the Advanced screen under:

DNP Slave → Slave → Slave 01 → Session

The following parameters exist:

- Max analog input events
- Max analog output events
- Max binary input events
- Max binary output events
- Max counter events
- Max frozen counter events

When multiple events occur within a single poll period for a particular point, it is possible to buffer all of the events for that point or to only buffer the last value. This behaviour can be defined for each DNP object type. The following parameters allow for this configuration:

- Analog input event mode
- Analog output event mode
- Binary input event mode
- Binary output event mode
- Counter event mode
- Frozen counter event mode

Each of these modes can be set to SOE (sequence of events) or most recent. If SOE is selected, then all events are buffered and sent back in response to an event poll. For Most Recent, only the latest value is returned via an event poll.

MultiSmart contains statistics under Info → Communications → DNP Slave - specifically the **Overflows** value is important. It will tell you if the buffer size is too small.

Note:
*The **default** DNP configuration in MultiSmart is for **Most Recent** except for binary inputs and binary outputs. The binary classes both use sequence of events (SOE).*

7.4 Planning for communications outages

The previous section explains how to set up SOE rather than using Most Recent values. This is especially useful for dealing with communications outages.

The principle is straightforward. Simply calculate:

- What is the maximum likely downtime – e.g. replacement time for a radio or a repeater
- What are the typical or worst case number of events in each type during this period

Set up the SOE size accordingly. The size can often be estimated by reviewing the statistics of the master station for each site.

You need to consider the impact of a repeater failing in a radio network – when all of the sites come back on line, how much traffic will be required to flow through the network. Therefore, there is a trade off between the desired buffer size for one site and the amount of traffic that might be required if a repeater fails then comes back on line.

7.5 Multiple DNP slaves

It is possible to configure a MultiSmart unit with multiple DNP Slaves. This is particularly useful if a redundant communications path is desired. In order to set up a second channel, do the following:

Go to Settings → More → Communications → DNP3 on the LCD, and press **Add DNP Slave**

Once the unit restarts, set up the communications channel for the second slave. This can be done via the Advanced screen.

- Advanced → DNP Slave → Slave → Slave 02 → Session
- Set the Comms Channel parameter to the desired comms channel
- If you need to modify the baud rate and other comms parameters:
  - Go to Advanced → Telemetry → Channel (Selected channel), and modify parameters.

The slave will be created with a blank points list. If you want the second slave to be configured with the default points list:

- Go to Settings-> More-> Communications-> DNP3-> DNP Slave
- Press Next Slave to bring up Slave 2 configuration.
- Press More, and then press Regenerate Points List
- When the unit restarts, the second slave will have the default points list

7.6 Peer to peer / "Store and forward"

MultiSmart supports multiple masters and slaves, and is also able to have multiple devices (e.g. one master + one slave or two slaves) on one comms port, using the **Connection manager** feature.

This is currently disabled by default – see section 0 for how to determine this – but needs to be enabled if the requirement is for multiple devices on the one port.

As well as communicating to an MTU as a slave device, MultiSmart can also operate as a Modbus or DNP Master device, talking to other slave devices. When set up in this manner, it is possible for a command to be sent via the MTU to the MultiSmart, which then forwards that command on to other slave device/s. The values received from the slave devices can also be sent back to the MTU by the MultiSmart device.

In order to forward commands, first define the command as a binary or analog output (DNP), or coil or holding register (Modbus). There are a pool of spare tags reserved for this purpose. For DNP, this pool of tags can be located under DnpMaster.Master._1.Slave._1.SpareTags. For a binary output, link to one of the tags under DigitalControls. For an analog output, link to one of the tags under AnalogControls.

Then, link this same tag as a binary or analog output in DNP Slave (which communicates back to the MTU). Then, when the device receives a command from the MTU, the master running on the MultiSmart device will automatically trigger a command to the slave device.

7.7 IO and deadbands

DNP3 works on reporting change of state. Analog values report a change of state when they change outside of their deadband.

Deadband values are accessed by accessing the DNP3 configuration on the LCD:
Settings → More → Communications → DNP3 → DNP3 Slave → Analog Inputs, then select a tag and the deadband, as well as other DNP parameters, are available to be changed. In v2.0, the DNP (and Modbus) configuration can be exported as .csv, amended and re-imported. This export/import will include deadband as well as other DNP parameters.

7.8 Remote configuration

MultiSmart supports DNP3 file transfer and will shortly be implementing a complete remote upgrade feature.

8 Modbus configuration

Because of the limited nature of Modbus, inter-operability issues are much less common, however, they can occur. This is often due to non-standard implementations of Modbus such as extensions or limitations in the master.

The MultiSmart Modbus master and slave are completely compliant to the standard and have been tested against many other devices. MultiSmart supports Modbus RTU, Modbus ASCII and Modbus TCP.

9 Export points list to csv

For easier management of the DNP3 and Modbus points list, the MultiSmart has an "Export to csv function" from the LCD interface. This creates a csv and copies it onto a Compact Flash card.

The Points list can also be cleared and regenerated via the LCD.

MultiSmart v2.0, available in Q3 2008, will support importing back from csv – allowing integrators the ability to manage and change the points list on a PC interface.

10 Media

MultiSmart supports a variety of communications media:

- Cellular data
  - GPRS
  - 3G
  - Next G (Australia)
  - 1XRTT (CDMA data)
- PSTN
- GSM voice
- RS232 locally
- RS232 for private and spread spectrum radio
- Ethernet radio
- Local Ethernet
- TCP/IP
- UDP/IP

11 Channels and ports

This is fully covered in section 22 of the I&O manual.

12 Comms redundancy

For redundancy, in a totally polled DNP environment, all that is needed are the 2 DNP slaves connected over different physical media. Likewise, in a Modbus environment all that is needed are 2 Modbus slaves. The master station will take care of trying to communicate over the different channels.

In unsolicited mode in DNP, the slaves themselves need configuration.

In the Advanced screen, **Show more options** needed to be checked.

Advanced → LCD **-> Show more options** and check the box, then exit the advanced screens, and when you re-enter the Advanced screens more options will be available in many areas (see also section 30.1 of the I&O manual).

For the first slave, navigate to Advanced → DNP Slave → Slave 01 → Session

- Set UnsolAllowed to true
- Set UnsolClassMask to the desired classes (typically **All**)

Navigate to Advanced → DNP Slave → Slave 01 → Communications

- Set UnsolPrimary to true
- Set the alias for UnsolPairActivate to DnpSlave.Slave._2.Comms.UnsolActivate
- Set the alias for UnsolPairDeactivate to DnpSlave.Slave._2.Comms.UnsolDeactivate

For the second slave, navigate to Advanced → DNP Slave → Slave 02 → Session

- Set UnsolAllowed to true
- Set UnsolClassMask to the desired classes (typically **All**)

Navigate to Advanced → DNP Slave → Slave 01 → Communications

- Under the Comms node, set UnsolPrimary to false
- Set the alias for UnsolPairActivate to DnpSlave.Slave._1.Comms.UnsolActivate
- Set the alias for UnsolPairDeactivate to DnpSlave.Slave._1.Comms.UnsolDeactivate

When the device is started, the first slave will have unsolicited responses enabled, and the second will have them disabled. If the first slave attempts to send an unsolicited response, but receives no confirmation from its master, then it will eventually go into comms fail state after the period of time defined by CommsFailTime. When this occurs, it will disable unsolicited responses and will activate unsolicited responses for the other slave.

When the second slave has unsolicited responses enabled, if it goes into comms fail state, then it will disable unsolicited responses and activate unsolicited responses for the primary slave again.

When the primary slave has unsolicited responses disabled, if it goes back online (i.e., its master sends through a successful poll), then it will automatically re-enable unsolicited responses, and will deactivate unsolicited responses for the secondary slave.

13 Comms troubleshooting

13.1 Basic

Basic level troubleshooting of DNP3 or Modbus communications can be done via the LCD interface:

Info → Communications → DNP3 → DNP3 Slave or

Info → Communications → Modbus → Modbus Slave

If a master has been configured, the master statistics can be seen via:

Info → Communications → DNP3 → DNP3 Master or

Info → Communications → Modbus → Modbus Master

13.2 Advanced - How to turn on DNP3 logging

DNP Logging can be turned on by editing the file /var/config/master.xml. There will be an entry that looks like this:

```
<Enable version="00.00.00: module="DNPSlave">
                <Launch file="dnps">
                                <Argument arg="-cDNPSlave.slave._1"/>
                </Launch>
                <Include file="dnpslave.xml"/>
</Enable>
```

Add the following line to this entry:

```
<Argument arg="-l/tmp/dnps.log"/>
```

The entry will now look like this:

```
<Enable version="00.00.00: module="DNPSlave">
                <Launch file="dnps">
                                <Argument arg="-cDNPSlave.slave._1"/>
<Argument arg="-l/tmp/dnps.log"/>
                </Launch>
                <Include file="dnpslave.xml"/>
</Enable>
```

Then, restart the RTU. When it restarts, a log file called /tmp/dnps.log will be continually updated and will give valuable diagnostic information.