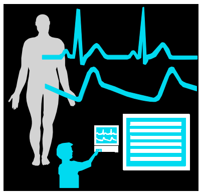
IntelliVue X2, MP Series & MX Series



DATA EXPORT INTERFACE PROGRAMMING GUIDE

IntelliVue Patient Monitor

X2, MP Series, MX Series

Patient Monitoring

4535 642 59271

PHILIPS

1

About this Guide

This Programming Guide is for use with the Philips IntelliVue X2, MP Series and MX Series patient monitors, hereafter referred to as the IntelliVue monitor. It describes the functionality in the monitor software version H.0.xx and later.

The information in this Programming Guide describes the capability of the Data Export Interface. It is the responsibility of the user to create applications using the capability provided.

This device is not intended for home use.

In this guide

- A warning alerts you to a potential serious outcome, adverse event or safety hazard. Failure to observe a warning may result in death or serious injury to the user or patient.
- A caution alerts you where special care is necessary for the safe and effective use of the product. Failure to observe a caution may result in minor or moderate personal injury or damage to the product or other property, and possibly in a remote risk of more serious injury.

Who Should Use this Guide?

This programming guide is intended to be used by software professionals and biomedical engineers at medical research clinics or industrial institutions.

To successfully create an application, users should have a **good** working knowledge of:

- Advanced software application design.
- C and/or C++ Programming Language.
- General digital communications theory.
- Local Area Network configuration guidelines.and communication protocols.
- RS232 communication protocols and the IrDA protocol.

Given this background knowledge, this Programming Guide provides the information necessary to create your own applications.

Philips cannot provide any technical assistance for individual programming efforts.

About the Data Export Interface

This document describes the IntelliVue Data Export Interface. Using a communication interface protocol, data from the Philips IntelliVue Patient Monitor can be transferred via the Local Area Network (LAN) Interface or Medical Information Bus (MIB/RS232) Interface to an external Computer.

By creating basic applications using the IntelliVue Data Export Interface, the following data can be accessed from the IntelliVue monitor:

- All measurement numerics and alarm data (real-time update rates up to 1024 ms).
- Wave data (see “Interpreting Wave Data” on page 286 for details)
- IntelliVue monitor system data.
- Patient demographic data entered by the user in the IntelliVue monitor.

The IntelliVue Data Export Interface cannot be accessed via the Local Area Network when the IntelliVue monitor is connected to the Philips LAN, e.g. to an Information Center (central station). Communication via the MIB/RS232 Interface is always possible (except with MP2/X2).

CAUTION

- Although alarm data can be accessed using the protocol, it must not be used as a real-time alarming system due to the delays in message transfer and the possibility of data loss.
- The computer client (the interfacing system) and/or the user of the communication system must comply with applicable data privacy regulations.

Data Export Interface Features

- The IntelliVue Data Export Interface uses the Local Area Network (LAN) and MIB/RS232 interfaces.
- The LAN interface uses the standard UDP/IP transport protocol.
- The MIB/RS232 interface can be configured to use either a fixed or a variable baudrate protocol.
- The Data Export Protocol is a connection-oriented, message-based request/ response protocol on top of the transport protocol. The UDP and fixed baudrate transport protocols are connection-less, whereas the variable baudrate protocol is connection-oriented.
- The LAN interface supports automatic configuration of the network IP address with the standard BootP protocol.

Changes in Rev. G.0

IntelliVue Rev. G.0 and higher differs from Rev F.0 and lower in the nomenclature of some numeric and wave labels. The labels that previously resided in the namespace NOM_EMFC are now merged into the NOM_SCADA namespace and the new defined NOM_SETTING namespace.

For details on identifying the software revision of the client interface protocol, please refer to “Building a Computer Client” on page 279

Manufacturer’s Information

The information contained in this document is subject to change without notice. Philips makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Philips shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Philips assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Philips. Philips reserves the right to change the protocol described in this document without prior notice. Philips makes no warranty of any kind with regard to software applications that are created by the user. Philips assumes no responsibility for adverse interaction of an external Computer Interface communicant with the Philips IntelliVue Series Patient Monitor.

This document contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without prior written consent of Philips.

Philips Medizin Systeme Boeblingen Gmbh
Hewlett-Packard-Str. 2
71034 Böblingen
Germany

Trademark Information

Microsoft ® is a U.S. registered trademark of Microsoft Corp. Windows 2000® is a U.S. registered trademark of Microsoft Corp.

INFRARED DATA ASSOCIATION (IrDA) is a trademark of the Infrared Data Association in the USA and other countries.

2

Connecting to the Network

The Philips IntelliVue Series monitor uses a standard IEEE802.3 10BaseT (10MBit/s) Local Area Network interface for the Data Export Capability.

The Data Export Interface via LAN is not available when the IntelliVue monitor is connected to the Philips LAN (e.g. to the Philips Information Center central station). Only devices approved for use with the Philips network may be connected to the Philips LAN.

Connecting to the Network via a LAN Interface

The IntelliVue monitor connects to the network using a standard unshielded LAN cable with an RJ45 connector. The network cable must be plugged to the orange-framed LAN connector of the IntelliVue monitor.Note that for IntelliVue MP2/X2 the LAN connector is located on the external power supply.

WARNING In order to maintain the galvanic isolation of the IntelliVue monitor, it is essential that UTP (Unshielded Twisted Pair) LAN cables must be used to connect the IntelliVue monitor to other devices.

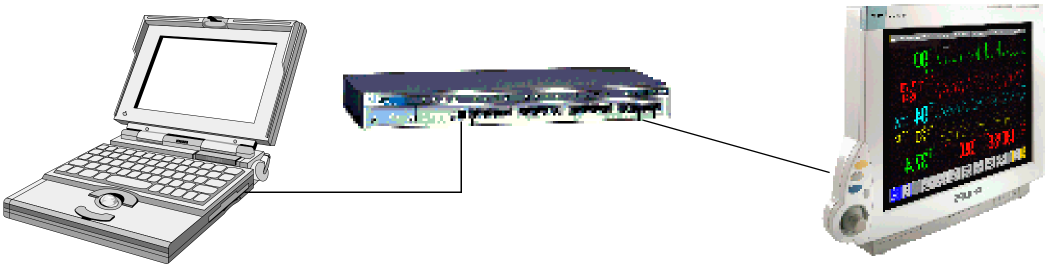
The following LAN cables supplied by Philips can be used to connect the IntelliVue monitor:

- M3199AI #J10 - 3ft (0.91m), Part No. M3199-60103 (12NC: 453563337391)
- M3199AI #J11 - 7ft (2.1m), Part No. M3199-60104 (12NC: 453563337401)
- M3199AI #J12 - 12ft (3.6m), Part No. M3199-60105 (12NC: 453563337411)

The maximum cable length between the IntelliVue monitor and the Computer Client should never exceed 330ft (100m) in total.

Connection via Hub/Switch

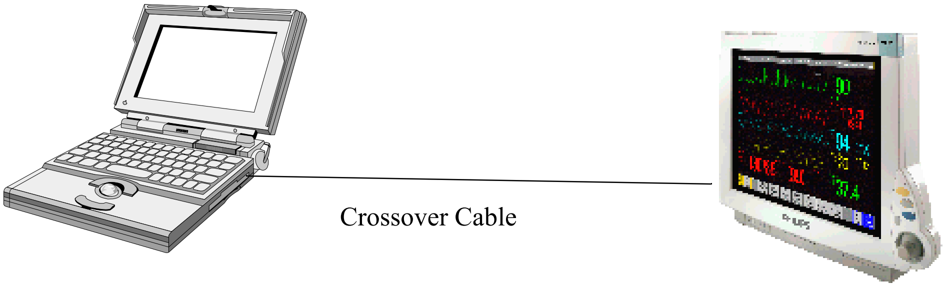
The IntelliVue monitor and the Computer Client are connected to a standard Ethernet switch or hub using UTP LAN cables.



NOTE In order to avoid high latency and data loss and to ensure data privacy, a dedicated network that is exclusively used for patient data collection by IntelliVue monitor devices and Computer Clients must be used.

Connection with Cross-over Cable

You can connect the IntelliVue monitor directly to the Computer Client, without a network hub or network switch, by using a UTP network crossover cable. In this case, the connection is a point-to-point connection only (one IntelliVue monitor connects to one Computer Client).



The following cross-over LAN cables supplied by Philips can be used to connect an IntelliVue monitor:

- M3199-60101 (453563337371) 3Ft UTP Crossover cbl Orange, 0,9m
- M3199-60102 (453563337381) 12Ft UTP Crossover cbl Orange, 3,6m

Avoiding Current Leakage

You must use Unshielded Twisted Pair (UTP) LAN cables to connect the IntelliVue monitor to other devices.

The Computer Client and network infrastructure devices typically are not classified as medical devices and must be located outside the patient vicinity. The patient vicinity is defined as an area within 6ft (1.85m) of the perimeter of the patient's bed or within 7.5ft (2.3m) of the floor.

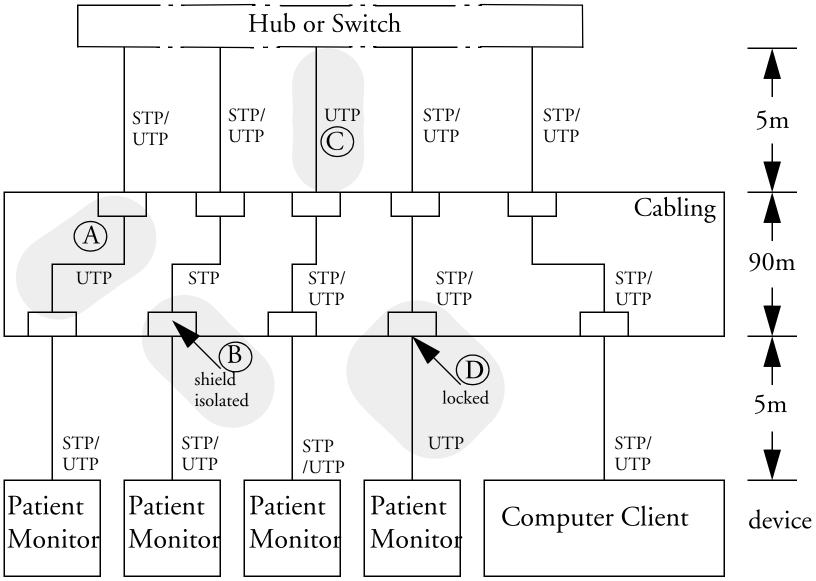
- If the Computer Client is installed in the patient vicinity and connected to the monitoring device, it must be correctly isolated from the mains power supply by an isolation transformer.
- If the Computer Client is installed in the patient vicinity and a network switch or hub is used to connect it to a monitoring device, it must be correctly isolated from the mains power supply by an isolation transformer.

WARNING All external devices in the patient vincinity must comply with IEC 60601-1:1988/A1:1991A2:1995 or EN 60601-1:1990/A1:1993/A2:1995. This applies also to all signal connections, entering the patient vicinity. Additional safety equipment, e.g. isolation transformers might be used.

The installation procedures e.g. for electrical connections as documented in the Instructions for Use must be strictly followed.

Using the Monitor with an Installed, Wired Network

The following diagram shows an overview of a possible LAN installation which provides galvanic isolation of the IntelliVue monitor:



If required by regulations valid in your hospital, the installation must comply to EN60601-1-1:1993/ A1:1996 or IEC 60601-1-1:1992/A1:1995.

The maximum cable length between the IntelliVue monitor and the Computer Client should never exceed 330ft (100m) in total.

Note regarding MP2 and X2:

The MP2 and X2 allow Data Export via the LAN interface only because they do not have an RS232 port. The LAN interface is only available if the MP2 or X2 is used in combination with the M8023A External Power Supply. If the X2 is connected to a host monitor, the data export can be performed via the host monitor.

IntelliVue Rev. G.0 or higher allows the combination of a LAN interface and one MIB/RS232 port for Data Export. Only one connection is able to request wave data at a time, the other connection responds with a notification that wave polling is not possible.The first connection to request a successful wave poll receives the wave data.

WARNING In order to maintain the galvanic isolation of the IntelliVue monitor, it is essential that the shield is not connected from the IntelliVue monitor through to the hub or switch. At least one of the following precautions must be taken:

- UTP (Unshielded Twisted Pair) LAN cables are used in the wall.
- If STP (Shielded Twisted Pair) LAN cables are used in the wall, do not connect the shield of the cable from the IntelliVue monitor to the wall socket. Ensure that the shield of the STP cable in the wall is isolated from the other contacts. For a reference voltage of 250V, a clearance of at least 2.5 mm and a creepage distance of at least 4.0 mm is required. Cutting the shield back and covering it with a nonconducting shroud will fulfill this requirement.
- Ensure that only UTP cables are used in the wiring closet for connections to the hub or switch.
- Use only UTP cables such as M3199AI #J10/J11/J12 to connect the IntelliVue monitor to the wall socket. To avoid these cables being replaced by non-UTP cables, the connector which goes into the wall socket must be modified so that it cannot be removed without using tools. This can be done by cutting off the part of the plug lock which normally extends beyond the socket.

Configuring the LAN Interface

Configuring the Network Address

No explicit configuration of the network addresses (IP addresses etc.) is required. The IntelliVue monitor uses the standard BootP protocol to acquire an IP address and subnet mask from a BootP server in the network. If you are using a DHCP server, make sure the server supports BootP clients.

Without a working BootP/DHCP server in the network, the IntelliVue monitor will show a technical alarm (INOP) “Unsupported LAN”, indicating that no (valid) IP address has been received.

NOTE

- For IntelliVue Software Revision E and later it is possible to manually enter the IP address used by the patient monitor by entering service mode and accessing the bed information window from the main setup menu.
- With IntelliVue Software Revision H and later, use of the DHCP protocol is also supported. This requires configuration of the network interface. See the IntelliVue Configuration Guide for details.

Configuring the LAN Data Export Setting

The data that can be exported via the LAN interface is configurable. You can choose between the following options: all, anonymous data, off. In case of anonymous data the patient name and given name are not included in the data stream.

To change the CentralMon configuration switch, first switch to configuration mode

To configure the LAN Data Export Setting, in Configuration Mode,

Select **Main Setup** to enter the Main Setup menu.

Select **Global Settings**

Select **LAN Data Export** and toggle the appropriate setting.

Configuring the Network Setting

The Central Monitoring setting on the IntelliVue monitor determines whether the monitor requires a connection to the Philips Information Center (central station). If **Central Monitoring** is set to **Mandatory**, the monitor issues a technical alarm (INOP) if a network is detected without an Information Center (central station). If you are connecting the IntelliVue monitor to a Computer Client, **Central Monitoring** should be set to **Optional**.

To do this, in Configuration Mode,

1 Select **Main Setup** to enter the Main Setup menu.

2 Select **Network**

3 Select **Central Monitoring** and toggle to the appropriate setting:

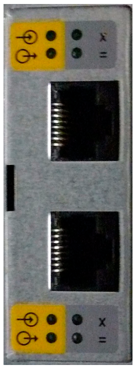
Mandatory	The IntelliVue monitor should be connected to an Information Center. An INOP is displayed if no connection is available.
Optional	The IntelliVue monitor can be connected to an Information Center. An INOP is only displayed if the connection to the Information Center is lost. No INOP is displayed if no connection is found at power on.

4 After the configuration, make sure you have stored all the active settings and leave Configuration Mode. You do not need a password to return to Monitoring Mode.

For further details on configuration, please refer to the IntelliVue configuration guide (M8000-9306X).

Connecting to the IntelliVue MP20-90 or MX Series MIB/RS232 Interface

The IntelliVue monitor MIB/RS232 interface provides an eight-pin RJ-45 modular jack. (MP70 and MX Series RS-232 connector shown as an example below).



MP20-90

MX Series

For the cable connection an eight conductor #24 American Wire Gauge (AWG) unshielded twisted-pair (UTP) cable must be used. The cable must follow ANSI/TIA/EIA-568-A-1995 Category 5 (CAT-5). The cable length must not exceed 65ft (20m). Straight-through pinning must be used.

The physical specification of the MIB/RS232 Interface follows the standard IEEE 1073.3.2. Refer to the standard for more information on cables and pin assignment.

The MIB/RS232 interface provides a RS232 port with the following pin assignment. This table is valid when the MIB/RS232 Interface is in DCC (Device Communication Controller) mode (DCC LED on the MIB/RS232 board is on - see below for details).

Computer Client	Pin and Signal Direction	IntelliVue monitor
	1<=	dDPWR
GND	4 <=>	GND
RxD	5 <=	TxD
TxD	7 =>	RxD

The pins of the RJ45 are counted from 1 for the lowest pin to 8 for the highest pin when looking at the RS232/MIB interface board.

LEDs on the MIB/RS232 Board (MP20-90 only)

There are four LEDs per port on the MIB/RS232 board which provide information on the configuration of the respective board. The MIB functionality is indicated by the LEDs in the yellow fields, other functionality (e.g. use for AGM or touch) is indicated by the LEDs in the grey fields. Only one LED is lit at a time.

LED	Meaning
yellow, arrow in	MIB BCC (Bedside Communication Controller) Mode
yellow, arrow out	MIB DCC (Device Communication Controller) Mode
grey, =	RS232 Mode, RX/TX lines straight
grey, X	RS232 Mode, RX/TX lines crossed

NOTE The drawings and descriptions of the RS232/MIB board above apply to the IntelliVue MP60/70 monitors. Location and orientation of the board may vary, depending on the monitor purchased.

Please note that Data Export will only function with the MIB/RS232 interface in DCC mode.

The TxD and RxD lines are the RS232 receive and transmit lines. The signals are referenced to the round (GND). The dDPWR can be used to power an external device with low power consumption. Refer to the Power Output specification in the table below.

Other applications in the IntelliVue monitor may be configured to use the MIB/RS232 Interface. These applications may use pins which are not used by the Data Export interface. Unused pins should not be connected. The IntelliVue monitor provides multiple RJ-45 connectors. Make sure, to use the correct connector with a port configured for Data Export.

The configuration of a specific MIB/RS232 port can be viewed in config mode and altered in **service mode**. To alter the configuration of an MIB port select **`Main Setup`** then **`Hardware`** then **`Interfaces.`** This brings up the MIB/RS232 card configuration. The port that you are using must be set to **`DtOut1`** for the "Data Out" function. If the MIB/RS232 port is configured for data export the yellow arrow out LED will be lit.

IntelliVue Rev. G.0 or higher allows the configuration of either a second MIB/RS232 port or the combination LAN interface and MIB/RS232 port for Data Export. The **`DtOut2`** driver is used to connect a second port to Data Export. Only one connection is able to request wave data at a time, the other connection responds with a notification that wave polling is not possible.The first connection to request a successful wave poll receives the wave data.

Parameter	Limit
Driver (TxD)	
Driver load output voltage (3 kOhm to 7 kOhm load)	5 V <= |Vout| <= 15 V
Driver open-circuit voltage	|Vout| <= 25 V
Driver short-circuit current (to +/- 15 V)	|Iosv| <= 100 mA
Receiver (RxD)	
Receiver input resistance	3 kOhm to 7 kOhm
Maximum receiver input voltage	+/- 25 V
Receiver threshold	+/- 3V
Power output (dDPWR)	
Minimum output voltage	4.75 V
Maximum output voltage	5.25 V
Minimum guaranteed output current	100 mA
Maximum typical output current	150 mA

REPEATED INFORMATION: If the Computer Client is not classified as a medical device, it must be located outside the patient vicinity. The patient vicinity is defined as an area within 6ft (1.85m) of the perimeter of the patient's bed or within 7.5ft (2.3m) of the floor.

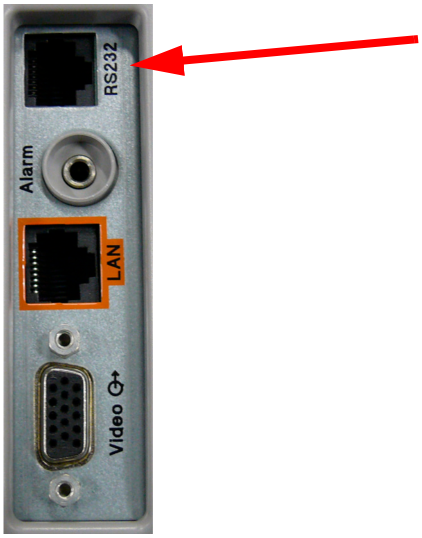
WARNING All external devices in the patient vincinity must comply with IEC 60601-1:1988/A1:1991A2:1995 or EN 60601-1:1990/A1:1993/A2:1995. This applies also to all signal connections, entering the patient vincinity. Additional safety equipment, e.g. isolation transformers might be used.

The installation procedures e.g. for electrical connections as documented in the User's Guide must be strictly followed.

If it is installed in patient vicinity, the Computer Client, connected to the instrument, must be correctly isolated from the mains power supply by an isolation transformer. The MIB/RS232 interface provides galvanic isolation of the IntelliVue monitor from a connected device.

Connecting to the Intellivue MP5 Monitor MIB/ RS232 Interface

The physical specification of the MP5 RS232 Interface follows the standard IEEE 1073.3.2. Refer to the standard for more information on cables and pin assignment. Note that the MP5 monitor's RS232 interface is always configured as a BCC device.



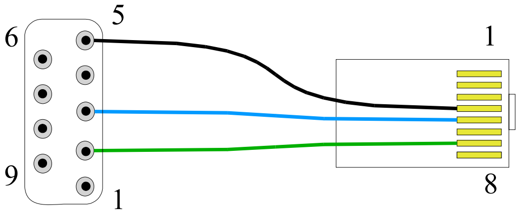
The MP5 RS232 interface provides an RS232 port with the following pin assignment.

Computer Client	Pin and Signal Direction	IntelliVue monitor
GND	4 <=>	GND
TxD	5 =>	RxD
RxD	7 <=	TxD

The TxD and RxD lines are the RS232 receive and transmit lines. The signals are referenced to the round (GND).

To connect a PC via RS232 to the MP5 monitor, use a cable configured as shown below.

The pins of the RJ45 connector are counted from 1 for the highest pin to 8 for the lowest pin looking directly at the pins with the cable leaving the connector to the left.



The MIB/RS232 port must be set to **DtOut1** for the "Data Out" function. See page 19 for details on how to change the configuration of the MIB/RS232 port.

REPEATED INFORMATION: If the Computer Client is not classified as a medical device, it must be located outside the patient vicinity. The patient vicinity is defined as an area within 6ft (1.85m) of the perimeter of the patient's bed or within 7.5ft (2.3m) of the floor.

WARNING All external devices in the patient vincinity must comply with IEC 60601-1:1988/A1:1991A2:1995 or EN 60601-1:1990/A1:1993/A2:1995. This applies also to all signal connections, entering the patient vincinity. Additional safety equipment, e.g. isolation transformers might be used.

The installation procedures e.g. for electrical connections as documented in the User's Guide must be strictly followed.

If it is installed in patient vicinity, the Computer Client, connected to the instrument, must be correctly isolated from the mains power supply by an isolation transformer. The MIB/RS232 interface provides galvanic isolation of the IntelliVue monitor from a connected device.

Configuring the IntelliVue Monitor MIB/RS232 Interface

The MIB/RS232 interface supports different transport protocols.To change the MIB/RS232 interface configuration, in Configuration Mode,

1 Select **`Main Setup`**

2 Select **`Hardware`**

3 Select **`Data Export`** and select the required setting:

AutoSpeed	Transport protocol with baudrate negotiation, based on the IrDA protocol.
Fix 19200	Transport protocol with a fixed baudrate of 19200 baud.
Fix 115200	Transport protocol with a fixed baudrate of 115200 baud.

4 Exit Configuration Mode. You do not need a password to return to Monitoring Mode.

IntelliVue Rev. G.0 and higher devices that allow two MIB/RS232 interfaces have two "Data Export" options. You can select **`DtOut1`** and/or **`DtOut2`**. These can be configured in service mode only. Please refer to the respective service guide for information on how to access service mode and teh required password.

For further details on configuration, please refer to the IntelliVue configuration guide (M8000-9306X).

3

Protocol Concept

The Protocol is based on a Client/Server Model. The Personal Computer *(Client)* maintains a logical connection with the Philips IntelliVue Series Patient Monitor (*Server*). Communication occurs by sending and receiving Command messages.

Supported Transport Protocols

The Data Export functionality in the IntelliVue monitor can be accessed via the LAN interface or via the MIB/RS232 interface. While the Association Control and Data Export Protocol is the same for both interfaces, the underlying transport protocol varies.

- For the LAN interface the transport protocol is the standard UDP/IP protocol.
- For the MIB/RS232 interface, two transport protocols are supported:
  - a fixed baudrate protocol at 19200 or 115200 baud and
  - a protocol with baudrate negotiation (Auto Speed) based on the IrDA protocol with a baudrate from 9600 baud to 115200 baud.

Association Control and Data Export Protocol		
UDP/IP	RS232 Fixed Baudrate	RS232 Auto Speed
LAN Interface	MIB/RS232 Interface	

UDP/IP Protocol

The transport protocol uses the Universal Datagram Protocol/ Internet Protocol (UDP/IP).The protocol is based on the Request For Comment (RFC) internet standard. UDP is defined in RFC 768; IP is defined in RFC 760.

The UDP/IP transport protocol is part of the internet protocol suite. Drivers and necessary hardware are available for all relevant computing platforms. It provides for a simple exchange of messages (Datagrams) across a Local Area Network. The maximum size of user data in a protocol message can be negotiated at connection time between the IntelliVue monitor and the Computer Client.

Fixed Baudrate Protocol

The Fixed Baudrate Protocol provides a transport protocol with minimal overhead and complexity. It is intended for Computer Clients which cannot use the Auto Speed Protocol. The protocol operates at a fixed baudrate and can be used with standard RS232 concentrators. It provides packet-oriented data exchange and checksum protection on top of the RS232 protocol. For the specification of the Fixed Baudrate Protocol see “Transport Protocols for the MIB/RS232 Interface” on page 30.

Auto Speed Protocol

The Auto Speed Protocol is based on the IrDA protocol. It offers a reliable transport layer with checksum protection and a retry mechanism in the case of transmission problems. The baudrate can be negotiated in a range from 9600 baud to 115200 baud. For the specification of the AutoSpeed Protocol see “Transport Protocols for the MIB/RS232 Interface” on page 30.

Protocol Model

The protocol is based on an object-oriented modelling concept. All information available through the Data Export Protocol is modelled as attribute values of information objects.

The following information object classes are supported by the IntelliVue monitor:

- Medical Device System (MDS)

  The MDS object contains attributes representing dynamic state information (e.g. current operating mode) and static device specific identification information (e.g. Serial Numbers).

- Alert Monitor

  The Alert Monitor object contains attributes representing the current technical and patient alarms, as e.g. displayed on the IntelliVue monitor.

- Numeric

  Numeric objects contain attributes representing the state and value of numerical measurements (e.g. Heart Rate).

- Waves

  Realtime sample array objects contain attributes representing the state and value of wave data (e.g. ECG).

- Patient Demographics

  The Patient Demographics object contains attributes representing patient information stored in the IntelliVue monitor (e.g. Patient Name).

The object attributes can be accessed by a poll of the MDS object, which allows a query of the sets of attribute values from all objects of a specified class.

The method can be called by sending a command message from a Computer Client to the IntelliVue monitor.

Protocol Dialog

The following diagram shows the protocol dialog between the IntelliVue monitor Data Export server and a Computer Client:

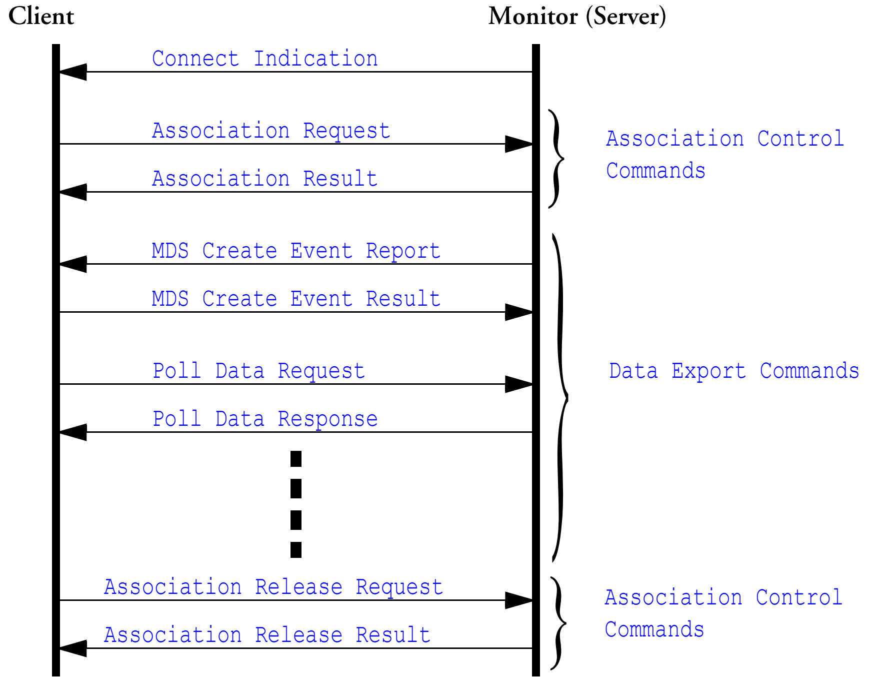


Figure 1 Protocol Dialog

Transport protocol-specific messages are not shown in the diagram. The Connect Indication message is only available on the LAN interface.

The Philips IntelliVue Series Patient Monitor processes global commands and sends response messages to the requests made by the client of the personal computer.

The messages shown in the diagram are explained in the following sections.

Connect Indication

The Connect Indication message is only sent on the LAN interface.

As soon as the IntelliVue monitor has received a valid IP address from the BootP server in the network, it sends out the Connect Indication message on its LAN interface. The message is a periodic subnet broadcast message that allows Computer Clients to find the IntelliVue monitor on the network. The message contains a set of device-related information, e.g., serial numbers, network addresses, internal states.

The IntelliVue monitor resends the Connect Indication message until a logical connection to a central station has been established. The IntelliVue monitor uses the retransmit strategy described in RFC 951. The resend period starts with 4 seconds and is doubled with each resend. The maximum resend period is about 64 seconds. The actual resend period contains a random component to avoid network congestion e.g. after a power failure.

Association Request

To establish a logical connection, the Computer Client sends the Association Request message to the IntelliVue monitor.

The Association Request can be used to set optional features of the logical connection between Computer Client and IntelliVue monitor.

Association Result

The IntelliVue monitor processes the Association Request and sends an Association Result. The result can be either a refuse message or an accept message.

The Computer Client must parse the Association Result to find out which protocol features can be used for this association.

MDS Create Event Report

If the IntelliVue monitor accepts the association, it sends a MDS Create Event Report after the positive Association Result message.

The MDS Create Event Report contains information about the system and its configuration.

MDS Create Event Result

The Computer Client must confirm the reception of the MDS Create Event Report. If the IntelliVue monitor does not receive a MDS Create Event Result message, the association is aborted.

Poll Data Request

After establishing an association, the Computer Client can send Poll Data Requests to access the data within the IntelliVue monitor.

The Poll Data Request contains a data-type parameter, which defines the specific type of requested data. The following data types are supported:

- Numeric Measurements
- Wave data
- Alerts (patient alarms and technical alarms)
- Patient Demographics
- System Attributes (e.g. dynamic state information, serial numbers, versions, etc.)

Only one type of data can be accessed per Poll Data Request.

Poll Result

Depending on the status of the IntelliVue monitor and the options set during the establishment of the logical connection (association phase), a Poll Data Request message can return:

- a single Poll Data Reply
- multiple, linked Poll Data Replies, if the size of the requested data exceeds the maximum size of a transport layer message
- a continuous number of periodic Poll Data Replies for a time period defined by the Computer Client (supported for Numeric Measurements, Waves and Alerts only).

Association Release Request

When the Computer Client wants to close an association, it can send a Release Request.

Association Release Result

The IntelliVue monitor parses the Release Request. If the Release Request is syntactically correct, the IntelliVue monitor sends an Association Release Result, indicating that the Association has been released.

Association Abort

In the case of communication problems, such as time-out, the IntelliVue monitor can send an Association Abort message. This message indicates that the association has been closed. A Computer Client should use the Association Release Request which provides a confirmation.

More Information

- For more details on the association control commands, such as Association Request, Association Result, Association Abort etc., please refer to the section “Definition of the Association Control Protocol” on page 65.
- For more detail on the data export commands, such as Poll Data Request, MDS Create Event Report, MDS Create Event Result, etc., please refer to the section “Definition of the Data Export Protocol” on page 35.

Connection Time-out Mechanism

The IntelliVue monitor automatically closes the connection if it detects a connection time-out condition. The connection time-out value is derived from the minimum poll period that is negotiated during the connection establishment phase.

A connection time-out period is 3 times the negotiated minimum poll period time. However, the minimum connection time-out is 10s, the maximum connection time-out period is 130s.

If the IntelliVue monitor does not receive a protocol message within the connection time-out period, the device closes the connection to the Computer Client by sending an Association Abort message. After that, a new connection can be established from the Computer Client to the IntelliVue monitor.

Network Load Consideration

Input Data

The IntelliVue monitor accepts a specific amount of input data per association. If the Computer Client sends more than the specified number of messages, the IntelliVue monitor will discard messages to avoid an unreasonably high system load. A Computer Client should be able to handle the loss of messages.

Message Type	Messages per Second
Association Control	1
Poll Request - Numerics (observed values)	1
Poll Request - Numerics (other attributes)	1
Poll Request - Enumerations	1
Poll Request - Waves	1
Poll Request - Alert Monitor	1
Poll Request - Patient Demographics	1
Poll Request - Medical Device System	1

The IntelliVue monitor will send a Remote Operation Error message if it receives a poll request for an object while it is still processing another poll request for the same object.

Output Data

The IntelliVue monitor processes the received message and sends the corresponding results. In rare cases, it can take up to several seconds until the response message is returned, and Poll Requests may be lost.

To avoid poll requests or poll responses getting lost, it is strongly recommended that the Computer Client uses the extended poll method to poll real-time numerics.

Definition of the Transport Protocols

Transport Protocols for the LAN Interface

UDP/IP

The Protocol uses the Universal Datagram Protocol/ Internet Protocol (UDP/IP) as the transport protocol. The protocol is based on the following internet standards (Request For Comment, RFC):

UDP is defined in RFC 768.

IP is defined in RFC 760.

The UDP/IP transport protocol is part of the internet protocol suite. Drivers and necessary hardware are available for all relevant computing platforms.

It provides for a simple exchange of messages (Datagrams) across a Local Area Network.

The maximum size of user data in a protocol message can be negotiated at connection time between the IntelliVue monitor and the Computer Client.

The upper limit for the negotiated user data size (MTU, Maximum Transport Unit) is 1364 bytes, the lower limit for the negotiated MTU is 500 Bytes. The maximum size of a UDP message sent by the IntelliVue monitor is 1380 bytes.

IP Address

The IP Address and the subnet mask necessary for communicating with the IP Protocol is set using the BootP protocol defined in the Internet RFC 951.

In order to communicate with the Philips IntelliVue Series Patient Monitor, a BootP server must exist in the network. The BootP server must be configured so that it answers BootP Request messages from the IntelliVue monitor.

UDP Port Number

The UDP Port Number used by the IntelliVue monitor for the Protocol can be extracted from the Connect Indication broadcast message used for Device Discovery (see “CONNECT INDICATION EVENT” on page 53). The current Protocol version uses the fixed UDP port 24105.

All messages sent from the Computer Client to the IntelliVue monitor must use this port number as the destination port number.

The Computer Client can chose any available source port for the communication. Once the Computer Client has chosen a source port, it must not use any other port. Protocol messages from another source port will be regarded as messages from a different Computer Client).

Any messages sent from the IntelliVue monitor back to the Computer Client use the source port number set by the Computer Client in first message (the Association Request message, see “Association Request Message” on page 67).

Transport Protocols for the MIB/RS232 Interface

The Fixed Baudrate Protocol, RS232 Port Settings

Each transmitted byte consists of one start bit, 8 data bits (no parity) and one stop bit. The baudrate can be set to 115kBit/s or 19.2kBit/s.

Flow control is not supported (same behavior as UDP). The monitor limits the number of Frames which will be processed in a given time. The monitor will process up to 4 frames within 128ms. If a client sends more frames, additional frames are ignored. (Implementation Note: the monitor allows 5 frames within 128ms, the additional frame is required because of possible jitter.)

A client system must be able to handle the loss of messages, because the Fixed Baudrate Protocol does not guarantee the reliable transmission of messages.

Framing

BOF	Hdr	User Data	FCS	EOF

The framing structure is the same as for AutoSpeed protocol. A frame starts with a single BOF.

BOF	Beginning Of Frame (0xC0)
Hdr	Header Information
User Data	Association Control or Data Export Command message
FCS	16 bit Frame Check Sequence using CRC-CCITT algorithm
EOF	End Of Frame (0xC1)

Header Information

The *Hdr* field is defined as follows:

```
typedef struct
{
    u_8            protocol_id;
    u_8            msg_type;
    u_16           length;
} FrameHdr;
```

The *protocol_id* field contains ID and version information. It can be used to define different service access points. Data Export uses the ID 0x11.

The *msg_type* field defines the type of message which is being sent. The value 0x01 indicates an Association Control or Data Export Command message, future message types could be used for flow control, lifetick, message confirmation etc.

The *length* field contains the length of the appended user data in bytes (without transparency characters).

If a client receives messages with an unknown *protocol_id* or *msg_type*, it should ignore the message.

Frame Check Sequence Field

The Frame Check Sequence Field can be used to detect transmission errors. The field contains a 16 bit CRC-CCITT cyclic redundancy check (not the popular XMODEM variation of CRC-CCITT). The CRC is computed from the *Hdr* and *User Data* field. Refer to "Serial Infrared Link Access Protocol (IrLAP)" Version 1.1 for the actual computation method of the CRC. A code snippet for the FCS algorithm can be found in the Network Working Group Request for Comment: 1171 (PPP protocol). The one's complement of the CRC is transmitted, rather than the CRC itself. The CRC is transmitted LSB first.

If the CRC is not correct, a client system should ignore the message.

Transparency

The contents of the *Hdr* and *User Data* fields is unrestricted. This can lead to problems if a BOF or EOF character appear in the *Hdr*, User Data, or FCS field. A Control Escape byte is defined as 0x7D. The sender must examine each byte in the User Data and FCS fields; for each byte with the value 0xC0, 0xC1, 0x7D it does the following:

- insert a 0x7D byte proceeding the byte
- complement bit 5 of the byte (XOR with 0x20).

Frame Abort

The sending station may abort the transmission of a frame by sending a control escape character followed by a EOF character (0x7D 0xC1) without sending the FCS field.

Examples The examples below do not include the *Hdr* field. For a correct message, the framing algorithm must be applied to the *Hdr* and *UserData* field of the message.

1 If a Computer Client wants to send the data:

“0x3a 0x71”

The CRC for this data would be:

“0xd9 0x64”

after building the one's-complement and byte-swapping, this results in:

“0x9b 0x26”

The whole frame would be:

“0xc0 0x3a 0x71 0x9b 0x26 0xc1”

2 If a Computer Client wants to send the data:

“0x3a 0x91”

The CRC for this data would be:

“0x3e 0x6a”

after building the one’s-complement and byte-swapping, this results in:

“0x95 0xc1”

The whole frame would be:

“0xc0 0x3a 0x91 0x95 0x7d 0xe1 0xc1”

Note that byte “0xc1” in the CRC is a reserved character and must be escape. This results in “0x7d 0xe1”.

The AutoSpeed Protocol

The AutoSpeed Protocol follows the definition of the Transport Protocol defined in the standard IEEE 1073.3.2: IEEE Standard for Medical Device Communications - Transport Profile - IrDA Based Cable Connection.

For a description of the IrDA Protocol refer to the specifications of the Infrared Data Association (www.irda.org):

- IrDA, Serial Infrared Link Access Protocol (IrLAP), Version 1.1, June 16, 1996
- IrDA, Link Management Protocol (IrLMP), Version 1.1, Oct. 20, 1996
- IrDA, Tiny TP: A Flow-Control Mechanism for use with IrLMP, Version 1.1, Oct. 20, 1996

Commercial IrDA stacks are available for most operating systems. Some operating systems, like Microsoft® Windows 2000® and Linux, come with an off-the-shelf IrDA stack.

The Data Export protocol resides as a packet oriented client on top of the IrDA TinyTP layer.

Establishing a Connection

A connection is created using the following steps:

- Discovery

The Computer Client sends an IrLAP discovery request to find out if a device is physically connected. The IntelliVue monitor answers with an discovery response message. The discovery procedure is done at a fixed baudrate of 9600 baud.

- Open an IrLAP connection

When the Computer Client finds a connected system, it can send an IrLAP Set Normal Response Mode message to establish a logical IrLAP connection. The IntelliVue monitor sends an response message. During this procedure parameters of the IrLAP connection, like baudrate, data size, etc. are negotiated.

- Open an IAS port

The Information Access Service (IAS) is provided by the IrLMP layer. It provides a database with device information which can be queried by the client. Before accessing the service, the client must connect to the special IrLMP service access point (SAP) 0.

- Perform an IAS query

The IrLMP layer does not specify a well-known SAP for the Data Export Protocol, hence the client should query the IAS database to find the SAP for the Data Export Protocol. The database contains the attribute “IrDA:TinyTP:LsapSel” under the object class “IEEE:1073:3:2:MDDL”. The attribute specifies the SAP for the Data Export Protocol on the IrDA TinyTP layer as an integer value.

- Close the IAS port

After performing the IAS query, the Computer Client should close the IAS port again with an IrLMP disconnect message.

- Open a Tiny TP connection

After retrieving the number for the TinyTP SAP, the client system can open a connection on this SAP. This is done with an IrLMP connect request message which contains a TinyTP connect in its user data.

- Send an Association Request

After the transport layer connection has been established, the Computer Client can send an Association Request message to start a Data Export session.

- Send a Release Request

When the client has no need for further communication, it can send a Release Request message to terminate the Data Export session.

- Close the IrLAP connection

After the Data Export session has been closed, theComputer Client should also close the TinyTP SAP. This can be done by sending an IrLMP disconnect message or by closing the whole IrLAP connection.

5

Definition of the Data Export Protocol

Definitions Shared by Protocols

Byte Order

The protocol data structures use the Network Byte Order. This means that bytes of a multi-byte data structure are transmitted on the network with the most significant byte first (as in big-endian data storage). This may or may not match the order in which numbers are normally stored in memory for a particular processor.

If the Computer Client is not using big-endian storage internally (many common Personal Computer Platforms use little-endian storage), protocol data structures (message structures) must be transformed before they are sent to an IntelliVue monitor or after they have been received from an IntelliVue monitor.

Byte Alignment

The Association Control and Data Export protocols assume no data alignment. However, most data types used in this guide have an even length for performance reasons. Many compilers use different alignment modes by default. Make sure that the compiler uses the right alignment when parsing and formatting protocol messages.

Bit Order

The index for bits starts with zero for the most significant bit.

MSB														LSB	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Common Data Types

Basic Data Types

The C data types defined here make use of the following basic types:

```
u_8      unsigned 8  bit wide integer
u_16     unsigned 16 bit wide integer
u_32     unsigned 32 bit wide integer
i_8      signed 8  bit wide integer
i_16     signed 16 bit wide integer
i_32     signed 32 bit wide integer
```

The mapping of these types to data types used in a Computer Client application is machine specific and compiler dependent.

Absolute Time

The Absolute Time data type is used whenever data is time stamped and a resolution of 1s is sufficient.

```
typedef struct {
        u_8             century;
        u_8             year;
        u_8             month;
        u_8             day;
        u_8             hour;
        u_8             minute;
        u_8             second;
        u_8             sec_fractions;
} AbsoluteTime;
```

The individual u_8 fields are BCD encoded, they are not encoded as regular integer values. E.g. the year 99 (decimal) is coded as 0x99. An invalid time is marked with 0xff in all positions.

Note that the time resolution in IntelliVue monitor with this format is 1 second. The *sec_fractions* element in the structure is not used.

Relative Time

The Relative Time is a high resolution time marker which defines a time relative to an event (e.g. power-on). It is used to position events (a particular event message) relative to each other with a higher resolution. It is defined as follows:

```
typedef u_32     RelativeTime;
```

The resolution of the *RelativeTime* is 1/8ms (125us). The IntelliVue monitor sets the Relative Time with a precision of 2 ms. The Computer Client can calculate the absolute time (wall clock) from a known relation between Absolute Time and Relative Time with a precision of about 1s. For more information on the time mapping refer to “MDS CREATE EVENT” on page 54.

OID Type

For the identification of all protocol elements (e.g. physiological meaning, alert codes, units of measure), the *OIDType* (Object Identifier Type) is used.

```
typedef u_16            OIDType;
```

Values for the *OIDType* (the nomenclature) are listed at the end of the section “Attribute Data Types and Constants Used” on page 75. Independent value ranges (partitions) exist, e.g. for physiological identifiers, alert condition identifiers, units of measurement etc.

Private OID

For the identification of private or manufacturer specific elements, a special type is used.

```
typedef u_16            PrivateOID;
```

Values for the *PrivateOIDs* are listed whenever a *PrivateOID* is used. Refer to the section “Attribute Data Types and Constants Used” on page 75 for a complete list of identifiers.

TYPE

Whenever it is not clear from the context, from which nomenclature value range the *OIDType* comes, the TYPE data type is used. Here, the nomenclature value range (the partition) is explicitly identified.

```
typedef u_16            NomPartition;
#define NOM_PART_OBJ          1
#define NOM_PART_SCADA        2
#define NOM_PART_EVT          3
#define NOM_PART_DIM          4
#define NOM_PART_PGRP         6
#define NOM_PART_INFRASTRUCT  8

typedef struct {
        NomPartition    partition;
        OIDType         code;
} TYPE;
```

The *code* values are grouped in the following partitions:

NOM_PART_OBJ: Object oriented element, device nomenclature

NOM_PART_SCADA: Types of measurement and place of the measurement

NOM_PART_EVT: Codes for alerts

NOM_PART_DIM: Units of measurement

NOM_PART_PGRP: Identification of parameter groups

NOM_PART_INFRASTRUCT: Infrastructure for Data Export applications

The *code* is only unique in a given partition. The values for the *OIDType* are defined in the section “Attribute Data Types and Constants Used” on page 75.

Handle

Object instances, e.g. Numeric object instances, are identified with a 16bit wide ID, the object Handle:

```
typedef u_16            Handle;
```

Global Handle

Handles are unique within the context of a particular system. The Protocol supports multiple measurement servers, where each measurement server assigns object handles independently. To assure handle uniqueness across system boundaries, the Global Handle contains an additional identifier for the source system, e.g., each measurement server has a distinct context id. The context id is assigned dynamically when a measurement server is connected.

```
typedef u_16            MdsContext;

typedef struct {
       MdsContext      context_id;
       Handle          handle;
} GlbHandle;
```

Managed Object Identifier

The Managed Object Identifier is a fully qualified object identifier which contains an identifier for the object class (e.g. Numeric object) together with a Global Handle.

```
typedef struct {
       OIDType        m_obj_class;
       GlbHandle      m_obj_inst;
} ManagedObjectId;
```

Attribute Value Assertion

Object attributes are represented in the form of data record structures which contain an identifier for the attribute, a length field for parsing and the actual value of the attribute.

The structure of such an attribute record is the Attribute Value Assertion, which is defined as follows:

```
typedef struct {
       OIDType        attribute_id;
       u_16           length;
       u_16           attribute_val;
} AVAType;
```

The *attribute_id* identifies the type of the attribute. The length field contains the size of the *attribute_val* field in bytes. The *attribute_val* field itself is only a placeholder in this structure. The parsing algorithm must assign the attribute value to the correct data structure based on the value of the *attribute_id*.

Attribute List

Typically, object instances have multiple attributes which are captured in a list with the following data type:

```
typedef struct{
        u_16           count;
        u_16           length;
        AVAType        value[1];
} AttributeList;
```

The count field contains the number of Attribute Value Assertion elements in the list.

The length field contains the size of the list (the value array) in bytes.

The value field itself again is only a placeholder data structure. A parser must be used to interpret the data structure. Refer to “Protocol Examples” on page 291 for an example of an *AttributeList*.

String

The text string is preceded by a length field, followed by the value. The *length* field denotes the number of octets in *value*. If the length is zero, no octets are appended. The *String* data type is used for UNICODE encoded texts.

```
typedef struct {
    u_16            length;
    u_16             value[1];
} String;
```

Where possible, the real string lengths have been included in this document. However, these string lengths may change in future releases, producing discrepancies between the actual string lengths and this document.

The *String* uses the same language as the IntelliVue monitor. The IntelliVue monitor uses UNICODE for the *String* data type (see “Connect Indication Attributes” on page 107). The *String* may contain code values from the UNICODE private use area (0xE000 to 0xF8FF). The Computer Client most likely will not support these characters. The following codes are frequently used:

```
#define SUBSCRIPT_CAPITAL_E_CHAR     0xE145
        /* SUBSCRIPT CAPITAL E                        */
#define SUBSCRIPT_CAPITAL_L_CHAR     0xE14C
        /* SUBSCRIPT CAPITAL L                        */
#define LITER_PER_CHAR               0xE400
        /* LITER PER - used in 4 char unit "l/min"    */
#define HYDROGEN_CHAR                0xE401
        /* HYDROGEN - Used in 4 char unit "cmH2O"     */
#define ALARM_STAR_CHAR              0xE40D
        /* ALARM STAR "*"                             */
#define CAPITAL_V_WITH_DOT_ABOVE_CHAR 0xE425
        /* CAPITAL_V_WITH_DOT_ABOVE (V with dot)      */
#define ZERO_WIDTH_NO_BREAK_SPACE_CHAR0xFEFF
        /* The character 0xFEFF is used as FILL character.
        For each wide asian character, a FILL character is
        appended for size calculations. */
```

Variable Label

The string is preceded by a length field, followed by the value. If the length is zero, no octets are appended. The *VariableLabel* data type uses 8 bit ASCII encoding for the text. The *length* of a *VariableLabel* is always even.

```
typedef struct {
       u_16    length;
       u_8     value[1];
} VariableLabel
```

Where possible, the real string lengths have been included in this document. However, these string lengths may change in future releases, producing discrepancies between the actual string lengths and this document.

TextId

The *TextId* type is a 32bit wide private ID.

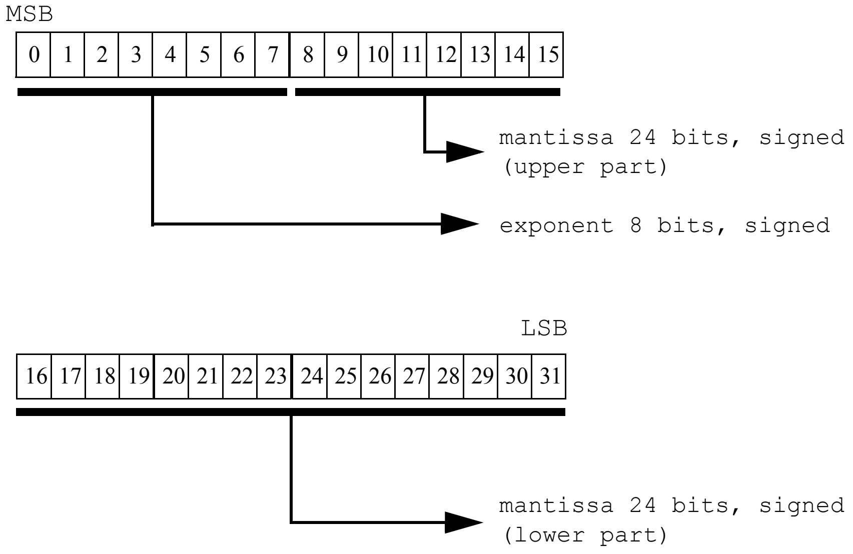
```
typedef u_32   TextId;
```

FLOAT-Type

For floating point numbers, a special 32bit wide format is used. For message parsing and for the definition of the message structures a 32bit wide placeholder structure is defined here.

```
typedef u_32   FLOATType;
```

The FLOAT-Type must be interpreted as follows:



The number represented is (mantissa)*($10^{exponent}$). Both the exponent and mantissa are in 2's complement form. The mantissa is not necessarily normalized.

There are four special values of the mantissa that can be represented:

NaN (Not a Number), which has a mantissa of +(2^{23} -1) (0x7fffff)

NRes (Not at this resolution), which has a mantissa of -(2^{23}) (0x800000)

+/- INFINITY, which have mantissa of +/- (2^{23}-2) (0x7ffffe, 0x800002).

The exponent is not important in these cases. This leaves the following ranges for normal number representation:

-128 <= exponent <= 127

-(2^{23}-3) <= mantissa <= +(2^{23}-3)

Definition on the number of the valid digits for the presentation on the IntelliVue monitor's display:

1.) If the exponent < 0, then the integer value of the exponent shows the number of valid digits after the point:

Examples:

value = 0xfd007d00:	exponent = -3,	mantissa = 32000	→	32.000
value = 0xff000140:	exponent = -1,	mantissa = 320	→	32.0

2.) If the exponent >= 0, then the number of valid digits after the point is zero.

Examples:

value = 0x01000140:	exponent = 1,	mantissa = 320	→	3200
value = 0x02000020:	exponent = 2,	mantissa = 32	→	3200

Protocol Command Structure

Protocol Command messages, as defined in this section, are the data structures that are transported within the transport layer message (UDP datagram, IrDA message or Fixed Baudrate Protocol message). The generic structure is common for messages sent from the Computer Client to the IntelliVue monitor (e.g. Poll Request messages) and messages sent from the IntelliVue monitor to the Computer Client (e.g. Poll Result messages).

The Protocol Command messages represent the ISO/ OSI layers 5 - 7 (session layer, presentation layer, application layer). The message that transports a Protocol Command contains a checksum. Computer Clients should validate this checksum to detect corrupted messages.

The Protocol command messages used to establish the logical connection (association) between the IntelliVue monitor and a Computer Client follow the definitions of the ACSE Standard (ISO/IEC 8649 and ISO/IEC 8650).

For the Protocol Commands during the logical connection, the message structure is layered and has the following basic format:

Session/Presentation Header
Remote Operation Header
Command Header
Command- Specific Parameter Data

The Session Header and Presentation Header are small fields only which contain fixed values for the life time of the logical connection between the IntelliVue monitor and the Computer Client.

The Remote Operation Header allows to distinguish between the different types of command messages, command response messages and error messages.

The Command Header contains the common part of the Command data structure identified in the Remote Operation Header.

Command-specific parameters or data are appended to the generic message structure.

Session/Presentation Header

Each protocol message starts with a common data structure representing the session and presentation protocol, defined as follows:

```
typedef struct {
    u_16    session_id;    /* contains a fixed value 0xE100 */
    u_16    p_context_id;  /* negotiated in association phase */
} SPpdu;
```

- session_id

  This field identifies a Protocol message. The field contains a fixed value 0xE100. Conceptually, this field represents the session header.

- p_context_id

  The presentation context identifier is negotiated during the exchange of the association messages.

The Computer Client can use the first byte of the *session_id* to distinguish between Data Export protocol commands and Association Control protocol commands.

If a Computer Client encodes the Association Control protocol commands as suggested in “Definition of the Association Control Protocol” on page 65, the *context_id* for the Data Export protocol commands is 2.

Remote Operation Header

A protocol message is considered a remote operation. There are different types of operations as defined below. The different operations are described by a common operation header data structure:

```
typedef struct {
    u_16            ro_type;            /* ID for operation */
#       define      ROIV_APDU 1
#       define      RORS_APDU 2
#       define      ROER_APDU 3
#       define      ROLRS_APDU 5
    u_16            length;             /* bytes to follow */
} ROapdus;
```

- ro_type

  This field defines which type of remote operation is appended.

  The following remote operation types exist:

  Remote Operation Invoke (ROIV_APDU) invokes (calls) a remote operation.

  Remote Operation Result (RORS_APDU) returns the result of a remote operation

  Remote Operation Error (ROER_APDU) returns an error for a remote operation.

  Remote Operation Linked Result (ROLRS_APDU) returns parts of the result of a remote operation. It is used when the size of the complete result exceeds the maximum size of one message.

- length

  This field defines the remaining number of bytes in the message.

Remote Operation Invoke

A Remote Operation Invoke message is defined as follows:

```
typedef struct {
    u_16    invoke_id;      /* identifies the transaction */
    CMDType command_type;   /* identifies type of command */
    u_16    length;         /* no. of bytes in rest of message */
} ROIVapdu;
```

- invoke_id

  The invoke identifier is used to reference the specific operation while it is being processed. Result messages or error messages will use this identifier as a reference. Therefore, the invoke identifier should be unique while the operation transaction is in process.

- command_type

  The command type identifier defines what command data type is appended to this structure.

- length

  This field defines the remaining number of bytes in the message.

Remote Operation Result

A Remote Operation Result message is a response to an Operation Invoke message requiring confirmation.

The message is defined as follows:

```
typedef struct {
    u_16     invoke_id;     /* mirrored back from op. invoke */
    CMDType  command_type;  /* identifies type of command */
    u_16     length;        /* no of bytes in rest of message */
} RORSapdu;
```

- invoke_id

  The invoke identifier is mirrored back from the related Remote Operation Invoke message that triggered this result. This field allows to relate the response message to the original request.

- command_type

  The command type identifier defines what command data type is appended to this structure.

- length

  This field defines the remaining number of bytes in the message. This length is not larger than the negotiated Maximum Transport Unit (MTU). For larger messages, the Remote Operation Linked Result mechanism will be used.

Remote Operation Linked Result

In some cases, the total data that must be returned as a result of a command may exceed the maximum message size. In these cases, multiple Remote Operation Linked Result messages are used.

These messages are defined as follows:

```
typedef struct {
    RorlsId linked_id;    /* see below */
    u_16    invoke_id;    /* see below */
    CMDType command_type; /* identifies type of command */
    u_16    length;       /* no of bytes in rest of message */
} ROLRSapdu;
```

- linked_id

  The linked identifier identifies each Remote Operation Linked Result message in a sequence of linked messages (see below).

- invoke_id

  The invoke identifier is mirrored back from the related Remote Operation Invoke message that triggered this result. This field allows to relate the response message to the original request.

- command_type

  The command type identifier defines what command data type is appended to this structure.

- length

  This field defines the remaining number of bytes in the message.

If the size of the result data exceeds the maximum message size, a combination of Remote Operation Linked Result Messages and Remote Operation Result messages is used, with the following rules:

- For all response messages except the very last one:
  - the ROLRS_APDU message type is used
  - the linked identifier is set by the responder to the *RorlsId* data type
  - the invoke identifier is the value of the invoke identifier of the associated Operation Invoke
- For the very last message:
  - The RORS_APDU message type is used

The invoke identifier in this response is the value of the invoke identifier of the associated Operation Invoke.

The following data type is used for the linked identifier:

```
typedef struct {
    u_8     state;
#     define RORLS_FIRST 1   /* set in the first message */
#     define RORLS_NOT_FIRST_NOT_LAST 2
#     define RORLS_LAST 3     /* last RORLSapdu, one RORSapdu
                                 to follow */
    u_8     count;            /* counter starts with 1 */
} RorlsId;
```

The first Remote Operation Linked Result message sets the state RORLS_FIRST.

The last Remote Operation Linked Result message sets the state RORLS_LAST. Note that there is one more Remote Operation Result message to follow.

All other Remote Operation Linked Result messages set the state RORLS_NOT_FIRST_NOT_LAST.

Examples:

- If a total of 3 messages are needed, the first message is a Remote Operation Linked Result with state RORLS_FIRST and count field 1.The second message is a Remote Operation Linked Result with state RORLS_LAST and count field 2.The third message is a Remote Operation Result message.
- If a total of 2 messages are needed, the first message is a Remote Operation Linked Result with state RORLS_LAST and count field 1. The second message is a Remote Operation Result message.

The *count* field starts with 1 for the first of the linked messages and is increased with each following message.

When a message is split, each message contains a full command data structure (see “Command Header” on page 47).

If the messages contain data from several objects, the Computer Client can not assume that all data belonging to one object is sent within one message. In some cases it can happen that the data belonging to one attribute of a given object must be sent in multiple messages (see the description of the available data in the section “Attribute Data Types and Constants Used” on page 75). This may only occur for attributes which are encoded in the form of a list (e.g Device T-Alarm List).

Object data which did not fit in one message is guaranteed to continue in the next linked message.

Remote Operation Error

If an error is detected at the Remote Operation level, an error message is returned:

```
typedef struct {
    u_16     invoke_id;
    u_16     error_value;
#      define      NO_SUCH_OBJECT_CLASS          0
#      define      NO_SUCH_OBJECT_INSTANCE       1
#      define      ACCESS_DENIED                 2
#      define      GET_LIST_ERROR                7
#      define      SET_LIST_ERROR                8
#      define      NO_SUCH_ACTION                9
#      define      PROCESSING_FAILURE            10
#      define      INVALID_ARGUMENT_VALUE        15
#      define      INVALID_SCOPE                 16
#      define      INVALID_OBJECT_INSTANCE       17
    u_16     length;
} ROERapdu;
```

- invoke_id

  The invoke identifier is mirrored back from the related Remote Operation Invoke message that triggered this result. This field allows to relate the response message to the original request.

- error_value

  The error values have the following meaning:

  GET_LIST_ERROR: Get operation failed. A *GetListError* is appended to the message.

  SET_LIST_ERROR: Set operation failed. A *SetListError* is appended to the message.

  NO_SUCH_ACTION: Unknown action type. The object class ID and action type are appended to the message.

  NO_SUCH_OBJECT_CLASS: There is no such object class in the system. An OIDType with the class ID is appended to the message.

  NO_SUCH_OBJECT_INSTANCE: The object instance does not exist. The *ManagedObjectId* of the instance is appended.

  ACCESS_DENIED: Computer Client has not required privileges to perform the operation. No data is appended.

  PROCESSING_FAILURE: Generic error indicating an invalid request. A *ProcessingFailure* is appended to the message.

  INVALID_ARGUMENT_VALUE: The argument of the ROSE message was not valid. An Action result is appended.

  INVALID_SCOPE: The scope is not valid for the operation. The value of the scope is appended.

  INVALID_OBJECT_INSTANCE: Wrong object instance. The *ManagedObjectId* of the instance is appended.

- length

  This field defines the remaining number of bytes in the message.

  The *GetListError* and *SetListError* structures are defined as follows:

```
typedef struct {
        ManagedObjectId       managed_object;
        struct {
                u_16                  count;
                u_16                  length;
                GetError              value[1];
        } getInfoList;
} GetListError;

typedef struct {
        ErrorStatus           errorStatus;
        OIDType               attributeId;
} GetError;

typedef struct {
        ManagedObjectId       managed_object;
        struct {
                u_16                  count;
                u_16                  length;
                SetError              value[1];
        } setInfoList;
} SetListError;
typedef struct {
        ErrorStatus           errorStatus;
        ModifyOperator        modifyOperator;
        OIDType               attributeId;
} SetError;

typedef u_16          ErrorStatus;
#define ATTR_ACCESS_DENIED            2
#define ATTR_NO_SUCH_ATTRIBUTE        5
#define ATTR_INVALID_ATTRIBUTE_VALUE  6
```

```
    #define ATTR_INVALID_OPERATION        24
    #define ATTR_INVALID_OPERATOR         25
```

The *ProcessingFailure* is defined as follows:

```
typedef struct {
    OIDType         error_id;
    u_16            length;
} ProcessingFailure;
```

Additional data with error information can be appended to the *ProcessingFailure*. The default *error_id* is 0 with no appended data.

Command Header

In each protocol message, a Command data structure is appended. The specific Command is identified by the value of the *CMDType* field in the Remote Operation Invoke/ Result/ Linked Result data structures.

The following Command types are used in the Protocol:

```
typedef u_16                CMDType;
#define                     CMD_EVENT_REPORT 0
#define                     CMD_CONFIRMED_EVENT_REPORT 1
#define                     CMD_GET 3
#define                     CMD_SET 4
#define                     CMD_CONFIRMED_SET 5
#define                     CMD_CONFIRMED_ACTION 7
```

The following command types are used:

CMD_EVENT_REPORT: An Event Report is used for an unsolicited event message.

CMD_CONFIRMED_EVENT_REPORT: The Confirmed Event Report is an unsolicited event message for which the receiver must send an Event Report Result message.

CMD_GET: The Get operation is used to request attribute values of managed objects. The receiver responds with a Get Result message.

CMD_SET: The Set operation is used to set values of managed objects.

CMD_CONFIRMED_SET: The Confirmed Set operation is used to set attribute values of managed objects. The receiver responds with a Set Result message.

CMD_CONFIRMED_ACTION: The Confirmed Action is a message to invoke an activity on the receiver side. The receiver must send an Action Result message.

For confirmed messages, the receiver must send the appropriate result message. For both the confirmed and unconfirmed Event Report, an *EventReportArgument* is appended.

If the result message is not received within 3 seconds, the IntelliVue monitor resends the message. If the message has not been confirmed after sending it 3 times (2 resend tries), the association is aborted by the IntelliVue monitor.

Event Report

The Event Report command (CMD_EVENT_REPORT) is used for unsolicited messages from the sending device to the receiving device. It is appended to the Remote Operation Invoke message. In the Protocol the Event Report may require a response from the receiver (if a response is required, the CMD_CONFIRMED_EVENT_REPORT Command identifier is used).

The Event Report message uses the following data structure:

```
typedef struct {
    ManagedObjectId managed_object; /* ident. of sender */
    RelativeTime    event_time;     /* event time stamp */
    OIDType         event_type;     /* identification of event */
    u_16            length;         /* size of appended data */
} EventReportArgument;
```

- managed_object

  Identifies the object that generates the unsolicited Event Report command.
- event_time

  The relative time (in 1/8ms time ticks) of the event.
- event_type

  Identifies the event type and thus the data structure that is appended.
- length

  This field defines the remaining number of bytes in the message (which is the size of the event specific data appended to this data structure).

Event-specific data is appended to the data type.

Event Report Result

The Event Report Result command is used as a response message to the Event Report message. It is appended to the Operation Result message with the *command_type* CMD_CONFIRMED_EVENT_REPORT.

The Event Report Result uses the following data structure:

```
typedef struct {
    ManagedObjectId managed_object; /* mirrored from EvRep */
    RelativeTime    current_time;   /* result time stamp */
    OIDType         event_type;     /* identification of event */
    u_16            length;         /* size of appended data */
} EventReportResult;
```

- managed_object

  Identifies the object to which the response is sent back. This field must be mirrored back from the Event Report message.
- event_time

  The relative time (in 1/8ms time ticks) of the event result.
- event_type

  Identifies the event type and thus the data structure that is appended. This field must contain the same value as the Event Report.
- length

  This field defines the remaining number of bytes in the message (which is the size of the event specific result data appended to this data structure).

Event-specific data is appended to the data type.

Action

The ACTION command (CMD_CONFIRMED_ACTION) is used to call a Protocol specific method in the receiver. The Protocol uses this command to call the *Data Poll* method which returns device data. The ACTION command is appended to the Operation Invoke message.

The Action command uses the following data structure:

```
typedef struct {
    ManagedObjectIdmanaged_object;/* addressed object */
    u_32            scope;          /* fixed value 0 */
    OIDType         action_type;    /* identification of method */
#define NOM_ACT_POLL_MDIB_DATA            3094
#define NOM_ACT_POLL_MDIB_DATA_EXT        61755
    u_16            length;         /* size of appended data */
} ActionArgument;
```

- managed_object

  Identifies the object to which the ACTION command is sent.

- scope

  Contains a fixed value 0 in this version of the protocol.

- action_type

  Identifies the specific method that should be called (and thus the data type that is appended to this data structure).

  NOM_ACT_POLL_MDIB_DATA is used for a Single Poll Data Request.

  NOM_ACT_POLL_MDIB_DATA_EXT is used for an Extended Poll Data Request

- length

  This field defines the remaining number of bytes in the message (which is the size of the method specific data appended to this data structure).

Method-specific data is appended to the data type.

Action Result

The Action Result command is used as a response message to the Action message. It is appended to the Operation Result message or an Operation Linked Result message (if the size of the returned data exceeds a maximum message size). The *command_type* is set to CMD_CONFIRMED_ACTION.

The Action Result uses the following data structure:

```
typedef struct {
    ManagedObjectId managed_object;
    OIDType         action_type;    /* identification of method */
    u_16            length;         /* size of appended data */
} ActionResult;
```

- managed_object

  Identifies the object that responds to the ACTION command (usually mirrored from ACTION command).

- action_type

  Identifies the specific method that was called (and thus the data type that is appended to this data structure).

- length

  This field defines the remaining number of bytes in the message (which is the size of the method specific result data appended to this data structure).

Method-specific data is appended to the data type.

Get

The Get command (CMD_GET) specifies attributes that should be returned. It is appended to an Operation Invoke message.

The Get command uses the following data structure:

```
typedef struct {
        ManagedObjectId       managed_object;
        u_32                  scope;
        AttributeIdList       attributeIdList;
} GetArgument;
```

- managed_object

  Identifies the object to which the Get command is sent.

- scope

  Contains a fixed value 0 in this version of the protocol.

- attributeIdList

  Contains the list of attribute identifiers.

```
typedef struct {
        u_16                  count;
        u_16                  length;
        OIDType               value[1];
} AttributeIdList;
```

Get Result

The Get Result is returned in response to the Get command. It is appended to an Operation Result or Operation Linked Result message.

The Get Result uses the following data structure:

```
typedef struct {
        ManagedObjectId       managed_object;
        AttributeList         attributeList;
} GetResult;
```

- managed_object

  Identifies the object that responds to the Get command.

- attributeList

  Contains the requested attributes.

Set

The Set command (CMD_SET) or Confirmed Set command (CMD_CONFIRMED_SET) specifies attributes that should be added, replaced, or removed. It is appended to an Operation Invoke message.

The Set command uses the following data structures:

```
typedef struct {
        ManagedObjectId       managed_object;
        u_32                  scope;
        ModificationList      modificationList;
} SetArgument;
```

- managed_object

  Identifies the object to which the Get command is sent.

- scope

  Contains a fixed value 0 in this version of the protocol.

- modificationList

  Contains the attribute ids and values to be modified.

```
typedef struct {
        u_16                  count;
        u_16                  length;
        AttributeModEntry value[1];
} ModificationList;

typedef struct {
        ModifyOperator        modifyOperator;
        AVAType               attribute;
} AttributeModEntry;

typedef u_16          ModifyOperator;
#define REPLACE                        0
#define ADD_VALUES                     1
#define REMOVE_VALUES                  2
#define SET_TO_DEFAULT                 3
```

Set Result

The Set Result is returned in response to the Confirmed Set command. It is appended to an Operation Result or Operation Linked Result message.

The Set Result uses the following data structure:

```
typedef struct {
        ManagedObjectId       managed_object;
        AttributeList         attributeList;
} SetResult;
```

- managed_object

  Identifies the object that responds to the Set command.

- attributeList

  Contains all modified attributes.

Command Structure Summary

The following diagram shows how the different generic Protocol Command command structures are built from the different data type definitions that were introduced in this section.

<table>
<tr><td colspan="10">SPpdu</td></tr>
<tr><td colspan="10">ROapdus</td></tr>
<tr><td colspan="4">ROIVapdu</td><td rowspan="3"></td><td colspan="3">RORSapdu
ROLRSapdu</td><td rowspan="3"></td><td>ROERapdu</td></tr>
<tr><td>Event Report Argument</td><td rowspan="2"></td><td>Action Argument</td><td rowspan="2">Get Argument
Set Argument</td><td>Event Report Result</td><td>Action Result</td><td rowspan="2">Get Result
Set Result</td><td>Error Data</td></tr>
<tr><td>Event Data</td><td>Action Data</td><td>Event Result Data</td><td>Action Result Data</td><td></td></tr>
</table>

From this generic message structure the specific Protocol Command messages introduced in “Protocol Dialog” on page 25 are derived by:

- Defining identifier codes for the supported specific Event Report and Action types. These identifier codes are the values of the *event_type* and *action_type* fields.
- Defining the specific Event Data and Action Data data types for these Event Report and Action types.

Protocol Commands

This section describes the actual commands as constructed from the building blocks. Consult the “Command Structure Summary” on page 51 as a reference.

Notation

The Protocol Commands are constructed from the data types previously defined. A generic protocol machine must parse the individual elements of a command message separately, so in this chapter a special notation is used to define how the command messages are constructed (rather than defining composite C data type definitions).

Example:

```
MDSCreateEventReport ::=
 <SPpdu>
 <ROapdus (ro_type := ROIV_APDU)>
 <ROIVapdu (command_type := CMD_CONFIRMED_EVENT_REPORT) >
 <EventReportArgument (event_type := NOM_NOTI_MDS_CREAT)>
 <MDSCreateInfo>
```

This notation means that an MDS Create Event Report Command message is constructed from the individual data types listed in the < > brackets, which are C data types. Some elements of these data types have specific values. E.g. the *ro_type* field in the *ROapdus* data type has the value *ROIV_APDU.*

Additional data structures for appended event specific or method specific data are defined in the usual C type definition notation.

Most of the elements of the command messages contain length fields. You must take care to correctly set and parse these fields so that the message can be correctly parsed.

Device Discovery Messages

The Device Discovery messages lets the client locate new IntelliVue monitor devices in the network without prior knowledge of their IP address. The IntelliVue monitor only *sends* a Device Discovery on the LAN interface. This message is not available on the MIB/RS232 interface.

CONNECT INDICATION EVENT

The Connect Indication Event message is a sub-net-wide broadcast message in the normal Event Report format. It is sent to the port 24005.

The IntelliVue monitor resends the Connect Indication message as long as no logical connection to a central station has been established. The connection of a Data Export Computer Client does not stop the transmission of Connect Indication messages.

The IntelliVue monitor uses the retransmit strategy described in RFC 951. The initial resend period is 4 seconds, and this is doubled with each resend. The maximum resend period is approximately 64 seconds. The actual resend period contains a random component to avoid network congestion, e.g., after a power failure.

The UDP checksum in the Connect Indication message may be set to 0, indicating that no checksum has been calculated.

The Connect Indication message has the following structure:

```
ConnectIndication ::=
    <Nomenclature>
    <ROapdus (ro_type := ROIVapdu)>
    <ROIVapdu (command_type := CMD_EVENT_REPORT)>
    <EventReportArgument
            (managed_object:={NOM_MOC_MDS_COMPOS_SINGLE_BED,0,0},
            event_type := NOM_NOTI_MDS_CONNECT_INDIC)>
    <ConnectIndInfo>

typedef u_32 Nomenclature;
```

The nomenclature starts with two bytes 0x0, followed by one byte major and one byte minor version.

```
typedef AttributeList ConnectIndInfo;
```

See the section “Connect Indication Attributes” on page 107 for a list of attributes contained in the appended attribute list.

The Computer Client should parse the *ConnectIndInfo* to find out about the port for the Data Export protocol. The Computer Client must send requests to the port that is specified for the Data Export protocol.

The Computer Client application can run on any free local port, but must not change the port during the association (refer to “Definition of the Association Control Protocol” on page 65 for more information).

Connection Startup

After the logical connection has been established between the IntelliVue monitor and the Computer Client, the IntelliVue monitor sends the MDS Create Event message to announce version and status information.

MDS CREATE EVENT

The MDS Create Event describes the software and hardware configuration of the IntelliVue monitor. The Computer Client should parse this message to learn about the system configuration.

The MDS Create Event message has the following structure:

```
MDSCreateEventReport ::=
 <SPpdu>
 <ROapdus (ro_type := ROIV_APDU)>
 <ROIVapdu (command_type := CMD_CONFIRMED_EVENT_REPORT)>
 <EventReportArgument
    (managed_object := {NOM_MOC_VMS_MDS, 0, 0},
     event_type := NOM_NOTI_MDS_CREAT)>
 <MDSCreateInfo>
```

The MDS Create Information uses the following C type definition:

```
typedef struct {
    ManagedObjectId       managed_object;
    AttributeList         attribute_list;
} MdsCreateInfo;
```

- managed_object

  Identifies the MDS object. Contents is the same as in the *managed_object* field in the Event Report structure.

- attribute_list

  The attached *attribute_list* contains the IntelliVue monitor MDS attributes from the System Identification and from the System Application Attribute Group. See "Wave Objects" on page 82 for a list of all attributes.

Depending on the protocol and the protocol options which were negotiated when the association was established, the IntelliVue monitor may map its internal data representation to a representation which is supported by the negotiated protocol. Hence, the Connect Indication message may describe the system differently from the MDS Create Event message. In the case of differences, the MDS Create Event is the relevant message.

The MDS Create Event message contains both the "Date and Time" and the "Relative Time" attributes. The Computer Client can use this data to make a mapping from the relative time to the absolute time of the IntelliVue monitor. The Computer Client should regularly check if the mapping is still valid by sending a Single Poll Data Request for the MDS attributes ("SINGLE POLL DATA REQUEST" on page 55).

If the size of the Event Report (Event Report Result + Event Result Data) exceeds the size of a maximum message (MTU - Maximum Transmit Unit), multiple messages are sent. Each of these messages is sent as a single Event Report.

The Computer Client must confirm the MDS CREATE EVENT with a MDS CREATE EVENT RESULT message, otherwise the association will be aborted by the IntelliVue monitor. The MDS CREATE EVENT message is resent with a period of about 3 seconds. The association is aborted if the Event message has been sent 3 times without receiving a confirmation.

When the MDS Create Event message is resent, it has the same invoke ID as the original message.

MDS CREATE EVENT RESULT

As the MDS Create Event Report is a confirmed operation, the Computer Client must send a MDS Create Event Result message to confirm it.

The reply message has the following structure:

```
MDSCreateEventResult ::=
 <SPpdu>
 <ROapdus (ro_type := RORS_APDU)>
 <RORSapdu
    (invoke_id := mirrored from event report,
     command_type := CMD_CONFIRMED_EVENT_REPORT)>
 <EventReportResult
    (managed_object := mirrored from event report,
     event_type := NOM_NOTI_MDS_CREAT)
     length := 0 >
```

As the MDS Create Event Result message does not contain any appended additional information, the length of the appended information is set to 0.

The result message must have the same *invoke_id* as the event message.

Specific Data Access Commands

The following protocol commands are used to access the different types of data in the IntelliVue monitor.

SINGLE POLL DATA REQUEST

This message can be sent as soon as the logical connection is established and the MDS Create Event/ Reply message sequence is finished. The message calls a method that returns IntelliVue monitor device data in a single response message.

The message has the following structure:

```
MDSPollAction ::=
 <SPpdu>
 <ROapdus (ro_type := ROIV_APDU)>
 <ROIVapdu (command_type := CMD_CONFIRMED_ACTION)>
 <ActionArgument
    (managed_object := {NOM_MOC_VMS_MDS, 0, 0},
     action_type := NOM_ACT_POLL_MDIB_DATA)>
 <PollMdibDataReq>
```

The *managed_object* must be the same as the *managed_object* in the MDS Create Event message. This is the top level object which actually implements the Data Export protocol.

The appended *PollMdibDataRequest* has the following data type:

```
typedef struct{
    u_16           poll_number;
    TYPE           polled_obj_type;
    OIDType        polled_attr_grp;
} PollMdibDataReq;
```

- poll_number

  This field will be sent back in the response message. It is recommended to use this field as a counter.
- polled_obj_type

  Defines which objects (Numerics or Alarms or MDS or Patient Demographics) is polled.

The following is a list of supported objects and their corresponding TYPE values:

```
NUMERICS:      partition:     0x0001
               code:          NOM_MOC_VMO_METRIC_NU
WAVES:         partition:     0x0001
               code:          NOM_MOC_VMO_METRIC_SA_RT
ALERTS:        partition:     0x0001
               code:          NOM_MOC_VMO_AL_MON
Pat.Demog:     partition:     0x0001
               code:          NOM_MOC_PT_DEMOG
MDS:           patition:      0x0001
               code:          NOM_MOC_VMS_MDS
```

The codes are taken from the Object Oriented Elements partition of the nomenclature (see "Object Classes" on page 111).

- polled_attr_grp

  Defines which set of attributes is polled. For more information on the supported attribute groups and their contents, please refer to the section "Attribute Data Types and Constants Used" on page 75.

The IntelliVue monitor specifies limits on the maximum frequency for incoming SINGLE POLL DATA REQUEST messages. If the Computer Client sends messages with a frequency above the limit, some of the messages will be ignored (no response is sent). Separate limits are calculated for each object.

The IntelliVue monitor will process a maximum of one POLL DATA REQUEST messages for each object type per second. An additional POLL DATA REQUEST for Numeric Observed Values is allowed.

SINGLE POLL DATA RESULT

This message is sent by the IntelliVue monitor in response to the Single Poll Data Request.

The message has the following structure:

```
MDSPollActionResult ::=
 <SPpdu>
 <ROapdus (ro_type := RORS_APDU)>
 <RORSapdu (invoke_id := "mirrored from request message"
            command_type := CMD_CONFIRMED_ACTION)>
 <ActionResult
    (managed_object := {NOM_MOC_VMS_MDS, 0, 0},
     action_type := NOM_ACT_POLL_MDIB_DATA)>
 <PollMdibDataReply>
```

The appended *PollMdibDataReply* is constructed from the following data types:

- The *PollMdibDataReply* structure is the top level data structure returned in the Single Poll Data Result message. It contains the following fields:

```
typedef struct {
    u_16                 poll_number;
    RelativeTime         rel_time_stamp;
    AbsoluteTime         abs_time_stamp;
    TYPE                 polled_obj_type;
    OIDType              polled_attr_grp;
    PollInfoList         poll_info_list;
} PollMdibDataReply;
```

- poll_number

  The poll number field contains the value of the same field in the Poll Request message.

- rel_time_stamp

  The Relative Time Stamp is a high resolution time stamp that represents the system time when the event message is sent by the IntelliVue monitor.

  For Numerics, the Relative Time Stamp denotes the time, when the Numeric measurement was generated. It may contain 0 if no measurement has been made yet.

- abs_time_stamp

  The IntelliVue monitor does not support Absolute Time Stamps in the Poll Data Result. All fields contain 0xff. If the Computer Client needs Absolute Time Stamps, it should use the corresponding MDS attributes ("Relative Time" and "Date and Time" to map the rel_time_stamp to an *abs_time_stamp*.

- polled_obj_type

  Defines for which objects (Numerics or Alarms or MDS or Patient Demographics) data is returned in the Poll Result message.

- polled_attr_grp

  Defines which set of attributes is returned in the Poll Result message.

- poll_info_list

  This structure contains the attribute values of the objects included in the poll.

  The Poll Info List is an array structure where each *SingleContextPoll* element contains the poll result data of one naming context.

```
typedef struct {
    u_16                    count;
    u_16                    length;
    SingleContextPoll       value[1];
} PollInfoList;
```

- count

  Number of Single Context Poll structures that are appended.

- length

  Size in bytes of the appended Single Context Poll structures.

- value

  This field is a placeholder field only. It represents the specified number of appended Single Context Poll structures.

The Single Context Poll structure contains polled data of all object instances within one unique naming context (IntelliVue monitor supports multiple naming contexts). It contains the following fields:

```
typedef struct {
    MdsContext              context_id;
    struct {
        u_16                count;
        u_16                length;
        ObservationPoll     value[1];
    } poll_info;
} SingleContextPoll;
```

- context_id

  The *context_id* field is used when the sourcing device represents multiple physical devices, so that the Handle attribute would not allow a unique identification of the object instance.

- poll_info.count

  This field contains number of appended Observation Poll structures.

- poll_info.length

  This field contains the length in bytes of the appended list of Observation Poll structures.

- poll_info.value

  This field is a placeholder field only. It represents the specified number of appended Observation Poll structures.

The *ObservationPoll* represents the polled data of one object instance. It contains the following fields:

```
typedef struct {
    Handle                  obj_handle;
    AttributeList           attributes;
} ObservationPoll;
```

- obj_handle

  The handle identifies the object instance. It is used to identify the object in different Poll Reply Messages.

- attributes

  The attributes field is a list structured field containing the values of the polled object attributes. For a list of supported object attributes, see the chapter on “Attribute Data Types and Constants Used” on page 75.

If the size of data returned for a Poll Result (Action Result + Action Result Data) exceeds the size of a maximum message (MTU - Maximum Transmit Unit), multiple messages are returned. These messages use the Remote Operation Linked Result mechanism (“Remote Operation Linked Result” on page 44). This means that in all result messages except the last result message the *ROLRSapdu* is used instead of the *RORSapdu*).

When the Linked Result mechanism is used, the IntelliVue monitor may send the terminating Remote Operation Result message with an empty *PollInfoList* (count and length fields of the *PollInfoList* set to 0). It also may send Linked Result messages with one empty SingleContextPoll (count and length field of the *SingleContxtPoll* set to 0).

EXTENDED POLL DATA REQUEST

The Extended Poll Data Request allows the following extensions of the Single Poll Data Request:

- Access 12 second, 1 minute and 5 minute averaged Numerics.
- Access wave data
- Request periodic Poll Replies without sending a Poll Request every time.
- Request that only a limited number of objects is encoded within a Poll Result

The Extended Poll Data Request message is only allowed, if the Poll Profile Extensions optional package has been negotiated during the association phase. For more information on the negotiation of optional packages see the sections "Association Request Message" on page 67 and "Association Response Message" on page 73.

The message has the following structure:

```
MDSPollAction ::=
 <SPpdu>
 <ROapdus (ro_type := ROIV_APDU)>
 <ROIVapdu (command_type := CMD_CONFIRMED_ACTION)>
 <ActionArgument
    (managed_object := {NOM_MOC_VMS_MDS, 0, 0},
     action_type := NOM_ACT_POLL_MDIB_DATA_EXT)>
 <PollMdibDataReqExt>
```

The appended *PollMdibDataRequestExt* has the following data type:

```
typedef struct{
    u_16          poll_number;
    TYPE          polled_obj_type;
    OIDType       polled_attr_grp;
    AttributeList poll_ext_attr;
} PollMdibDataReqExt;
```

- poll_number

  This field will be sent back in the response message. It is recommended to use this field as a counter. See also the section "EXTENDED POLL DATA RESULT" on page 61 for more information about the handling of *poll_number*.

- polled_obj_type

  Defines for which objects data is returned in the Poll Result message. The Extended Poll Data Request message only allows the polling of Numerics, Waves and the Alert Monitor.

- polled_attr_grp

  Defines which set of attributes is returned in the Poll Result message.

- poll_ext_attr

  The appended *AttributeList* allows to define additional options.

Accessing 12 second, 1 minute and 5 minute averaged Numerics

Within the Poll Profile Extensions optional package, the Computer Client and the IntelliVue monitor have negotiated which data source (real-time or averaged) is used to obtain the Numeric data (refer to the chapter "Definition of the Association Control Protocol" on page 65 for more information on how to negotiate optional packages). Currently, the IntelliVue monitor allows the specification of one data source for Numeric data.

The IntelliVue monitor responds to an Extended Poll Data Request message with an Extended Poll Data Result message, which contains the Numeric data from the source specified in the Poll Profile Extensions optional package.

The normal Poll Data Request message always returns data from real-time measurements. If another data source has been negotiated in the Poll Profile Extensions optional package, the Poll Data Request message will fail, if no data from real-time measurements is available.

The *poll_ext_attr AttributeList* in the Extended Poll Data Request message allows to specify additional options. Currently, the following attributes are supported:

Attribute: Time Periodic Data Poll

The Time Periodic Data Poll attribute allows to request periodic Poll Replies for a given time.

```
Attribute ID:      NOM_ATTR_TIME_PD_POLL
Attribute Type:    PollDataReqPeriod
Attribute Groups:  -
Availability:      Optional
```

The PollDataReqPeriod is defined as follows:

```
typedef struct {
    RelativeTime    active_period;
} PollDataReqPeriod;
```

The *active_period* specifies the time for which the IntelliVue monitor will send periodic Poll Replies.

The *AttributeList* Structure may contain additional attributes, e.g. in future releases.

If the Computer Client adds the Time Periodic Data Poll attribute to the Extended Poll Data Request message, the IntelliVue monitor sends periodic Extended Poll Data Result messages for the time specified in the attribute.

Data Source	Result Period
real-time waves	256ms
real-time measurements	1s
12 secound averaged data	6s
1 minute averaged data	30s
5 minute averaged data	150s
alert data	1s

When the IntelliVue monitor receives an Extended Poll Data Request message, the first result message is sent immediately as a confirmation. It has the sequence number zero (see below). This allows the Computer Client to detect that its request was successful. The following messages are sent with the period specified in the table above.

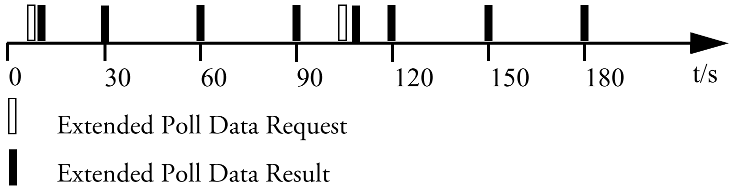


Figure 2 Period of Extended Poll Replies

The Computer Client should send a new Extended Poll Request before the time specified in the Time Periodic Data Poll attribute has expired. Each new Extended Poll Request is confirmed with an immediate Extended Poll Result message. However, the basic period of the replies is continued as illustrated in Figure 2 for 1 minute averaged data.

If the Computer Client uses the Extended Poll Request to access Realtime Numerics, it may happen that the IntelliVue monitor needs more than 1 second to encode all the data for the numerics (e.g. systems with a huge number of measurement modules). In this case the Poll Results will be sent at the highest possible frequency.

Limiting the Number of Objects in the Poll Result

In some cases, a Computer Client may want to limit the number of objects which are contained in a Poll Result. If the IntelliVue monitor is connected to a large number of measurement modules, a Poll Request for numerics will result in a large amount of data being sent from the IntelliVue monitor to the Computer Client.

Attribute: Number of Prioritized Objects

The attribute Number of Prioritized Objects specifies the maximum number of objects which will be encoded in the Poll Result.

```
Attribute ID:      NOM_ATTR_POLL_OBJ_PRIO_NUM
Attribute Type:    u_16
Attribute Groups:  -
Availability:      Optional
```

Based on an internal priority table, the IntelliVue monitor determines which objects will be added to the Poll Result. The priority table is constructed in the background, if the system configuration changes, it may take up to two minutes until the table has been updated. During this transition phase, the Poll Results sent by the monitor may contain less than the requested number of objects.

EXTENDED POLL DATA RESULT

When the IntelliVue monitor receives an Extended Poll Data Request message, it responds with a single or periodic Extended Poll Data Result messages.

The message has the following structure:

```
MDSPollActionResultExt ::=
 <SPpdu>
 <ROapdus (ro_type := RORS_APDU)>
 <RORSapdu (invoke_id := "mirrored from request message"
            command_type := CMD_CONFIRMED_ACTION)>
 <ActionResult
    (managed_object := {NOM_MOC_VMS_MDS, 0, 0},
     action_type := NOM_ACT_POLL_MDIB_DATA_EXT)>
 <PollMdibDataReplyExt>
```

The *PollMdibDataReplyExt* is defined as follows:

```
typedef struct PollMdibDataReplyExt {
    u_16                poll_number;
    u_16                sequence_no;
    RelativeTime        rel_time_stamp;
    AbsoluteTime        abs_time_stamp;
    TYPE                polled_obj_type;
    OIDType             polled_attr_grp;
    PollInfoList        poll_info_list;
} PollMdibDataReplyExt;
```

The *PollMdibDataReplyExt* structure is the top level data structure returned in the Extended Poll Data Result message. The appended data has the same structure as for the Single Poll Data Result.

The *PollMdibDataReplyExt* structure contains the following fields:

- poll_number

  The poll number field contains the value of the same field in the Extended Poll Request message.

- sequence_number

  The *sequence_number* is set to 0 when a new Extended Poll Data Request message is received. The IntelliVue monitor increases it with each periodic result message. This field allows the Computer Client to verify the sequence of the received result messages.

- rel_time_stamp

  The Relative Time Stamp is a high resolution time stamp that represents the system time when the event message is sent by the IntelliVue monitor.

  For Numerics, the Relative Time Stamp denotes the time when the Numeric measurement was generated. It may contain 0 if no measurement has been made yet.

  For Waves, the Relative Time Stamp denotes the beginning of the 256ms result period for real-time waves.

- abs_time_stamp

  The IntelliVue monitor does not support Absolute Time Stamps in the Poll Data Result. All fields contain 0xff. If the Computer Client needs Absolute Time Stamps, it should use the corresponding MDS attributes ("Relative Time" and "Date and Time" to map the rel_time_stamp to an abs_time_stamp).

- polled_obj_type

  Defines for which objects (Numerics or Alarms or MDS or Patient Demographic) data is returned in the Poll Result message.

- polled_attr_grp

  For more information on the supported attribute groups and their contents, please refer to the section “Attribute Data Types and Constants Used” on page 75.

- poll_info_list

  This structure contains the attribute values of the objects included in the poll.

Keep Alive Message

The IntelliVue monitor closes an association if it does not receive any protocol commands within a specified time (see “Definition of the Association Control Protocol” on page 65 to learn how the limit for a timeout is negotiated). If the Computer Client sends messages with a very low frequency (e.g. when using the extended poll mechanism) it must send a keep alive message to prevent the IntelliVue monitor from closing the association.

It is suggested that the Computer Client sends a Poll Data Request message for this purpose. This has the advantage that the message is confirmed and the Computer Client can detect a possible loss of the message. The Computer Client should chose a Poll Request which results in as little processing overhead as possible.

A suitable keep alive message would be a Poll Request for the Alert Monitor object, requesting the VMO Static Context Attribute group. The associated Poll Result sent by the IntelliVue monitor is a short message.

Specify Objects in the Poll Result

The Get and Set operations can be used to specify wave objects or numeric objects to be reported within the Poll Results.

There is a default priority list which depends on an internal priority table and the current system configuration. For wave objects and numeric objects, the default list can be replaced by a user defined priority list.

Due to the high amount of data it is always recommended to specify the required wave objects before requesting wave data.

NOTE Software versions < E.0 may have limited support of this command.

GET PRIORITY LIST REQUEST

The message has the following structure:

```
MDSGetPriorityList ::=
        <SPpdu>
        <ROapdus (ro_type := ROIV_APDU)>
        <ROIVapdu (command_type := CMD_GET)>
        <GetArgument
                (managed_object := {NOM_MOC_VMS_MDS, 0, 0})>
```

The Get argument’s *AttributeIdList* specifies the attribute identifiers:

- NOM_ATTR_POLL_RTSA_PRIO_LIST

  Wave object priority list.

- NOM_ATTR_POLL_NU_PRIO_LIST

  Numeric object priority list.

GET PRIORITY LIST RESULT

This message is sent in response to the Get Priority List Request.

The message has the following structure:

```
MDSGetPriorityListResult ::=
        <SPpdu>
        <ROapdus (ro_type := RORS_APDU)>
        <RORSapdu
                (invoke_id := "mirrored from request message",
                command_type := CMD_GET)>
        <GetResult
                (managed_object := {NOM_MOC_VMS_MDS, 0, 0})>
```

The Get result's *AttributeList* contains the requested attribute identifiers and values. The *TextIdList* structure is used to define the wave object priority list:

```
typedef struct {
        u_16                    count;
        u_16                    length;
        TextId                  value[1];
} TextIdList;
```

The array of *TextIds* specifies the objects by their label, as returned in the dynamic context.

SET PRIORITY LIST REQUEST

The message has the following structure:

```
MDSSetPriorityList ::=
        <SPpdu>
        <ROapdus (ro_type := ROIV_APDU)>
        <ROIVapdu (command_type := CMD_CONFIRMED_SET)>
        <SetArgument
                (managed_object := {NOM_MOC_VMS_MDS, 0, 0})>
```

The Set argument's *ModificationList* specifies the modify operations, attribute identifiers, and new values (if needed).

For the REPLACE operation, a wave object priority list attribute with modified *TextIdList* structure is attached.

For the SET_TO_DEFAULT operation, there is an empty attribute (*length* is 0)attached.

The ADD_VALUES and REMOVE_VALUES operations are not supported.

SET PRIORITY LIST RESULT

This message is sent in response to the Set Priority List Request.

The message has the following structure:

```
MDSSetPriorityListResult ::=
        <SPpdu>
        <ROapdus (ro_type := RORS_APDU)>
        <RORSapdu
                (invoke_id := "mirrored from request message",
                command_type := CMD_CONFIRMED_SET)>
        <SetResult
                (managed_object := {NOM_MOC_VMS_MDS, 0, 0})>
```

The Set result returns the modified *AttributeList*, as defined above.

6

Definition of the Association Control Protocol

Protocol Command Structure

The Protocol messages to establish the logical connection (association) between the IntelliVue monitor and a Computer Client follow the definitions of the ACSE Standard (ISO/IEC 8649 and ISO/IEC 8650), with some proprietary extensions.

All Association Control Commands share a common structure as shown here:

Session Header
Session Data
Presentation Header
User Data
Presentation Trailer

Figure 3 Protocol Commands for Association Control

For some messages, the Session Data and the User Data block may be empty.

A Computer Client can use the pre-defined building blocks for the Session Data, Presentation Header, and Presentation Trailer listed in the appendix to conveniently build valid messages ("Association Control Protocol Examples" on page 298 for a list of building blocks). Only the User Data block of the Association Request must be filled with Computer Client-specific data.

Protocol Commands

Protocol Command messages as defined in this section are the data structures that are transported within the transport layer messages.

The following commands are used to manage a logical connection between a Computer Client and a IntelliVue monitor:

- Association Request Message
- Association Response Message
- Refuse Message
- Release Request Message

- Release Response Message
- Abort Message

The Association Request message is sent from the Computer Client to the IntelliVue monitor when it wants to establish a new association. The *AssocReqUserData* contains information about the requested protocol and protocol options.

```
AssociationRequestMessage ::=
    <AssocReqSessionHeader>
    <AssocReqSessionData>
    <AssocReqPresentationHeader>
    <AssocReqUserData>
    <AssorReqPresentationTrailer>
```

The Association Response message is sent by the IntelliVue monitor if an Association Request message was parsed successfully and the association is accepted.

```
AssociationResponseMessage ::=
    <AssocRespSessionHeader>
    <AssocRespSessionData>
    <AssocRespPresentationHeader>
    <AssocRespUserData>
    <AssorRespPresentationTrailer>
```

If the Association Request message is corrupt, or if the association cannot be accepted (e.g. there is already another association), the IntelliVue monitor sends a Refuse message.

```
RefuseMessage ::=
    <RefuseSessionHeader>
    <RefuseSessionData>
    <RefusePresentationData>
    <RefuseUserData>
    <RefusePresentationTrailer>
```

When the Computer Client wants to terminate an association, it can send a Release Request message.

```
ReleaseRequestMessage ::=
    <ReleaseReqSessionHeader>
    <ReleaseReqSessionData>
    <ReleaseReqPresentationHeader>
    <ReleaseReqUserData>
    <ReleaseReqPresentationTrailer>
```

When the IntelliVue monitor receives a Release Request message, it sends a Release Response message as confirmation. The Release Response message indicates that the association has been terminated.

```
ReleaseRespMessage ::=
    <ReleaseRespSessionHeader>
    <ReleaseRespSessionData>
    <ReleaseRespPresentationHeader>
    <ReleaseRespUserData>
    <ReleaseRespPresentationTrailer>
```

The Abort message terminates an association without further confirmation. For example, the IntelliVue monitor sends an Abort message if an association is timed out (no communication from the Computer Client).

```
AbortMessage ::=
    <AbortSessionHeader>
    <AbortSessionData>
    <AbortPresentationHeader>
    <AbortUserData>
    <AbortPresentationTrailer>
```

Session Headers

The Session Headers can be used to identify the protocol commands. Each Session Header type maps to one protocol command.

The Session Header occupies the first bytes of the message. It is defined as follows:

```
typedef struct {
    u_8     type;
#   define CN_SPDU_SI             0x0D
#   define AC_SPDU_SI             0x0E
#   define RF_SPDU_SI             0x0C
#   define FN_SPDU_SI             0x09
#   define DN_SPDU_SI             0x0A
#   define AB_SPDU_SI             0x19
    LI      length;
} SessionHeader;
```

The *type* has the following meaning:

CN_SPDU_SI: A Session Connect header. The message contains an Association Request.

AC_SPDU_SI: A Session Accept header. The message contains an Association Response, indicating that the association has been established.

RF_SPDU_SI: A Session Refuse header. An association could not be established.

FN_SPDU_SI: A Session Finish header. The message contains a Release Request, indicating that the association should be terminated.

DN_SPDU_SI: A Session Disconnect header. The message contains a Release Response, indicating that the association has been terminated.

AB_SPDU_SI: A Session Abort header. The message contains an Abort message, indicating the immediate termination of the association.

If the first byte is 0xE1, the message is a Data Export Protocol command message (see “Definition of the Data Export Protocol” on page 35).

The *LI* field contains the length of the appended data (including all presentation data). The length encoding uses the following rules:

- If the length is smaller or equal 254 bytes, LI is one byte containing the actual length.
- If the length is greater than 254 bytes, LI is three bytes, the first being 0xff, the following two bytes containing the actual length.

Examples:

L = 15 is encoded as 0x0f

L = 256 is encoded as {0xff,0x01,0x00}

Message Encoding

The following section describes how a Computer Client can use the building blocks in the section “Association Control Protocol Examples” on page 298 to format correct Association Control messages.

Association Request Message

For the Association Request message, only the Session Header and the User Data must be filled out individually, as they contain variable data.

When using the building blocks, the presentation context ID for the Data Export Protocol is set to 2. This ID is sent in the SPpdu of all Data Export Protocol Commands.

The Session Header of the Association Request Message is defined as follows:

```
AssocReqSessionHeader ::=
    <SessionHead (type := CN_SPDU_SI)>
```

The length field in the Session Header must be set to the total length of the all appended data (including the presentation trailer).

Also the length field of the Presentation Header must be set to the total length of the appended message after this field. The field starts at the 2nd byte of the Presentation Header. It has the same format as the length field in the Session Header.

The User Data contains a specification of the requested protocol and protocol options. It is defined as follows:

```
AssocReqUserData ::=
    <ASNLength>
    <MDSEUserInfoStd>
```

The *ASNLength* contains the length of the *MDSEUserInfoStd*. It uses the following encoding rules:

- if the length is less or equal to 127, *ASNLength* is one byte, containing the actual length.
- if the length is greater than 127, *ASNLength* is several bytes long. The most significant bit (bit 0) of the first byte is set to 1, the bits 1 to 7 indicate the number of bytes which are appended to encode the actual length.

Examples:

L = 15 is encoded as 0x0f

L = 256 is encoded as {0x82,0x01,0x00}

The *MDSEUserInfoStd* is defined as follows:

```
typedef struct MDSEUserInfoStd {
    ProtocolVersion     protocol_version;
    NomenclatureVersion nomenclature_version;
    FunctionalUnits     functional_units;
    SystemType          system_type;
    StartupMode         startup_mode;
    AttributeList       option_list;
    AttributeList       supported_aprofiles;
} MDSEUserInfoStd;
```

The Computer Client must fill out the *MDSEUserInfoStd* data structure. It specifies the protocol versions and options the Computer Client supports. The IntelliVue monitor parses the *MDSEUserInfoStd* and constructs an Association Response message, which also contains a *MDSEUserInfoStd* data structure. The Association Response specifies which protocol versions and options will be used for the session.

The *ProtocolVersion* is a bit field containing the supported versions of the Data Export protocol. The Computer Client must set the bits for each version is supports. The IntelliVue monitor checks the supported versions and returns the bit for the highest commonly supported protocol version. If no matching version is found, the Association Request is refused.

```
typedef u_32       ProtocolVersion;
#define            MDDL_VERSION1  0x80000000
```

The *NomenclatureVersion* is a bit field containing the revision of the nomenclature which is used to name objects and their attributes. The Computer Client must set the bits for each version is supports. The IntelliVue monitor checks the supported versions and returns the bit for the highest commonly supported nomenclature version. If no matching version is found, the Association Request is refused.

```
typedef u_32 NomenclatureVersion;
#define            NOMEN_VERSION  0x40000000;
```

The *FunctionalUnits* is used to activate additional protocol functions. The Computer Client must set the bit for each functional unit it supports. The IntelliVue monitor checks the supported functional units and returns the bits for all commonly supported units (bitwise AND). No additional protocol functions have been defined yet.

```
typedef u_32       FunctionalUnits;
```

The *SystemType* is a bit field indicating whether the device is a Computer Client or a server. The Computer Client must set the SYST_CLIENT bit and the IntelliVue monitor will return the SYST_SERVER bit. If the SYST_CLIENT bit is not set in the Association Request, the association is refused.

```
typedef u_32 SystemType;
#define     SYST_CLIENT    0x80000000
#define     SYST_SERVER    0x00800000
```

The *StartupMode* is used to indicate the startup mode of the Computer Client and the IntelliVue monitor respectively. The IntelliVue monitor sets the bit for the startup mode which was used for the last reboot.

```
typedef u_32       StartupMode;
#define     HOT_START             0x80000000
#define     WARM_START            0x40000000
#define     COLD_START            0x20000000
```

If the IntelliVue monitor performs a COLD_START, all device settings are reset to the factory defaults. The configurations of the measurements might have changed and the patient data is lost.

The startup mode WARM_START and HOT_START indicate that configuration was not reset during the last restart.

The *option_list* can be used to negotiate additional protocol options in the form of an *AttributeList*. Currently, no further options are supported.

The *option_list* has a variable length. The offset of the *supported_aprofiles* field depends on the length of the *option_list.*

The *supported_aprofiles AttributeList* is used to define the available application profiles. An application profile specifies a set of protocol commands that is supported by the system. The Computer Client must add an entry for each supported profile to this list. The IntelliVue monitor parses the *supported_aprofiles* and returns the first profile in the list that is supported. If none of the profiles is supported, the Association Request is refused. The IntelliVue monitor supports the following profile:

Attribute: Poll Profile Support

The Poll Profile Support attribute contains the specification of the polling profile supported by the system.

```
Attribute ID:      NOM_POLL_PROFILE_SUPPORT
Attribute Type:    PollProfileSupport
Attribute Groups:  -
```

The *PollProfileSupport* is defined as follows:

```
typedef struct PollProfileSupport {
    PollProfileRevision    poll_profile_revision;
    RelativeTime           min_poll_period;
    u_32                   max_mtu_rx;
    u_32                   max_mtu_tx;
    u_32                   max_bw_tx;
    PollProfileOptions     options;
    AttributeList          optional_packages;
} PollProfileSupport;
```

The *PollProfileRevision* is a bit field containing the supported versions of the Polling Profile. The Computer Client must set the bits for each version it supports. The IntelliVue monitor checks the supported versions and returns the bit for the highest commonly supported profile version. If no matching version is found, the profile is not supported.

```
typedef u_32 PollProfileRevision;
#define POLL_PROFILE_REV_0       0x80000000
```

The *min_poll_period* specifies the minimum period with which the Computer Client wants to poll. If the IntelliVue monitor supports the requested poll period, it will return the value, otherwise it will return the minimum poll period it supports. The Computer Client should not send poll requests with a higher period than the negotiated value. For more information on poll periods, refer to the section “SINGLE POLL DATA REQUEST” on page 55.

The *min_poll_period* is also used to specify association time-outs. If the IntelliVue monitor does not receive any messages from the Computer Client within a given time, it sends an Abort message and terminates the association. The time-out periods depend on the negotiated *min_poll_period*, they are listed in the table below.

min_poll_period	**Association Time out**
< 3.3s	10s
3.3s ... 43s	3**min_poll_period*
> 43s	130s

The *max_mtu_rx* and *max_mtu_tx* fields contain the maximum size (MTU - Maximum Transport Unit) for protocol commands (the size of the protocol command is the size of the data appended after the Remote Operation Header).

The MTU negotiation uses the following procedure:

- The Computer Client determines the maximum size of a protocol command it can send and receive.
- The Computer Client sets *max_mtu_tx* to the maximum size it can transmit (i.e. the IntelliVue monitor should provide receive capabilities for messages of this size) and the *max_mtu_rx* to the maximum size it can receive (i.e. the IntelliVue monitor should not send larger commands).
- The IntelliVue monitor determines the maximum size of a protocol command it can send and receive.
- The IntelliVue monitor sets *max_mtu_tx* to the maximum size the Computer Client is allowed to transmit (this is the minimum of the *max_mtu_tx* the Computer Client requested and the message size the IntelliVue monitor *can receive).* The IntelliVue monitor sets *max_mtu_rx* to the maximum size the client must be able to receive (this is the minimum of the *max_mtu_rx the* Computer Client requested and the message size the IntelliVue monitor *can send).*

Example:

- The Computer Client can send 800 bytes and receive 500 bytes of user data in one message.
- The Computer Client sets *max_mtu_tx* to 800 and *max_mtu_rx* to 500.
- The IntelliVue monitor can send 700 bytes and receive 600 bytes in one message.
- The IntelliVue monitor sets *max_mtu_tx* to 600 bytes (the IntelliVue monitor cannot receive larger messages) and *max_mtu_rx* to 500 bytes (the Computer Client can not receive more than 500 bytes in a message).

The IntelliVue monitor requires that the Computer Client can receive protocol commands of at least 300 bytes. Otherwise the profile is not supported. Smaller command sizes would lead to a considerable communication overhead. The largest negotiable MTU is 1364 bytes for the LAN interface and 1000 Bytes for the MIB/RS232 interface. The resulting size of the data packets may be larger than the MTU, because the MTU covers only the size of the Command Header and the Command Specific Data.

It is recommended that the Computer Client uses a large MTU. This reduces processing overhead and in most cases avoids splitting of messages.

For wave data export, the Computer Client needs to be able to receive observed values with 256 ms of wave data in one message. The MTU should be at least 500 bytes (700 bytes with multiplexed context).

The *max_bw_tx* contains the estimated maximum transmit bandwidth which will be used. The IntelliVue monitor fills in the maximum transmit bandwidth it uses, the value 0xffffffff indicates that no estimation is possible (this is the default). The current software does not support bandwidth estimation.

The *PollProfileOptions* bit field is used to set additional profile options. The IntelliVue monitor sets the P_OPT_DYN_CREATE_OBJECTS and P_OPT_DYN_DELETE_OBJECTS bits to indicate that the number of internal objects (e.g. the number of Numerics) may change dynamically. The *PollProfileOptions* is defined as follows:

```
typedef u_32 PollProfileOptions;
#define P_OPT_DYN_CREATE_OBJECTS         0x40000000
#define P_OPT_DYN_DELETE_OBJECTS         0x20000000
```

The *optional_packages AttributeList* allows the definition of additional options supported in the profile. The Computer Client must add an entry for each optional package it requests. The IntelliVue monitor checks the packages and adds an entry for each package it supports in the Association Response.

An attribute constitutes an optional package. The Poll Profile Extension is an optional package available for use.

Attribute: Poll Profile Extensions

The Poll Profile Extensions attribute specifies some extensions for the standard polling profile. For more information on how to use these extensions refer to the section “EXTENDED POLL DATA REQUEST” on page 59.

```
Attribute ID:      NOM_ATTR_POLL_PROFILE_EXT
Attribute Type:    PollProfileExt
Attribute Groups:  -
```

The *PollProfileExt* is defined as follows:

```
typedef struct {
    PollProfileExtOptions options;
    AttributeList        ext_attr;
} PollProfileExt;

typedef u_32 PollProfileExtOptions;
#define POLL_EXT_PERIOD_NU_1SEC          0x80000000
#define POLL_EXT_PERIOD_NU_AVG_12SEC     0x40000000
#define POLL_EXT_PERIOD_NU_AVG_60SEC     0x20000000
#define POLL_EXT_PERIOD_NU_AVG_300SEC    0x10000000
#define POLL_EXT_PERIOD_RTSA             0x08000000
#define POLL_EXT_ENUM                    0x04000000
#define POLL_EXT_NU_PRIO_LIST            0x02000000
#define POLL_EXT_DYN_MODALITIES          0x01000000
```

The *PollProfileExtOptions* bit field defines available options for the Poll Profile Extensions package.

If the POLL_EXT_PERIOD_NU_1SEC bit is set, the Computer Client requests real-time measurements as source for Numeric data.

If the POLL_EXT_PERIOD_NU_AVG_12SEC bit is set, the Computer Client requests 12 second averaged data as source for Numeric data.

If the POLL_EXT_PERIOD_NU_AVG_60SEC bit is set, the Computer Client requests 1 minute averaged data as source for Numeric data.

If the POLL_EXT_PERIOD_NU_AVG_300SEC bit is set, the Computer Client requests 5 minute averaged data as source for Numeric data.

The Computer Client must set at least one of the bits for the numeric period, otherwise the optional package is ignored. Currently, the IntelliVue monitor supports only one source for an association. If more than one of the bits is set, the source with the smallest measurement period is selected. The IntelliVue monitor sets the corresponding bit in the Association Response message.

There may be only one active numeric source at a given time. If there is an active association on the LAN interface which has requested realtime numerics, it is not possible to establish another association on the MIB/RS232 interface which requests 1 minute averaged data. In this case, the association request would result in a refuse message.

If the POLL_EXT_PERIOD_RTSA bit is set, the computer client requests wave data. The patient monitor sets the corresponding bit in its response message to indicate wave data export capability.

The Computer Client must parse the Association Response message to find out whether the requested options have been accepted by the IntelliVue monitor.

If the POLL_EXT_ENUM bit is set, the computer client is allowed to request Enumeration objects.

If the POLL_NU_PRIO_LIST bit is set, the computer client is allowed to set the numeric priority list.

If the POLL_EXT_DYN_MODALITIES bit is set, the computer client gets all timestamps for metrics with dynamic modalities. They are not exported otherwise for compatibility reasons.

The *ext_attr AttributeList* is reserved for future extensions.

Release Request Message

The Release Request message does not contain variable data. It is sufficient for the Computer Client to use the building blocks listed in the section “Association Control Protocol Examples” on page 298.

Abort Message

The Abort message does not contain variable data. It is sufficient for the Computer Client to use the building blocks listed in the section “Association Control Protocol Examples” on page 298.

Message Parsing

In most cases, it is sufficient for the Computer Client to check the first byte of the association control message. The first byte defines the Session Layer header, which can be mapped to an Association Control command.

Association Response Message

The IntelliVue monitor sends the Association Response message if an association has been established successfully. The Computer Client must parse the User Data within this message to find out which protocol options have been negotiated.

The Computer Client should not assume that the same Association Request message will always lead to the same Association Response message. The internal state of the IntelliVue monitor might lead to different responses.

The Association Response message is identified by its Session Header:

```
AssocRespSessionHeader ::=
    <SessionHead (type := AC_SPDU_SI)>
```

When parsing the Association Response message, the Computer Client must find the beginning of the User Data. This can be done by identifying the following byte sequence within the message;

```
0xBE 0x80 0x28 0x80 0x81
```

or

```
0xBE 0x80 0x28 0x80 0x02 0x01 0x02 0x81
```

The User Data is defined as follows;

```
AssocRespUserData ::=
    <ASNLength>
    <MDSEUserInfoStd>
```

The last byte of the User Data must be followed by 16 bytes 0x00.

The *MDSEUserInfo* follows the same definitions as described above for the Association Request Message.

Refuse

The IntelliVue monitor sends a Refuse message if an Association Request message was not accepted, because it was formatted incorrectly or because the requested protocol and protocol options are not supported by the IntelliVue monitor.

A Refuse message is also sent, if the maximum number of concurrent associations has been reached. Currently, the IntelliVue monitor only supports one active association.

The Refuse messages is identified by its Session Header:

```
RefuseSessionHeader ::=
    <SessionHead (type := RF_SPDU_SI)>
```

Release Response

It is sufficient to check the Session Header to detect a Release Response message. The Session Header is defined as follows:

```
ReleaseRespSessionHeader ::=
    <SessionHead (type := DN_SPDU_SI)>
```

7

Attribute Data Types and Constants Used

The data types in this chapter are based on the data types introduced in the chapter “Definition of the Data Export Protocol” on page 35. Refer to this chapter for more information about the base data types.

All data types used in this guide assume that elements of structures are aligned on 2 byte boundaries. Many compilers use different alignment modes by default. Make sure that the compiler uses the right alignments when parsing and formatting protocol messages.

The Poll Reply messages may contain attributes which are not documented here. A Computer Client should ignore all unknown attributes.

With IntelliVue release G the nomenclature of some numeric and wave labels have been changed. The labels that previously resided in the namespace NOM_EMFC are now merged into the NOM_SCADA namespace and the new defined NOM_SETTING namespace.

At the end of this chapter is a mapping table to guide you through the transition.

Numeric Objects

Numeric Object Attributes

This section defines the attributes of the Numeric object, together with the attribute identifier codes and attribute data types.

Attribute: Handle

The Handle attribute identifies the Numeric object in the form of a numeric value. The Handle is unique within a device context (see Common Data Type - Global Handle). The actual value of the Handle attribute does not have a meaning. It is used for reference and relation purposes (e.g. Alert Monitor entries reference the Numeric object instance by means of the Handle).

```
Attribute ID:      NOM_ATTR_ID_HANDLE
Attribute Type:    Handle (see Definitions Shared by Protocols)
Attribute Groups:  VMO Static Context Group
Availability:      Mandatory
```

Attribute: Type

The Type attribute contains an identification of the object type.

```
Attribute ID:      NOM_ATTR_ID_TYPE
Attribute Type:    TYPE (see Definitions Shared by Protocols)
Attribute Groups:  VMO Static Context Group
Availability:      Mandatory
```

Attribute: Numeric Observed Value

The Numeric Observed Value attribute represents the (measured) value, along with state and identification data.

```
Attribute ID:      NOM_ATTR_NU_VAL_OBS
Attribute Type:    NuObsValue (see below)
Attribute Groups:  Metric Observed Value Group
Availability:      Conditional (either NuObsValue or
                   NuObsValueCmp must be present)
```

The *NuObsValue* data type is defined as follows:

```
typedef struct  {
    OIDType                 physio_id;
    MeasurementState        state;
    OIDType                 unit_code;
    FLOATType               value;
} NuObsValue;
```

The *physio_id* (physiological identifier) field contains a nomenclature code from the SCADA partition that identifies the represented value (typically a physiological measurement).

The *unit_code* field contains a nomenclature code from the dimension nomenclature partition. It identifies the units of measure.

The *value* field is a floating point number with the actual value. Before interpreting the numeric value, the *state* must be checked. Only if *state* indicates a valid measurement, should the *value* field be interpreted.

The *state* field is a bit field structure (multiple bits can be set simultaneously) defined as follows:

```
typedef u_16            MeasurementState;
#define INVALID                   0x8000
#define QUESTIONABLE              0x4000
#define UNAVAILABLE               0x2000
#define CALIBRATION_ONGOING       0x1000
#define TEST_DATA                 0x0800
#define DEMO_DATA                 0x0400
#define VALIDATED_DATA            0x0080
#define EARLY_INDICATION          0x0040
#define MSMT_ONGOING              0x0020
#define MSMT_STATE_IN_ALARM       0x0002
#define MSMT_STATE_AL_INHIBITED   0x0001
```

The bits have the following meaning:

INVALID: The source detects a sufficient degradation to render the data meaningless.

QUESTIONABLE: A problem exists, but it is still appropriate to present the data. This occurs when (1) either the degradation in the data is marginal or (2) the source cannot make a definite judgement on the reliability of the data.

UNAVAILABLE: The signal does not permit derivation of the numeric in question. This could be a transient state (e.g. first breath detected after an apnea -> no rate available), or a continuous state (no etCO_2 detection possible on a flat CO_2 wave).

CALIBRATION_ONGOING: Parameter is currently being calibrated.

TEST_DATA: The signal is an automatically generated test signal only and is not a valid patient signal. If this bit is set, the value is not suitable for patient diagnosis.

DEMO_DATA: The IntelliVue monitor runs in demonstration mode, the signal is automatically generated and is not a valid patient signal. If this bit is set, the value is not suitable for patient diagnosis.

VALIDATED_DATA: The value has been manually validated.

EARLY_INDICATION: The value represents an early estimate of the actual signal (the Non-Invasive Blood Pressure measurement e.g. sets this bit as soon as it has derived a systolic value, even if mean and diastolic values are still missing).

MSMT_ONGOING: A new aperiodic measurement is currently ongoing.

MSMT_STATE_IN_ALARM: Indicates that the numeric has an active alarm condition

MSMT_STATE_AL_INHIBITED: Alarms are switched off for the numeric (crossed bell)

The measurement is valid if the first octet of the state is all 0.

Attribute: Compound Numeric Observed Value

The Compound Numeric Observed Value attribute represents multiple (measured) values modelled in one Numeric object, along with state and identification data.

The Compound Numeric Observed Value is e.g. used to represent Blood Pressure measurements. For these measurements, systolic, diastolic and mean values are represented by a single Numeric object.

```
Attribute ID:      NOM_ATTR_NU_CMPD_OBS_VAL
Attribute Type:    NuObsValCmp (see below)
Attribute Groups:  Metric Observed Value Group
Availability:      Conditional (either NuObsValue or
                   NuObsValueCmp must be present)
```

The *NuObsValueCmp* data type is defined as follows:

```
typedef struct {
    u_16           count;
    u_16           length;
    NuObsValue     value[1];
} NuObsValueCmp;
```

The count field defines the number of *NuObsValue* elements in the structure. Note that the count field is variable, the number of elements may change over time. For a Blood Pressure measurement e.g there can be 3 values (systolic, diastolic, mean) or a single value only (mean only).

The length field defines the size of the array of *NuObsValue* structures in bytes.

The value field is a place holder for parsing.

Attribute: Absolute Time Stamp

The Absolute Time Stamp attribute is used to define a time tag for the current Numeric value. In the IntelliVue monitor, the attribute is used for aperiodic measurements only.

```
Attribute ID:      NOM_ATTR_TIME_STAMP_ABS
Attribute Type:    AbsoluteTime(see Definitions Shared by Protocols)
Attribute Groups:  Metric Observed Value Group
Availability:      Optional
```

Attribute: Relative Time Stamp

The Relative Time Stamp attribute is used to define a high resolution time tag for the current Numeric value.

```
Attribute ID:      NOM_ATTR_TIME_STAMP_REL
Attribute Type:    RelativeTime(see Definitions Shared by Protocols)
Attribute Groups:  Metric Observed Value Group
Availability:      Optional
```

Attribute: Label

The Label attribute is a 32 bit wide ID which represents the Numeric label string. The Label is unique for all numerics in the system.

```
Attribute ID:       NOM_ATTR_ID_LABEL
Attribute Type:     TextId
                    (see Protocol Common Definitions)
Attribute Group:    VMO Dynamic Context Group
Availability:       Optional
```

Attribute: Label String

The Label String attribute is a unicode string which contains the label string for a Numeric.

```
Attribute ID:       NOM_ATTR_ID_LABEL_STRING
Attribute Type:     String
                    (see Protocol Common Definitions)
Attribute Group:    VMO Dynamic Context Group
Availability:       Optional
```

The Label String does not contain the asterisk prefix displayed by the monitor to indicate whether a numeric has been manually entered. The client has to check the MetricCategory field of the MetricSpec attribute and add this prefix to obtain the identical label string that is displayed by the monitor.

Attribute: Display Resolution

The Display Resolution attribute is present if the resolution of the numeric shown on the display must be different from the resolution communicated in the Numeric Observed Value attribute. E.g. a Temperature is displayed with a resolution of 1/10, but the Observed Value is sent with a precision of 1/100 to get the necessary accuracy for differential temperatures. The Display Resolution attribute describes the format in which the value of a numeric is displayed on the screen.

```
Attribute ID:       NOM_ATTR_DISP_RES
Attribute Type:     DispResolution
Attribute Group:    VMO Dynamic Context Group
Availability:       Optional
```

The *DispResolution* is defined as follows:

```
typedef struct
{
    u_8     pre_point;
    u_8     post_point;
} DispResolution;
```

The value of *pre_point* denotes the number of digits before the decimal point. The value of *post_point* denotes the number of digits after the decimal point.

Attribute: Color

The Color attribute describes the color in which a numeric is displayed on the screen.

```
    Attribute ID:       NOM_ATTR_COLOR
    Attribute Type:     SimpleColour
    Attribute Group:    VMO Dynamic Context Group
    Availability:       Optional
```

The *SimpeColour* is defined as follows:

```
typedef u_16 SimpleColour;
#define COL_BLACK       0
#define COL_RED         1
#define COL_GREEN       2
#define COL_YELLOW      3
#define COL_BLUE        4
#define COL_MAGENTA     5
#define COL_CYAN        6
#define COL_WHITE       7
#define COL_PINK        20
#define COL_ORANGE      35
#define COL_LIGHT_GREEN 50
#define COL_LIGHT_RED   65
```

Attribute: Metric Specification

The Metric Specification attribute describes static properties of a numeric.

```
    Attribute ID:       NOM_ATTR_METRIC_SPECN
    Attribute Type:     MetricSpec
    Attribute Group:    VMO Static Context Group
    Availability:       Mandatory
```

The *MetricSpec* is defined as follows:

```
typedef struct
{
    RelativeTime        update_period;
    MetricCategory      category;
    MetricAccess        access;
    MetricStructure     structure;
    MetricRelevance     relevance;
} MetricSpec;
```

The *update_period* is the minimum time between changes of the observed value.

The *MetricCategory* is defined as follows:

```
typedef u_16            MetricCategory;
#define MCAT_UNSPEC                 0
#define AUTO_MEASUREMENT            1
#define MANUAL_MEASUREMENT          2
#define AUTO_SETTING                3
#define MANUAL_SETTING              4
#define AUTO_CALCULATION            5
#define MANUAL_CALCULATION          6
#define MULTI_DYNAMIC_CAPABILITIES  50
#define AUTO_ADJUST_PAT_TEMP        128
#define MANUAL_ADJUST_PAT_TEMP      129
#define AUTO_ALARM_LIMIT_SETTING    130
```

It allows to distinguish between measurements, calculations and settings. The values have the following meaning:

MCAT_UNSPEC: not specified

AUTO_MEASUREMENT: automatic measurement

MANUAL_MEASUREMENT: manual measurement

AUTO_SETTING: automatic setting

MANUAL_SETTING: manual setting

AUTO_CALCULATION: automatic calculation, e.g. differential temperature

MANUAL_CALCULATION: manual calculation

MULTI_DYNAMIC_CAPABILITIES: this measurement may change its category during operation or may be used in various modes.

AUTO_ADJUST_PAT_TEMP: measurement is automatically adjusted for patient temperature

MANUAL_ADJUST_PAT_TEMP: measurement manually adjusted for patient temperature

AUTO_ALARM_LIMIT_SETTING: this is not a measurement, but an alarm limit setting

The *MetricAccess* bit field provides info on how the metric value can be accessed and when a measurement is available.

```
typedef u_16            MetricAccess;
#define AVAIL_INTERMITTEND      0x8000
#define UPD_PERIODIC            0x4000
#define UPD_EPISODIC            0x2000
#define MSMT_NONCONTINUOUS      0x1000
```

The values have the following meaning:

AVAIL_INTERMITTEND: The intermitted availability bit is set, if the observed values not always available (e.g. only if a measurement is explicitly started).

UPD_PERIODIC: observed value is updated periodically

UPD_EPISODIC: observed value is updated episodically (exactly one update mode (UPD_) must be set

MSMT_NONCONTINUOUS: indicates that the measurement is non continuous (this is different from the update mode)

The *MetricStructure* describes if the object represents a single measurement or multiple

related measurements (an invasive blood pressure could be compound when it represents a pulsatile pressure like ABP and derives systolic, diastolic, mean values)

```
typedef struct MetricStructure {
    u_8             ms_struct;
    u_8             ms_comp_no;
} MetricStructure;
```

ms_struct describes the structure of the object, 0 means simple, 1 means compound object.

ms_comp_no contains the maximum number of components in the compound, it contains 0 for simple objects.

The MetricRelevance is a 16 bit wide field for internal use only.

```
typedef u_16 MetricRelevance;
```

Attribute MetricModality

The MetricModality attribute describes metric properties of a numeric that may depend on the usage of the measurement device to obtain a measurement.

```
    Attribute ID:       NOM_ATTR_METRIC_MODALITY
    Attribute Type:     MetricModality
    Attribute Group:    Metric Observed Value Group
    Availability:       Mandatory
```

The *MetricModality* is defined as follows:

```
typedef u_16 MetricModality;

#define METRIC_MODALITY_MANUAL         0x4000
#define METRIC_MODALITY_APERIODIC      0x2000
#define METRIC_MODALITY_VERIFIED       0x1000
```

Attribute Groups

The attributes of the Numeric object are arranged in the following attribute groups:

```
Attribute Group:   VMO Static Context Group
Group ID:          NOM_ATTR_GRP_VMO_STATIC
Description:       Static context of the object
Attributes:        Type, Handle, Metric Specification

Attribute Group:   VMO Dynamic Context Group
Group ID:          NOM_ATTR_GRP_VMO_DYN
Description:       Dynamic context of the object
Attributes:        Label, Label String, Color, Display Resolution

Attribute Group:   Metric Observed Value Group
Group ID:          NOM_ATTR_GRP_METRIC_VAL_OBS
Description:       Observed values of the object
Attributes:        Nu Observed Value,
                   Compound Nu Observed Value,
                   Absolute Time Stamp, Relative Time Stamp, MetricModality
```

Dynamic Context Changes

Internally, the IntelliVue monitor uses two different communication channels for attributes from the VMO Dynamic Context Group and the Metric Observed Value Group. This can lead to possible inconsistencies between these two attribute groups. Imagine that a Computer Client is polling all attribute groups. If the user changes the Label of a numeric (VMO Dynamic Context Group), the *physio_id* in the Nu Observed Value (Metric Observed Value Group) may be updated a short period later.

For real-time Numerics, this inconsistency is typically resolved after less than one second with the periodic update of the Observed Values. For averaged Numerics, the update of the Observed Values depends on the averaging period. It may be 12 seconds, 1 minute or 5 minutes.

Wave Objects

Wave Object Attributes

This section defines the attributes of the Wave object, together with the attribute identifier codes and attribute data types.

Attribute: Handle

The Handle attribute contains an identification of the wave object in the form of a numeric value. The actual value of the Handle attribute does not have a meaning. It is used for reference and relation purposes.

```
Attribute ID:          NOM_ATTR_ID_HANDLE
Attribute Type:        Handle (see Definitions Shared by Protocols)
Attribute Groups:      VMO Static Context Group
Availability:          Mandatory
```

Attribute: Type

The Type attribute contains an identification of the object type.

```
Attribute ID:          NOM_ATTR_ID_TYPE
Attribute Type:        TYPE (see Definitions Shared by Protocols)
Attribute Groups:      VMO Static Context Group
Availability:          Mandatory
```

Attribute: Metric Specification

The Metric Specification describes static properties of a metric object.

```
Attribute ID:     NOM_ATTR_METRIC_SPECN
Attribute Type:   MetricSpec
Attribute Groups: VMO Static Context Group
Availability:     Mandatory
```

The *MetricSpec* is defined as follows:

```
typedef struct {
       RelativeTime           update_period;
       MetricCategory         category;
       MetricAccess           access;
       MetricStructure        structure;
       MetricRelevance        relevance;
} MetricSpec;
```

The *update_period* specifies the time between observed values.

MetricCategory, *MetricAccess*, *MetricStructure*, and *MetricRelevance* are already defined for the Numeric object.

Attribute: Sample Array Specification

The Sample Array Specification describes static properties of a wave object.

```
Attribute ID:          NOM_ATTR_SA_SPECN
Attribute Type:        SaSpec
Attribute Groups:      VMO Static Context Group
Availability:          Mandatory
```

The *SaSpec* is defined as follows:

```
typedef struct {
       u_16                   array_size;
       SampleType             sample_type;
       SaFlags                flags;
} SaSpec;
```

The *array_size* specifies the maximum number of samples in one observed value.

The *SampleType* is defined as follows:

```
typedef struct {
       u_8                    sample_size;
       u_8                    significant_bits;
} SampleType;
```

The *sample_size* specifies the number of bits used to encode one wave sample.

The number of *significant_bits* is less or equal *sample_size*. To get the actual sample value, non-significant bits must be masked if indicated in the flags value.

The *SaFlags* is defined as follows:

```
typedef u_16           SaFlags;
#define SMOOTH_CURVE                  0x8000
#define DELAYED_CURVE                 0x4000
#define STATIC_SCALE                  0x2000
#define SA_EXT_VAL_RANGE              0x1000
```

The values have the following meaning:

SMOOTH_CURVE, DELAYED_CURVE : used for wave presentation
STATIC_SCALE: Scale and range specification does not change.
SA_EXT_VAL_RANGE: The non-significant bits in the sample value must be masked.

Attribute: Sample Array Fixed Value Specification

The Sample Array Fixed Value Specification defines a list of fixed sample values or bit masks that indicate specific conditions.

```
Attribute ID:          NOM_ATTR_SA_FIXED_VAL_SPECN
Attribute Type:        SaFixedValSpec16
Attribute Groups:      VMO Static Context Group
Availability:          Optional
```

The *SaFixedValSpec16* is a sequence of *SaFixedValSpecEntry16* elements:

```
typedef struct {
       u_16                   count;
       u_16                   length;
       SaFixedValSpecEntry16  value[1];
} SaFixedValSpec16;
typedef struct {
       SaFixedValId           sa_fixed_val_id;
       u_16                   sa_fixed_val;
} SaFixedValSpecEntry16;
```

The *SaFixedValId* is defined as follows:

```
typedef u_16           SaFixedValId;
#define SA_FIX_UNSPEC                 0
#define SA_FIX_INVALID_MASK           1
#define SA_FIX_PACER_MASK             2
```

```
#define SA_FIX_DEFIB_MARKER_MASK     3
#define SA_FIX_SATURATION            4
#define SA_FIX_QRS_MASK              5
```

The values have the following meaning:

SA_FIX_UNSPEC: Not specified.
SA_FIX_INVALID_MASK: Invalid sample mask.
SA_FIX_PACER_MASK: Pace pulse detected.
SA_FIX_DEFIB_MARKER_MASK: Defib marker in this sample.
SA_FIX_SATURATION: Indicates saturation condition in this sample.
(Note: despite the name, this is a mask as well.)
SA_FIX_QRS_MASK: Indicates QRS trigger around this sample.

The *sa_fixed_val* may be a value or a bit mask, as indicated in the *sa_fixed_val_id*.

Attribute: Sample Period

The Sample Period specifies the sample rate.

```
Attribute ID:         NOM_ATTR_TIME_PD_SAMP
Attribute Type:       RelativeTime (see Definitions Shared by Protocols)
Attribute Groups:     VMO Static Context Group
Availability:         Mandatory
```

Attribute: Label

The Label attribute contains a 32 bit wide ID which represents the wave label string. The Label is unique for all waves in the system.

```
Attribute ID:         NOM_ATTR_ID_LABEL
Attribute Type:       TextId (see Definitions Shared by Protocols)
Attribute Groups:     VMO Dynamic Context Group
Availability:         Optional
```

Attribute: Label String

The Label String is a unicode string which contains the label string for a wave.

```
Attribute ID:         NOM_ATTR_ID_LABEL_STRING
Attribute Type:       String (see Definitions Shared by Protocols)
Attribute Groups:     VMO Dynamic Context Group
Availability:         Optional
```

Attribute: Metric State

The Metric State attribute indicates metric on or off state.

```
Attribute ID:         NOM_ATTR_METRIC_STAT
Attribute Type:       MetricState
Attribute Groups:     VMO Dynamic Context Group
Availability:         Optional
```

The *MetricState* is a bit field defined as follows:

```
typedef u_16          MetricState;
#define METRIC_OFF                   0x8000
```

Attribute: Unit Code

The Unit Code attribute contains a nomenclature code from the dimension partition. It identifies the units of measure.

```
Attribute ID:          NOM_ATTR_UNIT_CODE
Attribute Type:        OIDType (see Definitions Shared by Protocols)
Attribute Groups:      VMO Dynamic Context Group
Availability:          Optional
```

Attribute: Color

The Color attribute describes the color in which a wave is displayed on the screen.

```
Attribute ID:          NOM_ATTR_COLOR
Attribute Type:        SimpleColour
Attribute Groups:      VMO Dynamic Context Group
Availability:          Optional
```

The *SimpleColour* is already defined for the Numeric object.

Attribute: Measure Mode

The Measure Mode attribute defines specific measurement modes.

```
Attribute ID:          NOM_ATTR_MODE_MSMT
Attribute Type:        MeasureMode
Attribute Groups:      VMO Dynamic Context Group
Availability:          Optional
```

For wave objects, the following *MeasureMode* bits are defined:

```
typedef u_16           MeasureMode;
#define CO2_SIDESTREAM               0x0400
#define ECG_PACED                    0x0200
#define ECG_NONPACED                 0x0100
#define ECG_DIAG                     0x0080
#define ECG_MONITOR                  0x0040
#define ECG_FILTER                   0x0020
#define ECG_MODE_EASI                0x0008
#define ECG_LEAD_PRIMARY             0x0004
```

The values have the following meaning:

CO2_SIDESTREAM: CO_2 sidestream.
ECG_PACED, ECG_NONPACED: Paced mode setting.
ECG_DIAG, ECG_MONITOR, ECG_FILTER: ECG filter setting.
ECG_MODE_EASI: EASI derived lead.
ECG_LEAD_PRIMARY: ECG primary lead.

Attribute: Metric Info Label

The Metric Info Label allows to specify an additional dynamic text (32 bit ID).

```
Attribute ID:          NOM_ATTR_METRIC_INFO_LABEL
Attribute Type:        TextId (see Definitions Shared by Protocols)
Attribute Groups:      VMO Dynamic Context Group
Availability:          Optional
```

Attribute: Metric Info Label String

The Metric Info Label String allows to specify an additional dynamic text (unicode string).

```
Attribute ID:          NOM_ATTR_METRIC_INFO_LABEL_STR
Attribute Type:        String (see Definitions Shared by Protocols)
Attribute Groups:      VMO Dynamic Context Group
Availability:          Optional
```

Attribute: Scale and Range Specification

The Scale and Range Specification describes a relation between scaled values and absolute values and also defines the range of the measured values and samples.

```
Attribute ID:          NOM_ATTR_SCALE_SPECN_I16
Attribute Type:        ScaleRangeSpec16
Attribute Groups:      VMO Dynamic Context Group
Availability:          Mandatory
```

The *ScaleRangeSpec16* is defined as follows:

```
typedef struct {
       FLOATType              lower_absolute_value;
       FLOATType              upper_absolute_value;
       u_16                   lower_scaled_value;
       u_16                   upper_scaled_value;
} ScaleRangeSpec16;
```

The scaled values refer to the wave samples in the observed values.

If the wave does not represent any absolute value, the absolute value fields must be *NaN* (Not a Number).

Attribute: Sample Array Physiological Range

The Sample Array Physiological Range is used for display scaling.

```
Attribute ID:          NOM_ATTR_SA_RANGE_PHYS_I16
Attribute Type:        ScaledRange16
Attribute Groups:      VMO Dynamic Context Group
Availability:          Optional
```

The *ScaledRange16* is defined as follows:

```
typedef struct {
       u_16                   lower_scaled_value;
       u_16                   upper_scaled_value;
} ScaledRange16;
```

Attribute: Visual Grid

The Visual Grid attribute allows to define grid lines.

```
Attribute ID:          NOM_ATTR_GRID_VIS_I16
Attribute Type:        SaVisualGrid16
Attribute Groups:      VMO Dynamic Context Group
Availability:          Optional
```

The *SaVisualGrid16* is defined as follows:

```
typedef struct {
       u_16                   count;
       u_16                   length;
       SaGridEntry16          value[1];
} SaVisualGrid16;
typedef struct {
       FLOATType              absolute_value;
       u_16                   scaled_value;
       u_16                   level;
} SaGridEntry16;
```

Different *levels* define relative importance of grid lines. *0* is the first (most important) level.

Attribute: Sample Array Calibration Specification

The Sample Array Calibration Specification allows to define the presence of a calibration bar or calibration stair.

```
Attribute ID:          NOM_ATTR_SA_CALIB_I16
Attribute Type:        SaCalData16
Attribute Groups:      VMO Dynamic Context Group
Availability:          Optional
```

The *SaCalData16* is defined as follows:

```
typedef struct {
       FLOATType             lower_absolute_value;
       FLOATType             upper_absolute_value;
       u_16                  lower_scaled_value;
       u_16                  upper_scaled_value;
       u_16                  increment;
       u_16                  cal_type;
#define BAR                         0
#define STAIR                       1
} SaCalData16;
```

Attribute: Sample Array Observed Value

The Sample Array Observed Value attribute represents the wave samples, along with state and identification data.

```
Attribute ID:          NOM_ATTR_SA_VAL_OBS
Attribute Type:        SaObsValue
Attribute Groups:      Metric Observed Value Group
Availability:          Conditional (either SaObsValue or
                       SaObsValueCmp is present)
```

The *SaObsValue* data type is defined as follows:

```
typedef struct {
       OIDType               physio_id;
       MeasurementState      state;
       struct {
              u_16                  length;
              u_8                   value[1];
       } array;
} SaObsValue;
```

The *physio_id* (physiological identifier) field contains a nomenclature code from the SCADA partition that identifies the represented wave (typically a physiological measurement).

The *state* indicates measurement validity. Refer to the Numeric object for a definition of the bit field. The measurement is valid if the first octet of the *state* is all *0*.

Attribute: Compound Sample Array Observed Value

The Compound Sample Array Observed Value attribute represents multiple waves modelled in one Wave object, along with state and identification data.

Compound Sample Array Observed Values are used to provide 250 samples/s ECG waves with common context.

```
Attribute ID:          NOM_ATTR_SA_CMPD_VAL_OBS
Attribute Type:        SaObsValueCmp
Attribute Groups:      Metric Observed Value Group
```

```
Availability:          Conditional (either SaObsValue or
                       SaObsValueCmp is present)
```

The *SaObsValueCmp* data type is defined as follows:

```
typedef struct {
       u_16                   count;
       u_16                   length;
       SaObsValue             value[1];
} SaObsValueCmp;
```

The *count* field defines the number of *SaObsValue* elements in the structure.

The *length* field defines the size of the array of *SaObsValue* structures in bytes.

The *SaObsValue* data type is defined above. The elements in a compound observed value can be identified by their *physio_id*.

Attributes Groups

The attributes of the Wave object are arranged in the following attribute groups:

```
Attribute Group:       VMO Static Context Group
Group ID:              NOM_ATTR_GRP_VMO_STATIC
Description:           Static context of the object
Attributes:            Handle, Type, Metric Specification, Sample Array
                       Specification, Sample Array Fixed Value Specification,
                       Sample Period
Attribute Group:       VMO Dynamic Context Group
Group ID:              NOM_ATTR_GRP_VMO_DYN
Description:           Dynamic context of the object
Attributes:            Label, Label String, Metric State, Unit Code,
                       Color, Measure Mode, Metric Info Label, Metric Info
                       Label String, Scale and Range Specification,
                       Sample Array Physiological Range, Visual Grid,
                       Sample Array Calibration Specification
Attribute Group:       Metric Observed Value Group
Group ID:              NOM_ATTR_GRP_METR_VAL_OBS
Description:           Observed values of the object
Attributes:            Sample Array Observed Value, Compound Sample Array
                       Observed Value
```

Enumeration Objects

Enumeration Object Attributes

This section defines the attributes of the enumeration objects, together with the attribute identifier codes and attribute data types.

Enumeration Objects are not available in software revisions below E.0.

Attribute: Handle

The Handle attribute identifies the enumeration object in the form of a numeral value. The Handle is unique within a device context (see Common Data Type - Global Handle). The actual value of the Handle attribute does not have a meaning. It is used for reference and relation purposes.

```
Attribute ID:      NOM_ATTR_ID_HANDLE
Attribute Type:    Handle (see Definitions Shared by Protocols)
Attribute Groups:  -
Availability:      Mandatory
```

Attribute: Type

The Type attribute contains an identification of the object type.

```
Attribute ID:          NOM_ATTR_ID_TYPE
Attribute Type:        TYPE (see Definitions Shared by Protocols)
Attribute Groups:      VMO Static Context Group
Availability:          Mandatory
```

Attribute: Metric Specification

The Metric Specification describes static properties of a metric object.

```
Attribute ID:          NOM_ATTR_METRIC_SPECN
Attribute Type:        MetricSpec
Attribute Groups:      VMO Static Context Group
Availability:          Mandatory
```

The *MetricSpec* is defined as follows:

```
typedef struct {
       RelativeTime           update_period;
       MetricCategory         category;
       MetricAccess           access;
       MetricStructure        structure;
       MetricRelevance        relevance;
} MetricSpec;
```

The *update_period* specifies the time between observed values.

MetricCategory, *MetricAccess*, *MetricStructure*, and *MetricRelevance* are already defined for the Numeric object.

Attribute: Label

The Label attribute is a 32 bit wide ID which represents the enumeration label string.

```
Attribute ID:        NOM_ATTR_ID_LABEL
Attribute Type:      TextId
                     (see Protocol Common Definitions)
Attribute Group:     VMO Dynamic Context Group
Availability:        Optional
```

Attribute: Label String

The Label String attribute is a unicode string which contains the label string for a enumeration.

```
Attribute ID:        NOM_ATTR_ID_LABEL_STRING
Attribute Type:      String
                     (see Protocol Common Definitions)
Attribute Group:     VMO Dynamic Context Group
Availability:        Optional
```

Attribute: Color

The Color attribute describes the color in which an enumeration is displayed on the screen.

```
Attribute ID:        NOM_ATTR_COLOR
Attribute Type:      SimpleColour
Attribute Group:     VMO Dynamic Context Group
Availability:        Optional
```

Attribute: Enum-Observed-Value

The Enum-Ovserved-Value attribute describes the current state of the enumeration object.

```
    Attribute ID:      NOM_ATTR_VAL_ENUM_OBS
    Attribute Type:    EnumObsVal
    Attribute Group:   VMO Observed Value Group
    Availablity:       Optional

typedef struct {
    OIDType            physio_id;
    MeasurementState   state;
    EnumVal            value;
} EnumObsVal;

typedef struct {
    OIDType            obj_id;
    FLOATType          num_val;
    OIDType            unit_code;
} EnumObjIdVal;

typedef struct {
    u_16               choice;
    u_16               length;
    union {
            OIDType        enum_obj_id;
            EnumObjIdVal   enum_obj_id_val;
    } u;
} EnumVal;
```

The field choice of structure EnumVal defines the valid structure of union u. Its values are defined as follows:

```
#define  ENUM_OBJ_ID_CHOSEN       1
#define  ENUM_OBJ_ID_VAL_CHOSEN   4
```

Attribute Groups

The attributes of the enumeration object are arranged in the following attribute groups:

```
Attribute Group:   VMO Static Context Group
Group ID:          NOM_ATTR_GRP_VMO_STATIC
Description:       Static context of the object
Attributes:        Type, Handle, Metric Specification

Attribute Group:   VMO Dynamic Context Group
Group ID:          NOM_ATTR_GRP_VMO_DYN
Description:       Dynamic context of the object
Attributes:        Label, Label String, Color

Attribute Group:   Metric Observed Value Group
Group ID:          NOM_ATTR_GRP_METRIC_VAL_OBS
Description:       Observed values of the object
Attributes:        Enumeration Observed Value,
                   Absolute Time Stamp, Relative Time Stamp
```

System Objects

System Objects Attributes

This section defines the attributes of the Medical Device System (MDS) object, together with the attribute identifier codes and attribute data types.

Attribute: Handle

The Handle attribute identifies the MDS object in the form of a numeral value. The Handle is unique within a device context (see Common Data Type - Global Handle). The actual value of the Handle attribute does not have a meaning. It is used for reference and relation purposes.

```
Attribute ID:      NOM_ATTR_ID_HANDLE
Attribute Type:    Handle (see Definitions Shared by Protocols)
Attribute Groups:  -
Availability:      Mandatory
```

Attribute: System Type

The System Type attribute contains an identification of the device type identified with the MDS object (e.g. monitor)

```
Attribute ID:      NOM_ATTR_SYS_TYPE
Attribute Type:    TYPE (see Definitions Shared by Protocols)
Attribute Groups:  System Identification Attribute Group
Availability:      Mandatory
```

For the MDS object, the OBJ nomenclature partition is used. The code value is a static identification.

Attribute: System Model

The System Model attribute contains a manufacturer ID and a manufacturer-specific model number for the device.

```
Attribute ID:      NOM_ATTR_ID_MODEL
Attribute Type:    SystemModel
Attribute Groups:  System Identification Attribute Group
Availability:      Mandatory
```

The *SystemModel* is defined as follows:

```
typedef struct  {
    VariableLabel manufacturer;
    VariableLabel model_number;
} SystemModel;
```

The *manufacturer* field is of variable length, hence the offset of *model_number* depends on the length of *manufacturer.* Currently, the IntelliVue monitor uses 4 characters for the *manufacturer* and 6 characters for the *model_number* (including the terminating '\0').

Attribute: System ID

The Sytem ID attribute contains a unique identifier for the device.

```
Attribute ID:      NOM_ATTR_SYS_ID
Attribute Type:    VariableLabel
                   (see Definitions Shared by Protocols)
Attribute Groups:  System Identification Attribute Group
Availability:      Mandatory
```

The IntelliVue monitor uses the 6 byte MAC address as identifier. Future versions might use an 8 byte EUI-64 identifier.

Attribute: Nomenclature Version

The Nomenclature Version attribute contains the version of the nomenclature used by the device.

```
Attribute ID:      NOM_ATTR_NOM_VERS
Attribute Type:    u_32
Attribute Groups:  System Identification Attribute Group
Availability:      Mandatory
```

The Nomenclature Version is composed of 16 bit major and 16 bit minor version number. The IntelliVue monitor currently uses the Nomenclature Version 1.0.

Attribute: System Localization

The System Localization attribute contains information about the language version used by the device.

```
Attribute ID:      NOM_ATTR_LOCALIZN
Attribute Type:    SystemLocal
Attribute Groups:  System Identification Attribute Group
Availability:      Optional
```

The *SystemLocal* is defined as follows:

```
typedef struct {
    u_32            text_catalog_revision;
    Language        language;
    StringFormat    format;
} SystemLocal;
```

The *text_catalog_revision* contains revision information about the texts used by the monitor. The two most significant bytes contain the version of the text catalog (one byte major, one byte minor revision). The text catalog defines the possible values for Attributes of the type *TextId*. A client which depends on a *TextId* having a specific value can use this information for revision control.

The lower two bytes of the *text_catalog_revision* are used for a language revision (one byte major, one byte minor revision). The language revision denotes the mapping from a *TextId* to an actual string in the monitor language.

The *Language* describes the language used by the monitor. It is defined as follows:

```
typedef u_16 Language;
#define LANGUAGE_UNSPEC           0
#define ENGLISH                   1
#define GERMAN                    2
#define FRENCH                    3
#define ITALIAN                   4
#define SPANISH                   5
#define DUTCH                     6
#define SWEDISH                   7
#define FINNISH                   8
#define NORWEG                    9
#define DANISH                    10
#define JAPANESE                  11
#define REP_OF_CHINA              12
#define PEOPLE_REP_CHINA          13
#define PORTUGUESE                14
#define RUSSIAN                   15
#define BYELORUSSIAN              16
#define UKRAINIAN                 17
#define CROATIAN                  18
#define SERBIAN                   19
#define MACEDONIAN                20
#define BULGARIAN                 21
#define GREEK                     22
#define POLISH                    23
#define CZECH                     24
#define SLOVAK                    25
#define SLOVENIAN                 26
#define HUNGARIAN                 27
#define ROMANIAN                  28
#define TURKISH                   29
#define LATVIAN                   30
#define LITHUANIAN                31
#define ESTONIAN                  32
#define KOREAN                    33
```

The *StringFormat* describes how strings are encoded. The IntelliVue monitor uses unicode encoding.

```
typedef u_16 StringFormat;
#define STRFMT_UNICODE_NT         11
```

Attribute: System Specification

The System Specification attribute contains a set of functional components supported by the system.

```
Attribute ID:      NOM_ATTR_SYS_SPECN
Attribute Type:    SystemSpec
Attribute Groups:  System Application Attribute Group
Availability:      Optional
```

The *SystemSpec* is defined as follows:

```
typedef struct {
    u_16            count;
    u_16            length;
    SystemSpecEntry value[1];
} SystemSpec;

typedef struct {
    PrivateOid     component_capab_id;
    u_16           length;
    u_16           value[1];
} SystemSpecEntry;
```

The supported components are:

```
Component ID:      NOM_MDIB_OBJ_SUPPORT
Component Type:    MdibObjectSupport
Availability:      Mandatory
```

The *MdibObjectSupport* is defined as follows:

```
typedef struct {
    u_16    count;
    u_16    length;
    MdibObjectSupportEntry        value[1];
} MdibObjectSupport;

typedef struct {
    TYPE    object_type;
    u_32    max_inst;
} MdibObjectSupportEntry;
```

The *MdibObjextSupport* contains a list of all object classes supported by the system and the maximum number of instances per class. If *max_inst* contains 0xffffffff, it is not defined.

Attribute: Mds General System Info

The Mds General System Info attribute contains global information about the monitor and its configuration.

```
    Attribute ID:       NOM_ATTR_MDS_GEN_INFO
    Attribute Type:     MdsGenSystemInfo
    Attribute Group:    System Application Attribute Group
    Availability:       Optional
```

The *MdsGenSystemInfo* is defined as follows:

```
typedef struct
{
    u_16            count;
    u_16            length;
    MdsGenSystemInfoEntry value[1];
} MdsGenSystemInfo;
```

The *MdsGenSysemInfoEntry* allows to encode generic system information. It has the following structure:

```
typedef struct
{
    u_16            choice;
#define MDS_GEN_SYSTEM_INFO_SYSTEM_PULSE_CHOSEN         1
    u_16            length;
    u_8             value[1]; /* placeholder for appended data */
} MdsGenSystemInfoEntry;
```

One MdsGenSytemInfoEntry is used to encode the System Pulse information. The monitor can generate a pulse rate from several sources.

```
Choice:         MDS_GEN_SYSTEM_INFO_SYSTEM_PULSE_CHOSEN 1
Type:           SystemPulseInfo
Availability:   Optional
```

The *SystemPulseInfo* is defined as follows:

```
typedef struct
{
    ManagedObjectId system_pulse;
    ManagedObjectId alarm_source;
} SystemPulseInfo;
```

It enfolds the *ManagedObjecId*s of the object instances selected as system-pulse respectively alarm-source.

Attribute: Production Specification

The Production Specification attribute contains a list of component revisions and serial numbers within the system.

```
Attribute ID:      NOM_ATTR_ID_PROD_SPECN
Attribute Type:    ProductionSpec
Attribute Groups:  System Production Attribute Group
Availability:      Optional
```

The *ProductionSpec* is defined as follows:

```
typedef struct {
    u_16            count;
    u_16            length;
    ProdSpecEntry   value[1];
} ProductionSpec;

typedef struct {
    u_16           spec_type;
#define UNSPECIFIED       0
#define SERIAL_NUMBER     1
#define PART_NUMBER       2
#define HW_REVISION       3
#define SW_REVISION       4
#define FW_REVISION       5
#define PROTOCOL_REVISION 6
    PrivateOid     component_id;
    VariableLabel  prod_spec;
} ProdSpecEntry;
```

The current IntelliVue monitor uses 10 characters for a serial number, 14 characters for part numbers and 8 characters for revision strings. The strings are not null-terminated.

The supported components are:

```
Component ID:      ID_COMP_PRODUCT
Description:       Overall product specification

Component ID:      ID_COMP_CONFIG
Description:       Specific system configuration

Component ID:      ID_COMP_BOOT
Description:       Boot code specification

Component ID:      ID_COMP_MAIN_BD
Description:       Mainboard hardware specification

Component ID:      ID_COMP_APPL_SW
Description:       Application software specification
```

See the section “Component IDs” on page 8-192 for the values of the *component_id*. The *ProductionSpec* may contain additional private entries.

To retrieve the IntelliVue monitor software revision, read the ProductSpecEntry with the Component ID “ID_COMP_APPL_SW”. Its prod_spec attribute contains a string of the form “H.00.00” describing the running software revision.

Attribute: MDS Status

The MDS Status attribute describes the device state.

```
Attribute ID:      NOM_ATTR_VMS_MDS_STAT
Attribute Type:    MDSStatus
Attribute Groups:  System Application Attribute Group
Availability:      Mandatory
```

The *MDSStatus* is defined as follows:

```
typedef u_16        MDSStatus;
#define     DISCONNECTED    0
#define     UNASSOCIATED    1
#define     OPERATING       6
```

The *MDSStatus* values have the following meaning:

DISCONNECTED: The IntelliVue monitor is not connected to the network.

UNASSOCIATED: The IntelliVue monitor is connected to the network, but no association is currently active.

OPERATING: The IntelliVue monitor has an association with a Computer Client.

Currently, a Computer Client will only see the MDS Status OPERATING, if the MDS has another Status, there is no association with a Computer Client.

Attribute: Bed Label

The Bed Label attribute contains a printable string identifying the system location.

```
Attribute ID:      NOM_ATTR_ID_BED_LABEL
Attribute Type:    String
                   (see Definitions Shared by Protocols)
Attribute Groups:  System Application Attribute Group
Availability:      Optional
```

The Bed Label can be entered in the Admit/Discharge dialog. It uses 16 bit unicode character encoding. Currently, the Bed Label is 17 characters (including terminating '\0'). If the actual label is shorter, the string is filled with '\0' characters.

Attribute: Operating Mode

The Operating Mode attribute identifies the current operating mode of the device.

```
Attribute ID:      NOM_ATTR_MODE_OP
Attribute Type:    PrivateOID
Attribute Groups:  System Application Attribute Group
Availability:      Optional
```

The Operating Mode is defined as a bit field. The following mode bits are defined:

```
#define     OPMODE_UNSPEC  0x8000
#define     MONITORING     0x4000
#define     DEMO           0x2000
#define     SERVICE        0x1000
#define     OPMODE_STANDBY 0x0002
#define     CONFIG         0x0001
```

The values have the following meaning:

OPMODE_UNSPEC: The Operating Mode is not specified.

MONITORING: Device is configured to monitor patient data (the default mode).

DEMO: Demonstration Mode with simulated patient data.

SERVICE: Device is in Service Mode.

STANDBY: Standby and Power Safe Mode.

CONFIG: Device is in Configuration Mode.

Exactly one of the bit out of the bits 0 - 4 must be set, bits 14 and 15 (the stand-by and config mode bits) can be set optionally.

Attribute: Application Area

The Application Area attribute describes the intended application area for the device.

```
Attribute ID:      NOM_ATTR_AREA_APPL
Attribute Type:    ApplicationArea
Attribute Groups:  System Application Attribute Group
Availability:      Optional
```

The *ApplicationArea* is defined as follows:

```
typedef u_16            ApplicationArea;
#define     AREA_UNSPEC            0
#define     AREA_OPERATING_ROOM    1
#define     AREA_INTENSIVE_CARE    2
#define     AREA_NEONATAL_INTENSIVE_CARE 3
#define     AREA_CARDIOLOGY_CARE   4
```

The values have the following meaning:

AREA_UNSPEC: The application area has not been specified.

AREA_OPERATING_ROOM: The application area has been specified as an operating room.

AREA_INTENSIVE_CARE: The application area has been specified as an intensive care unit.

AREA_NEONATAL_INTENSIVE_CARE: The application area has been specified as a neonatal intensive care unit.

AREA_CARDIOLOGY_CARE: The application area has been specified as a cardiology care unit.

Attribute: Date and Time

The Date and Time attribute contains the current device time.

```
Attribute ID:      NOM_ATTR_TIME_ABS
Attribute Type:    AbsoluteTime
                   (see Definitions Shared by Protocols)
Attribute Groups:  System Application Attribute Group
Availability:      Optional
```

Attribute: Relative Time

The Relative Time attribute contains the current device relative time.

```
Attribute ID:      NOM_ATTR_TIME_REL
Attribute Type:    RelativeTime
                   (see Definitions Shared by Protocols)
Attribute Groups:  System Application Attribute Group
Availability:      Optional
```

The Relative Time is set to zero after each power cycle.

Attribute: Altitude

The Altitude attribute contains the system altitude above or below sea level.

```
Attribute ID:      NOM_ATTR_ALTITUDE
Attribute Type:    i_16
Attribute Groups:  System Application Attribute Group
Availability:      Optional
```

Attribute: Line Frequency

The Line Frequency attribute describes the frequency of the main power supply in Hz.

```
Attribute ID:      NOM_ATTR_LINE_FREQ
Attribute Type:    LineFrequency
Attribute Groups:  System Application Attribute Group
Availability:      Optional
```

The *LineFrequency* is defined as follows:

```
typedef u_16            LineFrequency;
#define     LINE_F_UNSPEC  0
#define     LINE_F_50HZ    1
#define     LINE_F_60HZ    2
```

Attribute: Association Invoke ID

The Association Invoke ID attribute is a counter for the number of associations. It is incremented with each new association.

```
Attribute ID:      NOM_ATTR_ID_ASSOC_NO
Attribute Type:    u_16
Attribute Groups:  System Identification Attribute Group
Availability:      Optional
```

Attribute Groups

The attributes of the Medical Device System object are arranged in the following attribute groups:

```
Attribute Group:   System Identification Attribute Group
Group ID:          NOM_ATTR_GRP_SYS_ID
Description:       Identification of the system
Attributes:        System Type, System Model, System Id,
                   Nomenclature Version, System Localization, Association Invoke Id

Attribute Group:   System Application Attribute Group
Group ID:          NOM_ATTR_GRP_SYS_APPL
Description:       System Capabilities and Settings
Attributes:        System Specification, MDS Status, Bed Label,
                   Operating Mode, Application Area, Data and
                   Time, Relative Time, Altitude, Line
                   Frequency, Mds General System Info

Attribute Group:   System Production Attribute Group
Group ID:          NOM_ATTR_GRP_SYS_PROD
Description:       HW and SW configuration
Attributes:        Production Specification
```

Alert Monitor Object

Attributes of the Alert Monitor Object

This section defines the attributes of the Alert Monitor object, together with the attribute identifier codes and attribute data types.

The Alert Monitor object represents the overall device alert condition. It contains a global alert status and a list of active technical and patient alerts.

Attribute: Handle

The Handle attribute identifies the Alert Monitor object in the form of a numeral value. The Handle is unique within a device context (see Common Data Type - Global Handle). The actual value of the Handle attribute does not have a meaning. It is used for reference and relation purposes.

```
Attribute ID:      NOM_ATTR_ID_HANDLE
Attribute Type:    Handle (see Definitions Shared by Protocols)
Attribute Groups:  VMO Static Context Group
Availability:      Mandatory
```

Attribute: Type

The Type attribute contains an identification of the object type represented by the Alert Monitor.

```
Attribute ID:      NOM_ATTR_ID_TYPE
Attribute Type:    TYPE (see Definitions Shared by Protocols)
Attribute Groups:  VMO Static Context Group
Availability:      Mandatory
```

Attribute: Device Alert Condition

The Device Alert Condition attribute contains global device alert status information.

```
Attribute ID:      NOM_ATTR_DEV_AL_COND
Attribute Type:    DeviceAlertCondition
Attribute Groups:  Alert Monitor Group
Availability:      Mandatory
```

The *DeviceAlertCondtion* is defined as follows:

```
typedef struct {
    AlertState      device_alert_state;
    u_16            al_stat_chg_cnt;
    AlertType       max_p_alarm;
    AlertType       max_t_alarm;
    AlertType       max_aud_alarm;
} DeviceAlertCondition;
```

The *AlertState* is a bit field defined as follows:

```
typedef u_16 AlertState;
#define AL_INHIBITED              0x8000
#define AL_SUSPENDED              0x4000
#define AL_LATCHED                0x2000
#define AL_SILENCED_RESET         0x1000
#define AL_DEV_IN_TEST_MODE       0x0400
#define AL_DEV_IN_STANDBY         0x0200
#define AL_DEV_IN_DEMO_MODE       0x0100
#define AL_NEW_ALERT              0x0008
```

The *AlertState* is used for the overall device alert state and for the specific state of each alert. The bits in *AlertState* have the following meaning:

AL_INHIBITED: Alert is switched off.

AL_SUSPENDED: Alert inactivated temporarily, alert condition is acknowledged.

AL_LATCHED: Alert condition is not active but latched, note that technical alarms are never latching.

AL_SILENCED_RESET: Alert condition stopped but alarming re-enabled (only for *DeviceAlertCondition*).

AL_DEV_IN_TEST_MODE: Device is in a temporary test mode.

AL_DEV_IN_STANDBY: Device is in standby mode.

AL_DEV_IN_DEMO_MODE: Indicates that the device is in demo mode.

AL_NEW_ALERT: Indicate a new alarm (not in *DeviceAlertCondition*). A Computer Client might not see this bit if it does not poll fast enough or other delays occur.

The *al_stat_chg_cnt* is an internal change counter. A Computer Client should not interpret this field, because it can not be guaranteed that no internal message is missed.

The *AlertType* is a bit field defined as follows:

```
typedef u_16 AlertType;
#define     NO_ALERT       0
#define     LOW_PRI_T_AL   1
#define     MED_PRI_T_AL   2
#define     HI_PRI_T_AL    4
#define     LOW_PRI_P_AL   256
#define     MED_PRI_P_AL   512
#define     HI_PRI_P_AL    1024
```

Intellivue monitors with software revision E.0 or higher allow changing of the inop severity for various inop alarms. These changes are reflected in the AlertType bitfield.

The bits have the following meaning:

NO_ALERT: No alert active.

LOW_PRI_T_AL: Low priority technical alarm (soft inop). These inops are generated after a signal analysis (e.g "Noisy ECG").

MED_PRI_T_AL: Medium priority technical alarm (hard inop). These inops are generated during inoperable parameter measurement because of hardware faults or no transducer connected (e.g "Leads Off", "ABP No Transducer")

HI_PRI_T_AL: High priority technical alarm (severe inop).

LOW_PRI_P_AL: Awareness Condition (short yellow alarm): These alarms are marked with a "**" in the alarm string and a specific short yellow alarm sound is issued. Today short yellow alarms are generated only from arrhythmia computer.

MED_PRI_P_AL: Medium priority patient alarm (yellow alarm): These alarms are marked with a "**" in the alarm string. They indicate a less critical patient condition usually due to violation of user defined criteria (e.g. limit violation alarm).

HI_PRI_P_AL: High priority patient alarm (red alarm): These alarms are marked with a "***" in the alarm string. These alarms indicate a life threatening patient condition.

Attribute: Device T-Alarm List

The Device T-Alarm List attribute contains the active technical alarms (inops) in the system.

```
Attribute ID:      NOM_ATTR_AL_MON_T_AL_LIST
Attribute Type:    DevAlarmList
Attribute Groups:  Alert Monitor Group
Availability:      Mandatory
```

The *DevAlarmList* is defined as follows:

```
typedef struct {
    u_16           count;
    u_16           length;
    DevAlarmEntry  value[1];
} DevAlarmList;

typedef struct {
    OIDType               al_source;
    OIDType               al_code;
    AlertType             al_type;
    AlertState            al_state;
    ManagedObjectId       object;
    PrivateOid            alert_info_id;
#define GEN_ALMON_INFO 513
#define STR_ALMON_INFO 516
    u_16                  length;
} DevAlarmEntry;
```

The *al_source* is taken from the Object Oriented or the SCADA partition (depending on *al_code*). It identifies the origin of the alert (e.g. temperature).

The *al_code* is taken from the Events partition and describes the reason for the alert (e.g. high alarm). The least significant bit is used to define the nomenclature partition for *al_source*. Last bit 0 means SCADA partition, last bit 1 means Object Oriented partition.

The definitions for *AlertType* and *AlertState* can be found in the paragraph about the Device Alert Condition.

The *object* field contains a reference to the object which generated the alert. The object may not be known to the Computer Client, if the Data Export protocol does not allow accessing the specific object.

If the *alert_info_id* is set to GEN_ALMON_INFO, an *AlMonGenInfo* structure is appended:

```
typedef struct {
    u_16                  al_inst_no;
    TextId                al_text;
    AlertPriority         priority;
    AlertFlags            flags;
} AlMonGenInfo;
```

If the *alert_info_id* is set to STR_ALMON_INFO, an *StrAlMonInfo* structure is appended:

```
typedef struct {
    u_16                  al_inst_no;
    TextId                al_text;
    AlertPriority         priority;
    AlertFlags            flags;
    String                string;
} StrAlMonInfo;
```

Currently, the IntelliVue monitor only supports the *StrAlMonInfo* data type.

The *al_inst_no* is a private ID.

The *al_text* is a private ID.

The *AlertPriority* is defined as follows:

```
typedef u_16       AlertPriority;
```

The *AlertPriority* only allows prioritization within a group of alarms. A Computer Client application should use the *AlertType* to distinguish low and high priority alarms.

The *AlertFlags* type is defined as follows:

```
typedef u_16            AlertFlags;
#define BEDSIDE_AUDIBLE           0x4000
#define CENTRAL_AUDIBLE           0x2000
#define VISUAL_LATCHING           0x1000
#define AUDIBLE_LATCHING          0x0800
#define SHORT_YELLOW_EXTENSION    0x0400
#define DERIVED                   0x0200
```

The bits in the *AlertFlag* have the following meaning:

BEDSIDE_AUDIBLE: Alert sound at the bedside

CENTRAL_AUDIBLE: Alert sound at the central station

VISUAL_LATCHING: Alert is visible after the alarm condition has ceased. The alarm indication will exist until a specific action is taken by a user (e.g. Silence/Reset).

AUDIBLE_LATCHING: Alert is sound issued after the alarm condition has ceased. The alarm indication will exist until a specific action is taken by a user (e.g. Silence/Reset).

SHORT_YELLOW_EXTENSION: Alarm is not active but artificially extended for short yellow behavior.

DERIVED: Derived alarm.

The *String* contains the a description of the alarm in the language supported by the IntelliVue monitor. *Strings* for patient alarms are prefixed with two "**" or three "***" alarm stars (see “Definitions Shared by Protocols” on page 6-35 for UNICODE character encoding). Currently, the String is 19 characters long, including the terminating ’\0’.

Attribute: Device P-Alarm List

The Device P-Alarm List attribute contains the active patient alarm in the system.

```
Attribute ID:      NOM_ATTR_AL_MON_P_AL_LIST
Attribute Type:    DevAlarmList
Attribute Groups:  Alert Monitor Group
Availability:      Mandatory
```

The *DevAlarmList* data type is the same as for the Device T-Alarm List.

The data in a Device T-Alarm List or Device P-Alarm List might be too large to fit in a single message. In this case the Remote Operation Linked Result message will be used (see “Remote Operation Linked Result” on page 6-44). In this case each message will contain a correctly formatted Alarm list and the Computer Client must merge the lists to get the complete Device T-Alarm List or Device P-Alarm List.

Attribute Groups

The attributes of the Alert Monitor object are arranged in the following attribute groups:

```
Attribute Group:   VMO Static Context Group
Group ID:          NOM_ATTR_GRP_VMO_STATIC
Description:       Static context of the object
Attributes:        TYPE, Handle

Attribute Group:   Alert Monitor Group
Group ID:          NOM_ATTR_GRP_AL_MON
Description:       Alarm related attributes
Attributes:        Device Alert Condition, Device P-Alarm List,
                   Device T-Alarm List
```

Patient Demographics Object

Attributes of the Patient Demographic Object

This section defines the attributes of the Patient Demographics object, together with the attribute identifier codes and attribute data types.

The Patient Demographics object contains the patient information present in the system.

Attribute: Handle

The Handle attribute identifies the Patient Demographics object in the form of a numeral value. The Handle is unique within a device context (see Common Data Type - Global Handle). The actual value of the Handle attribute does not have a meaning. It is used for reference and relation purposes.

```
Attribute ID:      NOM_ATTR_ID_HANDLE
Attribute Type:    Handle (see Definitions Shared by Protocols)
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Mandatory
```

Attribute: Pat Demo State

The Pat Demo State attribute describes the current state of the Patient Demographics object.

```
Attribute ID:      NOM_ATTR_PT_DEMOG_ST
Attribute Type:    PatDemoState
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Mandatory
```

The *PatDemoState* is defined as follows:

```
typedef u_16       PatDmgState;
#define     EMPTY           0
#define     PRE_ADMITTED    1
#define     ADMITTED        2
#define     DISCHARGED      8
```

The values have the following meaning:

EMPTY: No patient information present.

PRE_ADMITTED: Currently not used.

ADMITTED: Patient information is present and valid.

DISCHARGED: Data is still available, but patient is no longer assigned to device.

Attribute: Patient Type

The Patient Type attribute describes the type of patient admitted to the system.

```
Attribute ID:      NOM_ATTR_PT_TYPE
Attribute Type:    PatientType
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

The Patient Type is defined as follows:

```
typedef u_16       PatientType;
#define     PAT_TYPE_UNSPECIFIED   0
#define     ADULT                  1
#define     PEDIATRIC              2
#define     NEONATAL               3
```

The Patient Type can be set by the user in the Admit/Discharge dialog (Patient Cat.).

Attribute: Patient Paced Mode

The Patient Paced Mode attribute indicates whether the patient is paced or not.

```
Attribute ID:      NOM_ATTR_PT_PACED_MOD
Attribute Type:    PatPacedMode
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

The *PatPacedMode* is defined as follows:

```
typedef u_16       PatPacedMode;
#define     PAT_NOT_PACED          0
#define     PAT_PACED_GEN          1
```

Values greater one are reserved to indicate special paced modes. The Computer Client should test for "== 0" or "!= 0".

Attribute: Given Name

The Given Name attribute contains the first name of the patient.

```
Attribute ID:      NOM_ATTR_PT_NAME_GIVEN
Attribute Type:    String
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

Currently, the Given Name can be up to 19 characters long, including the terminating ’\0’.

Attribute: Middle Name

The Middle Name attribute contains the middle name of the patient.

```
Attribute ID:      NOM_ATTR_PT_NAME_MIDDLE
Attribute Type:    String
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

Currently, the Middle Name can be up to 19 characters long, including terminating ’\0’.

Attribute: Family Name

The Family Name attribute contains the last name of the patient.

```
Attribute ID:      NOM_ATTR_PT_NAME_FAMILY
Attribute Type:    String
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

Currently, the Family Name can be up to 19 characters long, including terminating '\0'.

Attribute: Patient ID

The Patient ID attribute contains the ID of the patient.

```
Attribute ID:      NOM_ATTR_PT_LIFETIME_ID (identical to previous NOM_ATTR_PT_ID)
Attribute Type:    String
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

Currently, the Patient ID (Medical Record Number - MRN) can be up to 17 characters long, including the terminating '\0'.

Since Rev. G Intellivue the Patient ID is called Lifetime ID.

Attribute: Encounter ID

The Encounter ID attribute contains the ID of the current visit of the patient.

```
Attribute ID:      NOM_ATTR_PT_ENCOUNTER_ID
Attribute Type:    String
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

Currently, the Encounter ID can be up to 17 characters long, including the terminating '\0'.

Attribute: Patient Sex

The Patient Sex attribute contains the sex of the patient.

```
Attribute ID:      NOM_ATTR_PT_SEX
Attribute Type:    PatientSex
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

The *PatientSex* is described as follows:

```
typedef u_16       PatientSex;
#define     SEX_UNKNOWN           0
#define     MALE                  1
#define     FEMALE                2
#define     SEX_UNSPECIFIED       9
```

The values have the following meaning:

SEX_UNKNOWN: Patient sex is not known

MALE: Patient is male

FEMALE: Patient is female

SEX_UNSPECIFIED: Patient sex is not specified

Attribute: Date of Birth

The Date of Birth attribute contains the Date of Birth of the patient.

```
Attribute ID:      NOM_ATTR_PT_DOB
Attribute Type:    AbsoluteTime
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

Attribute: Patient Height

The Patient Height attribute contains the height of the patient.

```
Attribute ID:      NOM_ATTR_PT_HEIGHT
Attribute Type:    PatMeasure
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

The *PatMeasure* is defined as follows:

```
typedef struct {
    FLOATType        value;
    OIDType          m_unit;
} PatMeasure;
```

The *value* contains the actual value of the attribute and the *m_units* indicates the unit of measurement for the *value*.

Attribute: Patient Weight

The Patient Height attribute contains the weight of the patient.

```
Attribute ID:      NOM_ATTR_PT_WEIGHT
Attribute Type:    PatMeasure
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

Attribute: Patient Age

The Patient Age attribute contains the age of the patient.

```
Attribute ID:      NOM_ATTR_PT_AGE
Attribute Type:    PatMeasure
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

Attribute: Patient BSA

The Patient BSA attribute contains the body surface area of the patient.

```
Attribute ID:      NOM_ATTR_PT_BSA
Attribute Type:    PatMeasure
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

Attribute: Patient BSA Formula

The Patient BSA Formula attribute describes the formula which is used for the calculation of the patient body surface area.

```
Attribute ID:      NOM_ATTR_PT_BSA_FORMULA
Attribute Type:    PatBsaFormula
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

The *PatBsaFormula* is described as follows:

```
typedef u_16       PtBsaFormula;
#define     BSA_FORMULA_UNSPEC     0
#define     BSA_FORMULA_BOYD       1
#define     BSA_FORMULA_DUBOIS     2
```

The values have the following meaning:

BSA_FORMULA_UNSPEC: Formula not specified

BSA_FORMULA_BOYD: BSA calculation according to Boyd

BDA_FORMULA_DUBOIS:: BSA calculation according to Dubois

Attribute: Notes1

The Notes1 attribute provides additional information about the patient.

```
Attribute ID:      NOM_ATTR_PT_NOTES1
Attribute Type:    String
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

Currently, the Notes1 field can be up to 31 characters long, including the terminating '\0'.

Attribute: Notes2

The Notes2 attribute provides additional information about the patient.

```
Attribute ID:      NOM_ATTR_PT_NOTES2
Attribute Type:    String
Attribute Groups:  Patient Demographics Attribute Group
Availability:      Optional
```

Currently, the Notes2 field can be up to 31 characters long, including the terminating '\0'.

Attribute Groups

The attributes of the Patient Demographics object are arranged in the following attribute groups:

```
Attribute Group:   Patient Demographics Attribute Group
Group ID:          NOM_ATTR_GRP_PT_DEMOG
Description:       Attributes containing patient information
Attributes:        all attributes
```

Patient Conflict Handling

The patient information is stored in the monitor, the measurement server and the central station (if present). This can lead to patient conflicts when the patient information in these locations differ. If the IntelliVue monitor detects a patient conflict, it will display a “Patient Selection” window which allows the user to resolve the conflict.

In the case of a patient conflict, the behavior of the Data Export software is as follows:

- If the Patient Type or Patient Paced Mode attribute is different, the data from the measurement server is considered as relevant.
- If the patient is different (devices have been disconnected and a new patient has been admitted), the Patient Type and Patient Paced Mode information from the measurement server is exported. The other attributes are cleared and the Family Name attribute is set to “???”.

Connect Indication Attributes

This section describes the attributes contained in the Connect Indication Message.

Attribute: System Type

The System Type attribute describes the type of the system (e.g. Monitor).

```
Attribute ID:      NOM_ATTR_SYS_TYPE
Attribute Type:    TYPE (see Definitions Shared by Protocols)
Attribute Groups:  -
Availability:      -
```

Attribute: Protocol Support

The Protocol Support contains an entry for each protocol supported on the network interface.

```
Attribute ID:      NOM_ATTR_PCOL_SUPPORT
Attribute Type:    ProtoSupport
Attribute Groups:  -
Availability:      -
```

The *ProtoSupport* is defined as follows:

```
typedef struct {
    u_16            count;
    u_16            length;
    ProtoSupportEntry value[1];
} ProtoSupport;

typedef struct {
    ApplProtoId     appl_proto;
    TransProtoId    trans_proto;
    u_16            port_number;
    ProtoOptions    options;
} ProtoSupportEntry;

typedef u_16 ApplProtoId;
#define     AP_ID_ACSE             1
#define     AP_ID_DATA_OUT         5

typedef u_16 TransProtoId;
#define     TP_ID_UDP              1

typedef u_16            ProtoOptions;
#define     P_OPT_WIRELESS         0x8000
```

The Computer Client should parse the available protocols and search for the AP_ID_DATA_OUT. This entry specifies the port for the Data Export Protocol. The corresponding Association Control Protocol runs on the same port.

The Computer Client must only send requests to the port specified for the Data Export Protocol.

Attribute: System Localization

The System Localization attribute describes the handling of natural language items.

```
Attribute ID:      NOM_ATTR_LOCALIZN
Attribute Type:    SystemLocal
Attribute Groups:  -
Availability:      -
```

The *SystemLocal* is defined as follows:

```
typedef struct {
    u_32           syslocal_revision;
    Language       language;
    StringFormat   format;
} SystemLocal;
```

The *syslocal_revision* contains the revision of the text catalog used for internal texts.

The *Language* describes the language used in any String type. It is defined as follows:

```
typedef u_16            Language;
#define     LANGUAGE_UNSPEC       0
#define     ENGLISH               1
#define     GERMAN                2
#define     FRENCH                3
#define     ITALIAN               4
#define     SPANISH               5
#define     DUTCH                 6
#define     SWEDISH               7
#define     FINNISH               8
#define     NORWEG                9
#define     DANISH                10
#define     JAPANESE              11
#define     REP_OF_CHINA          12
#define     PEOPLE_REP_CHINA      13
#define     PORTUGUESE            14
#define     RUSSIAN               15
#define     BYELORUSSIAN          16
#define     UKRAINIAN             17
#define     CROATIAN              18
#define     SERBIAN               19
#define     MACEDONIAN            20
#define     BULGARIAN             21
#define     GREEK                 22
#define     POLISH                23
#define     CZECH                 24
#define     SLOVAK                25
#define     SLOVENIAN             26
#define     HUNGARIAN             27
#define     ROMANIAN              28
#define     TURKISH               29
#define     LATVIAN               30
#define     LITHUANIAN            31
#define     ESTONIAN              32
#define     KOREAN                33
```

The *StringFormat* defines the format used for the *String* data type. The IntelliVue monitor uses 16bit Unicode characters.

```
typedef u_16            StringFormat;
#define                     STRFMT_UNICODE_NT 11
```

Attribute: IP Address Information

The IP Address Information attribute identifies the network interface of the IntelliVue monitor.

```
Attribute ID:      NOM_ATTR_NET_ADDR_INFO
Attribute Type:    IpAddressInfo
Attribute Groups:  -
Availability:      -
```

The *IpAddressInfo* is defined as follows:

```
typedef struct IpAddressInfo {
    MACAddress      mac_address;
    IPAddress       ip_address;
    IPAddress       subnet_mask;
} IpAddressInfo;

typedef struct MACAddress {
    u_8             value[6];
} MACAddress;

typedef struct IPAddress {
    u_8             value[4];
} IPAddress;
```

Partition IDs

The following sections contain a list of identifiers which are used within the IntelliVue monitor. Each identifier is unique within a given partition.

```
#define NOM_PART_OBJ                             1
    /* Object Oriented Elements */
#define NOM_PART_SCADA                           2
    /* Physiological Measurements */
#define NOM_PART_EVT                             3
    /* Events for Alerts */
#define NOM_PART_DIM                             4
    /* Units of Measurement */
#define NOM_PART_PGRP                            6
    /* Identification of Parameter Groups */
#define NOM_PART_INFRASTRUCT                     8
    /* Infrastructure Elements */
#define NOM_PART_EMFC                            1025
    /* EMFC */
#define NOM_PART_SETTINGS                        1026
    /* Settings */
```

Object Classes

The following IDs identify object types. They are taken from the Object Oriented Elements partition. These objects may be the source of alerts (see “Alert Monitor Object” on page 99).

NOM_MOC_VMO VMO	1
NOM_MOC_VMO_METRIC_NU Numeric	6
NOM_MOC_VMO_METRIC_SA_RT Realtime Sample Array	9
NOM_MOC_VMS_MDS MDS	33
NOM_MOC_VMS_MDS_COMPOS_SINGLE_BED Composit Single Bed MDS	35
NOM_MOC_VMS_MDS_SIMP Simple MDS	37
NOM_MOC_BATT Battery	41
NOM_MOC_PT_DEMOG Patient Demographics	42
NOM_MOC_VMO_AL_MON Alert Monitor	54
NOM_ACT_POLL_MDIB_DATA Poll Action	3094
NOM_NOTI_MDS_CREAT MDS Create	3334
NOM_NOTI_CONN_INDIC Connect Indication	3351
NOM_DEV_METER_CONC_SKIN_GAS Skin Gas	4264
NOM_DEV_METER_FLOW_BLD Blood Flow	4284
NOM_DEV_ANALY_CONC_GAS_MULTI_PARAM_MDS Gas Analyzer	4113
NOM_DEV_ANALY_CONC_GAS_MULTI_PARAM_VMD Gas	4114
NOM_DEV_METER_CONC_SKIN_GAS_MDS Skin Gas	4265
NOM_DEV_MON_PHYSIO_MULTI_PARAM_MDS Multi-Param	4429
NOM_DEV_PUMP_INFUS_MDS Pump Infus	4449
NOM_DEV_SYS_PT_VENT_MDS Ventilator	4465
NOM_DEV_SYS_PT_VENT_VMD Ventilator	4466
NOM_DEV_SYS_MULTI_MODAL_MDS Multi-Modal MDS	4493
NOM_DEV_SYS_MULTI_MODAL_VMD Multi-Modal	4494
NOM_DEV_SYS_VS_CONFIG_MDS config MDS	5209
NOM_DEV_SYS_VS_UNCONFIG_MDS unconfig MDS	5213
NOM_DEV_ANALY_SAT_O2_VMD sat O2	4106
NOM_DEV_ANALY_CONC_GAS_MULTI_PARAM_VMD Gas	4114
NOM_DEV_ANALY_FLOW_AWAY_VMD Flow Away	4130
NOM_DEV_ANALY_CARD_OUTPUT_VMD C.O.	4134

NOM_DEV_ANALY_PRESS_BLD_VMD Press	4174
NOM_DEV_ANALY_RESP_RATE_VMD RR	4186
NOM_DEV_CALC_VMD Calculation	4206
NOM_DEV_ECG_VMD ECG	4262
NOM_DEV_METER_CONC_SKIN_GAS_VMD Skin Gas	4266
NOM_DEV_EEG_VMD EEG	4274
NOM_DEV_METER_TEMP_BLD_VMD Blood Temp	4350
NOM_DEV_METER_TEMP_VMD Temp	4366
NOM_DEV_MON_BLD_CHEM_MULTI_PARAM_VMD Bld Chem	4398
NOM_DEV_SYS_PT_VENT_VMD Ventilator	4466
NOM_DEV_SYS_MULTI_MODAL_VMD Multi-Modal	4494
NOM_DEV_SYS_ANESTH_VMD Aneshesia	4506
NOM_DEV_GENERAL_VMD General	5122
NOM_DEV_ECG_RESP_VMD ECG-Resp	5130
NOM_DEV_ARRHY_VMD Arrythmia	5134
NOM_DEV_PULS_VMD Pulse	5138
NOM_DEV_ST_VMD ST	5142
NOM_DEV_CO2_VMD CO2	5146
NOM_DEV_PRESS_BLD_NONINV_VMD Noninv Press	5150
NOM_DEV_CEREB_PERF_VMD Cereb Perf	5154
NOM_DEV_CO2_CTS_VMD CO2 CTS	5158
NOM_DEV_CO2_TCUT_VMD TcCO2	5162
NOM_DEV_O2_VMD O2	5166
NOM_DEV_O2_CTS_VMD CTS	5170
NOM_DEV_O2_TCUT_VMD Tc02	5174
NOM_DEV_TEMP_DIFF_VMD Diff Temp	5178
NOM_DEV_CNTRL_VMD Control	5182
NOM_DEV_WEDGE_VMD Wedge	5190
NOM_DEV_O2_VEN_SAT_VMD O2 Vent Sat	5194
NOM_DEV_CARD_RATE_VMD HR	5202
NOM_DEV_PLETH_VMD Pleth	5238
NOM_SAT_O2_TONE_FREQ Private Attribute	61448

NOM_OBJ_HIF_KEY	Key	61584
NOM_OBJ_DISP	Display	61616
NOM_OBJ_SOUND_GEN	Sound Generator	61648
NOM_OBJ_SETTING	Setting	61649
NOM_OBJ_PRINTER	Printer	61650
NOM_OBJ_EVENT	Event	61683
NOM_OBJ_BATT_CHARGER	Battery Charger	61690
NOM_OBJ_ECG_OUT	ECG out	61691
NOM_OBJ_INPUT_DEV	Input Device	61692
NOM_OBJ_NETWORK	Network	61693
NOM_OBJ_QUICKLINK	Quicklink Bar	61694
NOM_OBJ_SPEAKER	Speaker	61695
NOM_OBJ_PUMP	Pump	61716
NOM_OBJ_IR	IR	61717
NOM_ACT_POLL_MDIB_DATA_EXT	Extended Poll Action	61755
NOM_DEV_ANALY_PULS_CONT	Puls Cont	61800
NOM_DEV_ANALY_BISPECTRAL_INDEX_VMD	BIS	61806
NOM_DEV_HIRES_TREND	Hires Trend	61820
NOM_DEV_HIRES_TREND_MDS	Hires Trend	61821
NOM_DEV_HIRES_TREND_VMD	Hires Trend	61822
NOM_DEV_MON_PT_EVENT_VMD	Events	61826
NOM_DEV_DERIVED_MSMT	Derived Measurement	61828
NOM_DEV_DERIVED_MSMT_MDS	Derived Measurement	61829
NOM_DEV_DERIVED_MSMT_VMD	Derived Measurement	61830
NOM_OBJ_SENSOR	Sensor	61902
NOM_OBJ_XDUCR	Transducer	61903
NOM_OBJ_CHAN_1	Channel 1	61916
NOM_OBJ_CHAN_2	Channel 2	61917
NOM_OBJ_AWAY_AGENT_1	Agent 1	61918
NOM_OBJ_AWAY_AGENT_2	Agent 2	61919
NOM_OBJ_HIF_MOUSE	MOUSE	61983
NOM_OBJ_HIF_TOUCH	TOUCH	61984

NOM_OBJ_HIF_SPEEDPOINT Speedpoint	61985
NOM_OBJ_HIF_ALARMBOX Alarmbox	61986
NOM_OBJ_BUS_I2C I2C Bus	61987
NOM_OBJ_CPU_SEC 2nd CPU	61988
NOM_OBJ_LED LED	61990
NOM_OBJ_RELAY Relay	61991
NOM_OBJ_BATT_1 Battery 1	61996
NOM_OBJ_BATT_2 Battery 2	61997
NOM_OBJ_DISP_SEC 2nd Display	61998
NOM_OBJ_AGM AGM	61999
NOM_OBJ_TELEMON TeleMon	62014
NOM_OBJ_XMTR Transmitter	62015
NOM_OBJ_CABLE Cable	62016
NOM_OBJ_TELEMETRY_XMTR Telemetry Transmitter	62053
NOM_OBJ_MMS MMS	62070
NOM_OBJ_DISP_THIRD Third Display	62073
NOM_OBJ_BATT Battery	62078
NOM_OBJ_BATT_TELE Battery Tele	62091
NOM_DEV_PROT_WATCH_CHAN Protocol Watch generic	62095
NOM_OBJ_PROT_WATCH_1 Protocol Watch Protocol No. 1	62097
NOM_OBJ_PROT_WATCH_2 Protocol Watch Protocol No. 2	62098
NOM_OBJ_PROT_WATCH_3 Protocol Watch Protocol No. 3	62099
NOM_OBJ_ECG_SYNC ECG Sync	62147
NOM_DEV_METAB_VMD Metabolism	62162
NOM_OBJ_SENSOR_O2_CO2 SENSOR O2 CO2	62165
NOM_OBJ_SRR_IF_1 SRR Interface 1	62208
NOM_OBJ_DISP_REMOTE REMOTE DISPLAY	62228

Physiological Identifier

A Physiological Identifier denotes the origin of a physiological measurement. The identifiers are located in the SCADA partition. The Physiological Identifier is transmitted as part of the numeric or wave observed value. The Physiological Identifier may not be unique. However, it is guaranteed that the Label ID is unique. The Label ID is mapped to a Label String based on the text catalogue (see “Attribute: System Localization” on page 92). Note that the mapping listed below may not be complete and is subject to changes and additions, due to revision changes and additions from additional interfaced devices. The table below should be viewed as an example.

The list below shows the numerics and waves which are supported by the monitor. The numerics and waves are sorted according to their internal priority, i.e. numerics or waves with a higher priority are listed first. This information depends heavily on the software revision of the monitor and the connected devices. Especially data coming from a VueLink module depends on the version of the VueLink driver and the specification of the connected external device. The list contains the possible unit codes for the numerics and waves. The unit codes for numerics/waves acquired through data import interfaces (e.g. VueLink) are not documented, because this data depends on the implementation of the specific data import driver.

For a given software revision, the IntelliVue monitor may not export all of the numerics specified below. The IntelliVue monitor may export numerics, which are not specified here. If a numeric is exported also depends on the configuration of the monitor. In general, a numeric will only be available if the required measurement module is connected and if the specific measurement is activated. Some measurements require the presents of more than one measurement module or special configuration steps may be necessary to activate the measurement.

Numerics

HR	Heart Rate	
	Label:	
	NLS_NOM_ECG_CARD_BEAT_RATE	0x00024182
	Observed Value:	
	NOM_ECG_CARD_BEAT_RATE	0x4182
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
btbHR	Cardiac Beat-to-Beat Rate	
	Label:	
	NLS_NOM_ECG_CARD_BEAT_RATE_BTB	0x0002418A
	Observed Value:	
	NOM_ECG_CARD_BEAT_RATE_BTB	0x418A
PVC	Premature Ventricular Contractions	
	Label:	
	NLS_NOM_ECG_V_P_C_CNT	0x00024261
	Observed Value:	
	NOM_ECG_V_P_C_CNT	0x4261
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
ST	ST generic label	
	Label:	
	NLS_NOM_ECG_AMPL_ST	0x00020300
	Compound Observed Value:	
	NOM_ECG_AMPL_ST_I	0x0301
	NOM_ECG_AMPL_ST_II	0x0302
	NOM_ECG_AMPL_ST_III	0x033D
	NOM_ECG_AMPL_ST_AVR	0x033E
	NOM_ECG_AMPL_ST_AVL	0x033F
	NOM_ECG_AMPL_ST_AVF	0x0340
	NOM_ECG_AMPL_ST_V	0x0343
	NOM_ECG_AMPL_ST_MCL	0x034B

	NOM_ECG_AMPL_ST_V1	0x0303
	NOM_ECG_AMPL_ST_V2	0x0304
	NOM_ECG_AMPL_ST_V3	0x0305
	NOM_ECG_AMPL_ST_V4	0x0306
	NOM_ECG_AMPL_ST_V5	0x0307
	NOM_ECG_AMPL_ST_V6	0x0308
	NOM_ECG_AMPL_ST_AS	0x0365
	NOM_ECG_AMPL_ST_ES	0x0364
	NOM_ECG_AMPL_ST_AI	0x0366
	Units:	
	NOM_DIM_MILLI_M	0x0512
STindx	ST Index	
	Label:	
	NLS_NOM_ECG_AMPL_ST_INDEX	0x0002F03D
	Observed Value:	
	NOM_ECG_AMPL_ST_INDEX	0xF03D
	Units:	
	NOM_DIM_MILLI_M	0x0512
QTc		
	Label:	
	NLS_NOM_ECG_TIME_PD_QTc	0x00023F24
	Observed Value:	
	NOM_ECG_TIME_PD_QTc	0x3F24
	Units:	
	NOM_DIM_MILLI_SEC	0x0892
DeltaQTc		
	Label:	
	NLS_NOM_ECG_TIME_PD_QTc_DELTA	0x0002F156
	Observed Value:	
	NOM_ECG_TIME_PD_QTc_DELTA	0xF156
	Units:	
	NOM_DIM_MILLI_SEC	0x0892
QT		
	Label:	
	NLS_NOM_ECG_TIME_PD_QT_GL	0x00023F20
	Observed Value:	
	NOM_ECG_TIME_PD_QT_GL	0x3F20
	Units:	
	NOM_DIM_MILLI_SEC	0x0892
QT-HR	QT HEARTRATE	
	Label:	
	NLS_NOM_ECG_TIME_PD_QT_HEART_RATE	0x0002F154
	Observed Value:	
	NOM_ECG_TIME_PD_QT_HEART_RATE	0xF154
	Units:	
	NOM_DIM_BEATS_PER_MIN	
QT Bsl		
	Label:	
	NLS_NOM_ECG_TIME_PD_QT_BASELINE	0x0002F155
	Observed Value:	
	NOM_ECG_TIME_PD_QT_BASELINE	0xF155
	Units:	
	NOM_DIM_MILLI_SEC	0x0892
QTHRBl	QT BASELINE HEARTRATE	
	Label:	
	NLS_NOM_ECG_TIME_PD_QT_BASELINE_HEART_RATE	0x0002F157
	Observed Value:	
	NOM_ECG_TIME_PD_QT_BASELINE_HEART_RATE	0xF157
	Units:	
	NOM_DIM_MILLI_SEC	0x0892
Pulse	Pulse Rate	
	Label:	
	NLS_NOM_PULS_RATE	0x0002480A
	Observed Value:	
	NOM_PULS_RATE	0x480A

	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
SpO2	Arterial Oxigen Saturation	
	Label:	
	NLS_NOM_PULS_OXIM_SAT_O2	0x00024BB8
	Observed Value:	
	NOM_PULS_OXIM_SAT_O2	0x4BB8
	Units:	
	NOM_DIM_PERCENT	0x0220
Pulse	Pulse Rate from Plethysmogram	
	Label:	
	NLS_NOM_PULS_OXIM_PULS_RATE	0x00024822
	Observed Value:	
	NOM_PLETH_PULS_RATE	0x4822
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
SpO2pr	Oxigen Saturation	
	Label:	
	NLS_NOM_PULS_OXIM_SAT_O2_PRE_DUCTAL	0x0002F1C0
	Observed Value:	
	NOM_PULS_OXIM_SAT_O2_PRE_DUCTAL	0xF1C0
	Units:	
	NOM_DIM_PERCENT	0x0220
Pulse	Pulse Rate from Plethysmogram (pre ductal)	
	Label:	
	NLS_SPO2_NAMES_PULS_OXIM_PULS_RATE_PRE_DUCTAL	0x8015543D
	Observed Value:	
	NOM_PLETH_PULS_RATE	0x4822
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
SpO2po	Oxigen Saturation	
	Label:	
	NLS_NOM_PULS_OXIM_SAT_O2_POST_DUCTAL	0x0002F1D4
	Observed Value:	
	NOM_PULS_OXIM_SAT_O2_POST_DUCTAL	0xF1D4
	Units:	
	NOM_DIM_PERCENT	0x0220
Pulse	Pulse Rate from Plethysmogram (post ductal)	
	Label:	
	NLS_SPO2_NAMES_PULS_OXIM_PULS_RATE_POST_DUCTAL	0x80155440
	Observed Value:	
	NOM_PLETH_PULS_RATE	0x4822
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
%SpO2T	SpO2 parameter label as sourced by the Telemetry system	
	Label:	
	NLS_NOM_PULS_OXIM_SAT_O2_TELE	0x0002F09C
	Observed Value:	
	NOM_PULS_OXIM_SAT_O2_TELE	0xF09C
	Units:	
	NOM_DIM_PERCENT	0x0220
PulseT	Pulse parameter label as sourced by the Telemetry system	
	Label:	
	NLS_NOM_PULS_OXIM_PULS_RATE_TELE	0x0002F09D
	Observed Value:	
	NOM_PULS_OXIM_PULS_RATE_TELE	0xF09D
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
SpO2 r	Arterial Oxigen Saturation (right)	
	Label:	
	NLS_NOM_PULS_OXIM_SAT_O2_ART_RIGHT	0x00024BCC
	Observed Value:	
	NOM_PULS_OXIM_SAT_O2_ART_RIGHT	0x4BCC
	Units:	
	NOM_DIM_PERCENT	0x0220

```
Pulse       Pulse Rate from Plethysmogram (right)
            Label:
            NLS_SPO2_NAMES_PULS_OXIM_PULS_RATE_RIGHT                    0x80155402
            Observed Value:
            NOM_PLETH_PULS_RATE                                         0x4822
            Units:
            NOM_DIM_BEAT_PER_MIN                                        0x0AA0
SpO2 l      Arterial Oxigen Saturation (left)
            Label:
            NLS_NOM_PULS_OXIM_SAT_O2_ART_LEFT                           0x00024BC8
            Observed Value:
            NOM_PULS_OXIM_SAT_O2_ART_LEFT                               0x4BC8
            Units:
            NOM_DIM_PERCENT                                             0x0220
Pulse       Pulse Rate from Plethysmogram (left)
            Label:
            NLS_SPO2_NAMES_PULS_OXIM_PULS_RATE_LEFT                     0x80155401
            Observed Value:
            NOM_PLETH_PULS_RATE                                         0x4822
            Units:
            NOM_DIM_BEAT_PER_MIN                                        0x0AA0
DeltaSpO2   Difference between two SpO2 Values (like Left - Right)
            Label:
            NLS_NOM_PULS_OXIM_SAT_O2_DIFF                               0x00024BC4
            Observed Value:
            NOM_PULS_OXIM_SAT_O2_DIFF                                   0x4BC4
            Units:
            NOM_DIM_PERCENT                                             0x0220
Perf        Perfusion Indicator
            Label:
            NLS_NOM_PULS_OXIM_PERF_REL                                  0x00024BB0
            Observed Value:
            NOM_PULS_OXIM_PERF_REL                                      0x4BB0
            Units:
            NOM_DIM_DIMLESS                                             0x0200
PerfPr      Relative Perfusion Left
            Label:
            NLS_NOM_PULS_OXIM_PERF_REL_PRE_DUCTAL                       0x0002F22C
            Observed Value:
            NOM_PULS_OXIM_PERF_REL_PRE_DUCTAL                           0xF22C
            Units:
            NOM_DIM_DIMLESS                                             0x0200
PerfPo      Relative Perfusion Left
            Label:
            NLS_NOM_PULS_OXIM_PERF_REL_POST_DUCTAL                      0x0002F1DC
            Observed Value:
            NOM_PULS_OXIM_PERF_REL_POST_DUCTAL                          0xF1DC
            Units:
            NOM_DIM_DIMLESS                                             0x0200
Perf T      Perf  from Telemetry
            Label:
            NLS_NOM_PULS_OXIM_PERF_REL_TELE                             0x0002F12C
            Observed Value:
            NOM_PULS_OXIM_PERF_REL_TELE                                 0xF12C
            Units:
            NOM_DIM_DIMLESS                                             0x0200
Perf r      Relative Perfusion Right label
            Label:
            NLS_NOM_PULS_OXIM_PERF_REL_RIGHT                            0x0002F08B
            Observed Value:
            NOM_PULS_OXIM_PERF_REL_RIGHT                                0xF08B
            Units:
            NOM_DIM_DIMLESS                                             0x0200
Perf l      Relative Perfusion Left
            Label:
```

	NLS_NOM_PULS_OXIM_PERF_REL_LEFT	0x0002F08A
	Observed Value:	
	NOM_PULS_OXIM_PERF_REL_LEFT	0xF08A
	Units:	
	NOM_DIM_DIMLESS	0x0200
NBP	non-invasive blood pressure	
	Label:	
	NLS_NOM_PRESS_BLD_NONINV	0x00024A04
	Observed Value (from VueLink):	
	NOM_PRESS_BLD_NONINV	0x4A04
	Compound Observed Value:	
	NOM_PRESS_BLD_NONINV_SYS	0x4A05
	NOM_PRESS_BLD_NONINV_DIA	0x4A06
	NOM_PRESS_BLD_NONINV_MEAN	0x4A07
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse from NBP	
	Label:	
	NLS_NOM_PRESS_BLD_NONINV_PULS_RATE	0x0002F0E5
	Observed Value:	
	NOM_PRESS_BLD_NONINV_PULS_RATE	0xF0E5
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
ABP	Arterial Blood Pressure (ABP)	
	Label:	
	NLS_NOM_PRESS_BLD_ART_ABP	0x00024A14
	Observed Value (from VueLink):	
	NOM_PRESS_BLD_ART_ABP	0x4A14
	Compound Observed Value:	
	NOM_PRESS_BLD_ART_ABP_SYS	0x4A15
	NOM_PRESS_BLD_ART_ABP_DIA	0x4A16
	NOM_PRESS_BLD_ART_ABP_MEAN	0x4A17
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from ABP	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_ABP	0x80035402
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
ART	Arterial Blood Pressure (ART)	
	Label:	
	NLS_NOM_PRESS_BLD_ART	0x00024A10
	Observed Value (from VueLink):	
	NOM_PRESS_BLD_ART	0x4A10
	Compound Observed Value:	
	NOM_PRESS_BLD_ART_SYS	0x4A11
	NOM_PRESS_BLD_ART_DIA	0x4A12
	NOM_PRESS_BLD_ART_MEAN	0x4A13
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from ART	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_ART	0x80035403
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
Ao	Arterial Blood Pressure in the Aorta (Ao)	
	Label:	
	NLS_NOM_PRESS_BLD_AORT	0x00024A0C

	Observed Value (from VueLink):	
	NOM_PRESS_BLD_AORT	0x4A0C
	Compound Observed Value:	
	NOM_PRESS_BLD_AORT_SYS	0x4A0D
	NOM_PRESS_BLD_AORT_DIA	0x4A0E
	NOM_PRESS_BLD_AORT_MEAN	0x4A0F
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from Ao	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_AO	0x80035404
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
PAP	Pulmonary Arterial Pressure (PAP)	
	Label:	
	NLS_NOM_PRESS_BLD_ART_PULM	0x00024A1C
	Observed Value (from VueLink):	
	NOM_PRESS_BLD_ART_PULM	0x4A1C
	Compound Observed Value:	
	NOM_PRESS_BLD_ART_PULM_SYS	0x4A1D
	NOM_PRESS_BLD_ART_PULM_DIA	0x4A1E
	NOM_PRESS_BLD_ART_PULM_MEAN	0x4A1F
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from PAP	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_PAP	0x80035405
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
CVP	Central Venous Pressure (CVP)	
	Label:	
	NLS_NOM_PRESS_BLD_VEN_CENT	0x00024A44
	Observed Value (from VueLink):	
	NOM_PRESS_BLD_VEN_CENT	0x4A44
	Compound Observed Value:	
	NOM_PRESS_BLD_VEN_CENT_SYS	0x4A45
	NOM_PRESS_BLD_VEN_CENT_DIA	0x4A46
	NOM_PRESS_BLD_VEN_CENT_MEAN	0x4A47
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from CVP	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_CVP	0x80035406
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
RAP	Right Atrial Pressure (RAP)	
	Label:	
	NLS_NOM_PRESS_BLD_ATR_RIGHT	0x00024A34
	Observed Value (from VueLink):	
	NOM_PRESS_BLD_ATR_RIGHT	0x4A34
	Compound Observed Value:	
	NOM_PRESS_BLD_ATR_RIGHT_SYS	0x4A35
	NOM_PRESS_BLD_ATR_RIGHT_DIA	0x4A36
	NOM_PRESS_BLD_ATR_RIGHT_MEAN	0x4A37
	Units:	
	NOM_DIM_MMHG	0x0F20

	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from RAP	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_RAP	0x80035407
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
LAP	Left Atrial Pressure (LAP)	
	Label:	
	NLS_NOM_PRESS_BLD_ATR_LEFT	0x00024A30
	Observed Value (from VueLink):	
	NOM_PRESS_BLD_ATR_LEFT	0x4A30
	Compound Observed Value:	
	NOM_PRESS_BLD_ATR_LEFT_SYS	0x4A31
	NOM_PRESS_BLD_ATR_LEFT_DIA	0x4A32
	NOM_PRESS_BLD_ATR_LEFT_MEAN	0x4A33
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from LAP	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_LAP	0x80035408
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
ICP	Intra-cranial Pressure (ICP)	
	Label:	
	NLS_NOM_PRESS_INTRA_CRAN	0x00025808
	Observed Value (from VueLink):	
	NOM_PRESS_INTRA_CRAN	0x5808
	Compound Observed Value:	
	NOM_PRESS_INTRA_CRAN_SYS	0x5809
	NOM_PRESS_INTRA_CRAN_DIA	0x580A
	NOM_PRESS_INTRA_CRAN_MEAN	0x580B
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from ICP	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_ICP	0x80035409
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
UAP	Umbilical Arterial Pressure (UAP)	
	Label:	
	NLS_NOM_PRESS_BLD_ART_UMB	0x00024A28
	Observed Value (from VueLink):	
	NOM_PRESS_BLD_ART_UMB	0x4A28
	Compound Observed Value:	
	NOM_PRESS_BLD_ART_UMB_SYS	0x4A29
	NOM_PRESS_BLD_ART_UMB_DIA	0x4A2A
	NOM_PRESS_BLD_ART_UMB_MEAN	0x4A2B
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from UAP	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_UAP	0x8003540A
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0

UVP	Umbilical Venous Pressure (UVP)	
	Label:	
	NLS_NOM_PRESS_BLD_VEN_UMB	0x00024A48
	Observed Value (from VueLink):	
	NOM_PRESS_BLD_VEN_UMB	0x4A48
	Compound Observed Value:	
	NOM_PRESS_BLD_VEN_UMB_SYS	0x4A49
	NOM_PRESS_BLD_VEN_UMB_DIA	0x4A4A
	NOM_PRESS_BLD_VEN_UMB_MEAN	0x4A4B
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from UVP	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_UVP	0x8003540B
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
FAP	Femoral Arterial Pressure (FAP)	
	Label:	
	NLS_NOM_PRESS_BLD_ART_FEMORAL	0x0002F0BC
	Compound Observed Value:	
	NOM_PRESS_BLD_ART_FEMORAL_SYS	0xF0BD
	NOM_PRESS_BLD_ART_FEMORAL_DIA	0xF0BE
	NOM_PRESS_BLD_ART_FEMORAL_MEAN	0xF0BF
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from FAP	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_FAP	0x80035434
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
BAP	Brachial Arterial Blood Pressure (BAP)	
	Label:	
	NLS_NOM_PRESS_BLD_ART_BRACHIAL	0x0002F0C0
	Compound Observed Value:	
	NOM_PRESS_BLD_ART_BRACHIAL_SYS	0xF0C1
	NOM_PRESS_BLD_ART_BRACHIAL_DIA	0xF0C2
	NOM_PRESS_BLD_ART_BRACHIAL_MEAN	0xF0C3
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from BAP	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_BAP	0x80035437
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
IC1	Intracranial Pressure 1 (IC1)	
	Label:	
	NLS_NOM_PRESS_INTRA_CRAN_1	0x0002F0B4
	Compound Observed Value:	
	NOM_PRESS_INTRA_CRAN_1_DIA	0xF0B6
	NOM_PRESS_INTRA_CRAN_1_SYS	0xF0B5
	NOM_PRESS_INTRA_CRAN_1_MEAN	0xF0B7
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from IC1	
	Label:	

	NLS_PRESS_NAMES_PULSE_FROM_IC1	0x8003542E
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
IC2	Intracranial Pressure 2 (IC2)	
	Label:	
	NLS_NOM_PRESS_INTRA_CRAN_2	0x0002F0B8
	Compound Observed Value:	
	NOM_PRESS_INTRA_CRAN_2_SYS	0xF0B9
	NOM_PRESS_INTRA_CRAN_2_DIA	0xF0BA
	NOM_PRESS_INTRA_CRAN_2_MEAN	0xF0BB
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from IC2	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_IC2	0x80035431
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
P	unspecific pressure	
	Label:	
	NLS_NOM_PRESS_BLD	0x00024A00
	Observed Value (from VueLink):	
	NOM_PRESS_BLD	0x4A00
	Compound Observed Value:	
	NOM_PRESS_BLD_SYS	0x4A01
	NOM_PRESS_BLD_DIA	0x4A02
	NOM_PRESS_BLD_MEAN	0x4A03
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from unspecific Pressure	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_P	0x80035401
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
P1	Generic Pressure 1 (P1)	
	Label:	
	NLS_NOM_PRESS_GEN_1	0x0002F0A4
	Observed Value (from VueLink):	
	NOM_PRESS_GEN_1	0xF0A4
	Compound Observed Value:	
	NOM_PRESS_GEN_1_SYS	0xF0A5
	NOM_PRESS_GEN_1_DIA	0xF0A6
	NOM_PRESS_GEN_1_MEAN	0xF0A7
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from P1	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_P1	0x80035422
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
P2	Generic Pressure 2 (P2)	
	Label:	
	NLS_NOM_PRESS_GEN_2	0x0002F0A8
	Observed Value (from VueLink):	
	NOM_PRESS_GEN_2	0xF0A8

	Compound Observed Value:	
	NOM_PRESS_GEN_2_SYS	0xF0A9
	NOM_PRESS_GEN_2_DIA	0xF0AA
	NOM_PRESS_GEN_2_MEAN	0xF0AB
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from P2	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_P2	0x80035425
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
P3	Generic Pressure 3 (P3)	
	Label:	
	NLS_NOM_PRESS_GEN_3	0x0002F0AC
	Observed Value (from VueLink):	
	NOM_PRESS_GEN_3	0xF0AC
	Compound Observed Value:	
	NOM_PRESS_GEN_3_SYS	0xF0AD
	NOM_PRESS_GEN_3	0xF0AC
	NOM_PRESS_GEN_3_MEAN	0xF0AF
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from P3	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_P3	0x80035428
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
P4	Generic Pressure 4 (P4)	
	Label:	
	NLS_NOM_PRESS_GEN_4	0x0002F0B0
	Observed Value (from VueLink):	
	NOM_PRESS_GEN_4	0xF0B0
	Compound Observed Value:	
	NOM_PRESS_GEN_4_SYS	0xF0B1
	NOM_PRESS_GEN_4_DIA	0xF0B2
	NOM_PRESS_GEN_4_MEAN	0xF0B3
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Pulse	Pulse derived from P4	
	Label:	
	NLS_PRESS_NAMES_PULSE_FROM_P4	0x8003542B
	Observed Value:	
	NOM_PULS_RATE	0x480A
	Units:	
	NOM_DIM_BEAT_PER_MIN	0x0AA0
IUP	Intra-Uterine Pressure	
	Label:	
	NLS_NOM_PRESS_INTRA_UTERAL	0x0002F0D8
	Observed Value:	
	NOM_PRESS_BLD	0x4A00
PAWP	Pulmonary Artery Wedge Pressure	
	Label:	
	NLS_NOM_PRESS_BLD_ART_PULM_WEDGE	0x00024A24
	Observed Value:	
	NOM_PRESS_BLD_ART_PULM_WEDGE	0x4A24
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03

CPP	Cerebral Perfusion Pressure	
	Label:	
	NLS_NOM_PRESS_CEREB_PERF	0x00025804
	Observed Value:	
	NOM_PRESS_CEREB_PERF	0x5804
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
PPV	Pulse Pressure Variation	
	Label:	
	NLS_NOM_PULS_PRESS_VAR	0x0002F0E3
	Observed Value:	
	NOM_PULS_PRESS_VAR	0xF0E3
CCO	Continuous Cardiac Output	
	Label:	
	NLS_NOM_OUTPUT_CARD_CTS	0x00024BDC
	Observed Value:	
	NOM_OUTPUT_CARD_CTS	0x4BDC
	Units:	
	NOM_DIM_X_L_PER_MIN	0x0C00
CCI	Continuous Cardiac Output Index	
	Label:	
	NLS_NOM_OUTPUT_CARD_INDEX_CTS	0x0002F047
	Observed Value:	
	NOM_OUTPUT_CARD_INDEX_CTS	0xF047
	Units:	
	NOM_DIM_X_L_PER_MIN_PER_M_SQ	0x0B20
SV	Stroke Volume	
	Label:	
	NLS_NOM_VOL_BLD_STROKE	0x00024B84
	Observed Value:	
	NOM_VOL_BLD_STROKE	0x4B84
	Units:	
	NOM_DIM_MILLI_L	0x0652
SI	Stroke Index	
	Label:	
	NLS_NOM_VOL_BLD_STROKE_INDEX	0x0002F048
	Observed Value:	
	NOM_VOL_BLD_STROKE_INDEX	0xF048
	Units:	
	NOM_DIM_MILLI_L_PER_M_SQ	0x0592
SVV	Stroke Volume Variation	
	Label:	
	NLS_NOM_VOL_BLD_STROKE_VAR	0x0002F049
	Observed Value:	
	NOM_VOL_BLD_STROKE_VAR	0xF049
	Units:	
	NOM_DIM_PERCENT	0x0220
dPmax	Index of Left Ventricular Contractility	
	Label:	
	NLS_NOM_GRAD_PRESS_BLD_AORT_POS_MAX	0x00024C25
	Observed Value:	
	NOM_GRAD_PRESS_BLD_AORT_POS_MAX	0x4C25
C.O.	Cardiac Output	
	Label:	
	NLS_NOM_OUTPUT_CARD	0x00024B04
	Observed Value:	
	NOM_OUTPUT_CARD	0x4B04
	Units:	
	NOM_DIM_X_L_PER_MIN	0x0C00
C.I.	Cardiac Index	
	Label:	
	NLS_NOM_OUTPUT_CARD_INDEX	0x0002490C
	Observed Value:	
	NOM_OUTPUT_CARD_INDEX	0x490C

	Units:	
	NOM_DIM_X_L_PER_MIN_PER_M_SQ	0x0B20
ITBV	Intrathoracic Blood Volume	
	Label:	
	NLS_NOM_VOL_BLD_INTRA_THOR	0x0002F040
	Observed Value:	
	NOM_VOL_BLD_INTRA_THOR	0xF040
	Units:	
	NOM_DIM_MILLI_L	0x0652
ITBVI	Intrathoracic Blood Volume Index	
	Label:	
	NLS_NOM_VOL_BLD_INTRA_THOR_INDEX	0x0002F041
	Observed Value:	
	NOM_VOL_BLD_INTRA_THOR_INDEX	0xF041
	Units:	
	NOM_DIM_MILLI_L_PER_M_SQ	0x0592
EVLW	Extravascular Lung Water	
	Label:	
	NLS_NOM_VOL_LUNG_WATER_EXTRA_VASC	0x0002F042
	Observed Value:	
	NOM_VOL_LUNG_WATER_EXTRA_VASC	0xF042
	Units:	
	NOM_DIM_MILLI_L	0x0652
EVLWI	Extravascular Lung Water Index	
	Label:	
	NLS_NOM_VOL_LUNG_WATER_EXTRA_VASC_INDEX	0x0002F043
	Observed Value:	
	NOM_VOL_LUNG_WATER_EXTRA_VASC_INDEX	0xF043
	Units:	
	NOM_DIM_MILLI_L_PER_KG	0x0C72
GEDV	Global End Diastolic Volume	
	Label:	
	NLS_NOM_VOL_GLOBAL_END_DIA	0x0002F044
	Observed Value:	
	NOM_VOL_GLOBAL_END_DIA	0xF044
	Units:	
	NOM_DIM_MILLI_L	0x0652
GEDVI	Global End Diastolic Volume Index	
	Label:	
	NLS_NOM_VOL_GLOBAL_END_DIA_INDEX	0x0002F045
	Observed Value:	
	NOM_VOL_GLOBAL_END_DIA_INDEX	0xF045
	Units:	
	NOM_DIM_MILLI_L_PER_M_SQ	0x0592
CFI	Cardiac Function Index	
	Label:	
	NLS_NOM_CARD_FUNC_INDEX	0x0002F046
	Observed Value:	
	NOM_CARD_FUNC_INDEX	0xF046
	Units:	
	NOM_DIM_DIMLESS	0x0200
PVPI	Pulmonary Vascular Permeability Index	
	Label:	
	NLS_NOM_PERM_VASC_PULM_INDEX	0x0002F106
	Observed Value:	
	NOM_PERM_VASC_PULM_INDEX	0xF106
GEF	Global Ejection Fraction	
	Label:	
	NLS_NOM_FRACT_EJECT	0x0002F105
	Observed Value:	
	NOM_FRACT_EJECT	0xF105
SNR	Signal to Noise ratio	
	Label:	
	NLS_NOM_SNR	0x0002F101
	Observed Value:	

	NOM_SNR	0xF101
RLShnt	Right-to-Left Heart Shunt	
	Label:	
	NLS_NOM_SHUNT_RIGHT_LEFT	0x0002F14A
	Observed Value:	
	NOM_SHUNT_RIGHT_LEFT	0xF14A
	Units:	
	NOM_DIM_MILLI_SECOND	
SaO2	Oxygen Saturation	
	Label:	
	NLS_NOM_SAT_O2_ART	0x00024B34
	Observed Value:	
	NOM_SAT_O2_ART	0x4B34
SvO2	Mixed Venous Oxygen Saturation	
	Label:	
	NLS_NOM_SAT_O2_VEN	0x00024B3C
	Observed Value:	
	NOM_SAT_O2_VEN	0x4B3C
	Units:	
	NOM_DIM_PERCENT	0x0220
ScvO2	Central Venous Oxygen Saturation	
	Label:	
	NLS_NOM_SAT_O2_VEN_CENT	0x0002F100
	Observed Value:	
	NOM_SAT_O2_VEN_CENT	0xF100
SO2	O2 Saturation	
	Label:	
	NLS_NOM_SAT_O2	0x00024B2C
	Observed Value:	
	NOM_SAT_O2	0x4B2C
	Units:	
	NOM_DIM_PERCENT	
SO2 l	Oxygen Saturation Left Side	
	Label:	
	NLS_NOM_SAT_O2_LEFT	0x0002F89D
	Observed Value:	
	NOM_SAT_O2_ART	0x4B34
SO2 r	Oxygen Saturation Right Side	
	Label:	
	NLS_NOM_SAT_O2_RIGHT	0x0002F89E
	Observed Value:	
	NOM_SAT_O2_ART	0x4B34
SO2 1	O2 Saturation 1 (generic)	
	Label:	
	NLS_NOM_SAT_O2_GEN_1	0x0002F962
	Observed Value:	
	NOM_SAT_O2_GEN_1	0xF962
	Units:	
	NOM_DIM_PERCENT	0x0220
SO2 2	O2 Saturation 2 (generic)	
	Label:	
	NLS_NOM_SAT_O2_GEN_2	0x0002F963
	Observed Value:	
	NOM_SAT_O2_GEN_2	0xF963
	Units:	
	NOM_DIM_PERCENT	0x0220
SO2 3	O2 Saturation 3 (generic)	
	Label:	
	NLS_NOM_SAT_O2_GEN_3	0x0002F964
	Observed Value:	
	NOM_SAT_O2_GEN_3	0xF964
	Units:	
	NOM_DIM_PERCENT	0x0220
SO2 4	O2 Saturation 4 (generic)	
	Label:	

	NLS_NOM_SAT_O2_GEN_4	0x0002F965
	Observed Value:	
	NOM_SAT_O2_GEN_4	0xF965
	Units:	
	NOM_DIM_PERCENT	0x0220
LI	Light Intenisty. SvO2	
	Label:	
	NLS_NOM_INTENS_LIGHT	0x0002F072
	Observed Value:	
	NOM_INTENS_LIGHT	0xF072
DO2	Oxygen Availability DO2	
	Label:	
	NLS_NOM_SAT_O2_DELIVER	0x0002F06D
	Observed Value:	
	NOM_SAT_O2_DELIVER	0xF06D
DO2I	Oxygen Availability Index	
	Label:	
	NLS_NOM_SAT_O2_DELIVER_INDEX	0x0002F06E
	Observed Value:	
	NOM_SAT_O2_DELIVER_INDEX	0xF06E
O2ER	Oxygen Extraction Ratio	
	Label:	
	NLS_NOM_RATIO_SAT_O2_CONSUMP_DELIVER	0x0002F06F
	Observed Value:	
	NOM_RATIO_SAT_O2_CONSUMP_DELIVER	0xF06F
Qs/Qt	Percent Alveolarvenous Shunt Qs/Qt	
	Label:	
	NLS_NOM_RATIO_ART_VEN_SHUNT	0x0002F070
	Observed Value:	
	NOM_RATIO_ART_VEN_SHUNT	0xF070
AaDO2	Alveolar- Arterial Oxygen Difference	
	Label:	
	NLS_NOM_SAT_DIFF_O2_ART_ALV	0x00024B40
	Observed Value:	
	NOM_SAT_DIFF_O2_ART_ALV	0x4B40
Sp-vO2	Difference between Spo2 and SvO2	
	Label:	
	NLS_NOM_SAT_DIFF_O2_ART_VEN	0x0002F06C
	Observed Value:	
	NOM_SAT_DIFF_O2_ART_VEN	0xF06C
tcGas	Generic Term for the Transcutaneous Gases	
	Label:	
	NLS_NOM_GAS_TCUT	0x0002F051
	Observed Value:	
	NOM_GAS_TCUT	0xF051
tcpO2	Transcutaneous Oxygen Partial Pressure	
	Label:	
	NLS_NOM_O2_TCUT	0x000250D0
	Observed Value:	
	NOM_O2_TCUT	0x50D0
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
tcpCO2	Transcutaneous Carbon Dioxide Partial Pressure	
	Label:	
	NLS_NOM_CO2_TCUT	0x000250CC
	Observed Value:	
	NOM_CO2_TCUT	0x50CC
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
SitTim	NOM_DIM_MIN	
	Label:	
	NLS_NOM_TIME_TCUT_SENSOR	0x0002F03E
	Observed Value:	

```
            NOM_TIME_TCUT_SENSOR                                         0xF03E
SensrT Sensor Temperature
            Label:
            NLS_NOM_TEMP_TCUT_SENSOR                                     0x0002F03F
            Observed Value:
            NOM_TEMP_TCUT_SENSOR                                         0xF03F
            Units:
            NOM_DIM_DEGC                                                 0x17A0
            NOM_DIM_FAHR                                                 0x1140
HeatPw NOM_DIM_MILLI_WATT
            Label:
            NLS_NOM_HEATING_PWR_TCUT_SENSOR                              0x0002F076
            Observed Value:
            NOM_HEATING_PWR_TCUT_SENSOR                                  0xF076
CO2         CO2 concentration
            Label:
            NLS_NOM_AWAY_CO2                                             0x000250AC
            Observed Value (from VueLink):
            NOM_AWAY_CO2                                                 0x50AC
            Compound Observed Value:
            NOM_AWAY_CO2_ET                                              0x50B0
            NOM_AWAY_CO2_INSP_MIN                                        0x50BA
            Units:
            NOM_DIM_MMHG                                                 0x0F20
            NOM_DIM_PERCENT                                              0x0220
            NOM_DIM_KILO_PASCAL                                          0x0F03
RRspir      Respiration Rate from Spirometry
            Label:
            NLS_NOM_AWAY_RESP_RATE_SPIRO                                 0x0002F0E2
            Observed Value:
            NOM_AWAY_RESP_RATE_SPIRO                                     0xF0E2
            Units:
            NOM_DIM_RESP_PER_MIN                                         0x0AE0
awRR        Airway Respiration Rate
            Label:
            NLS_NOM_AWAY_RESP_RATE                                       0x00025012
            Observed Value:
            NOM_AWAY_RESP_RATE                                           0x5012
            Units:
            NOM_DIM_RESP_PER_MIN                                         0x0AE0
O2          Generic oxigen measurement label
            Label:
            NLS_NOM_CONC_AWAY_O2                                         0x00025164
            Observed Value (from VueLink):
            NOM_CONC_AWAY_O2                                             0x5164
            Compound Observed Value:
            NOM_CONC_AWAY_O2_ET                                          0x5378
            NOM_CONC_AWAY_O2_INSP                                        0x5284
            Units:
            NOM_DIM_MMHG                                                 0x0F20
            NOM_DIM_PERCENT                                              0x0220
            NOM_DIM_KILO_PASCAL                                          0x0F03
FIO2        Fractional Inspired Oxygen FIO2
            Label:
            NLS_NOM_VENT_CONC_AWAY_O2_INSP                               0x00027498
            Observed Value:
            NOM_VENT_CONC_AWAY_O2_INSP                                   0x7498
RR          Respiration Rate
            Label:
            NLS_NOM_RESP_RATE                                            0x0002500A
            Observed Value:
            NOM_RESP_RATE                                                0x500A
            Units:
            NOM_DIM_RESP_PER_MIN                                         0x0AE0
T.I.        Transthoracic Impedance
```

	Label:	
	NLS_NOM_IMPED_TTHOR	0x000250E4
	Observed Value:	
	NOM_IMPED_TTHOR	0x50E4
VCO2	CO2 Production	
	Label:	
	NLS_NOM_FLOW_CO2_PROD_RESP	0x000250E0
	Observed Value:	
	NOM_FLOW_CO2_PROD_RESP	0x50E0
VCO2ti	CO2 Tidal Production	
	Label:	
	NLS_NOM_FLOW_CO2_PROD_RESP_TIDAL	0x0002F882
	Observed Value:	
	NOM_FLOW_CO2_PROD_RESP_TIDAL	0xF882
Pplat	Plateau Pressure	
	Label:	
	NLS_NOM_PRESS_RESP_PLAT	0x000250E8
	Observed Value:	
	NOM_PRESS_RESP_PLAT	0x50E8
AWP	Airway Pressure Wave	
	Label:	
	NLS_NOM_PRESS_AWAY	0x000250F0
	Observed Value:	
	NOM_PRESS_AWAY	0x50F0
AWPmin	Airway Pressure Minimum	
	Label:	
	NLS_NOM_PRESS_AWAY_MIN	0x000250F2
	Observed Value:	
	NOM_PRESS_AWAY_MIN	0x50F2
CPAP	Continuous Positive Airway Pressure	
	Label:	
	NLS_NOM_PRESS_AWAY_CTS_POS	0x000250F4
	Observed Value:	
	NOM_PRESS_AWAY_CTS_POS	0x50F4
iPEEP	Intrinsic PEEP Breathing Pressure	
	Label:	
	NLS_NOM_PRESS_AWAY_END_EXP_POS_INTRINSIC	0x00025100
	Observed Value:	
	NOM_PRESS_AWAY_END_EXP_POS_INTRINSIC	0x5100
AWPin	Airway Pressure Wave - measured in the inspiratory path	
	Label:	
	NLS_NOM_PRESS_AWAY_INSP	0x00025108
	Observed Value:	
	NOM_PRESS_AWAY_INSP	0x5108
PIP	Positive Inspiratory ressure	
	Label:	
	NLS_NOM_PRESS_AWAY_INSP_MAX	0x00025109
	Observed Value:	
	NOM_PRESS_AWAY_INSP_MAX	0x5109
MnAwP	Mean Airway Pressure. Printer Context	
	Label:	
	NLS_NOM_PRESS_AWAY_INSP_MEAN	0x0002510B
	Observed Value:	
	NOM_PRESS_AWAY_INSP_MEAN	0x510B
I:E 1:	Inpired:Expired Ratio	
	Label:	
	NLS_NOM_RATIO_IE	0x00025118
	Observed Value:	
	NOM_RATIO_IE	0x5118
Vd/Vt	Ratio of Deadspace to Tidal Volume Vd/Vt	
	Label:	
	NLS_NOM_RATIO_AWAY_DEADSP_TIDAL	0x0002511C
	Observed Value:	
	NOM_RATIO_AWAY_DEADSP_TIDAL	0x511C
Raw	Static Lung Resistance	

	Label:	
	NLS_NOM_RES_AWAY	0x00025120
	Observed Value:	
	NOM_RES_AWAY	0x5120
TV	Tidal Volume	
	Label:	
	NLS_NOM_VOL_AWAY_TIDAL	0x0002513C
	Observed Value (from VueLink):	
	NOM_VOL_AWAY_TIDAL	0x513C
	Compound Observed Value:	
TVexp	expired Tidal Volume	
	Label:	
	NLS_NOM_VOL_AWAY_EXP_TIDAL	0x0002F0E1
	Observed Value:	
	NOM_VOL_AWAY_EXP_TIDAL	0xF0E1
TVin	inspired Tidal Volume	
	Label:	
	NLS_NOM_VOL_AWAY_INSP_TIDAL	0x0002F0E0
	Observed Value:	
	NOM_VOL_AWAY_INSP_TIDAL	0xF0E0
MINVOL	Airway Minute Volum Inspiratory	
	Label:	
	NLS_NOM_VOL_MINUTE_AWAY	0x00025148
	Observed Value (from VueLink):	
	NOM_VOL_MINUTE_AWAY	0x5148
	Compound Observed Value:	
	NOM_VOL_MINUTE_AWAY_EXP	0x514C
	NOM_VOL_MINUTE_AWAY_INSP	0x5150
	Units:	
	NOM_DIM_X_L_PER_MIN	0x0C00
PlatTi	Plateau Time	
	Label:	
	NLS_NOM_TIME_PD_RESP_PLAT	0x0002F0FF
	Observed Value:	
	NOM_TIME_PD_RESP_PLAT	0xF0FF
SpMV	Spontaneous Minute Volume	
	Label:	
	NLS_NOM_VENT_VOL_MINUTE_AWAY_SPONT	0x0002F091
	Observed Value:	
	NOM_VENT_VOL_MINUTE_AWAY_SPONT	0xF091
DeltaO2	relative Dead Space	
	Label:	
	NLS_NOM_VENT_CONC_AWAY_O2_DELTA	0x00025168
	Observed Value:	
	NOM_VENT_CONC_AWAY_O2_DELTA	0x5168
PECO2	Partial O2 Venous	
	Label:	
	NLS_NOM_VENT_AWAY_CO2_EXP	0x0002517C
	Observed Value:	
	NOM_VENT_AWAY_CO2_EXP	0x517C
AWFin	Airway Flow Wave - measured in the inspiratory path	
	Label:	
	NLS_NOM_VENT_FLOW_INSP	0x0002518C
	Observed Value:	
	NOM_VENT_FLOW_INSP	0x518C
VQI	Ventilation Perfusion Index	
	Label:	
	NLS_NOM_VENT_FLOW_RATIO_PERF_ALV_INDEX	0x00025190
	Observed Value:	
	NOM_VENT_FLOW_RATIO_PERF_ALV_INDEX	0x5190
Poccl	Occlusion Pressure	
	Label:	
	NLS_NOM_VENT_PRESS_OCCL	0x0002519C
	Observed Value:	
	NOM_VENT_PRESS_OCCL	0x519C

```
PEEP        Positive End-Expiratory Pressure PEEP
            Label:
            NLS_NOM_VENT_PRESS_AWAY_END_EXP_POS                      0x000251A8
            Observed Value:
            NOM_VENT_PRESS_AWAY_END_EXP_POS                          0x51A8
Vd          Dead Space Volume Vd
            Label:
            NLS_NOM_VENT_VOL_AWAY_DEADSP                             0x000251B0
            Observed Value:
            NOM_VENT_VOL_AWAY_DEADSP                                 0x51B0
relVd       relative Dead Space
            Label:
            NLS_NOM_VENT_VOL_AWAY_DEADSP_REL                         0x000251B4
            Observed Value:
            NOM_VENT_VOL_AWAY_DEADSP_REL                             0x51B4
TrpVol      Lung Volume Trapped
            Label:
            NLS_NOM_VENT_VOL_LUNG_TRAPD                              0x000251B8
            Observed Value:
            NOM_VENT_VOL_LUNG_TRAPD                                  0x51B8
Leak         Leakage
            Label:
            NLS_NOM_VENT_VOL_LEAK                                    0x00025370
            Observed Value:
            NOM_VENT_VOL_LEAK                                        0x5370
ALVENT      Alveolar Ventilation ALVENT
            Label:
            NLS_NOM_VENT_VOL_LUNG_ALV                                0x00025374
            Observed Value:
            NOM_VENT_VOL_LUNG_ALV                                    0x5374
VC          Vital Lung Capacity
            Label:
            NLS_NOM_CAPAC_VITAL                                      0x00025080
            Observed Value:
            NOM_CAPAC_VITAL                                          0x5080
COMP        generic label Lung Compliance
            Label:
            NLS_NOM_COMPL_LUNG                                       0x00025088
            Observed Value:
            NOM_COMPL_LUNG                                           0x5088
Cdyn        Dynamic Lung Compliance
            Label:
            NLS_NOM_COMPL_LUNG_DYN                                   0x0002508C
            Observed Value:
            NOM_COMPL_LUNG_DYN                                       0x508C
Cstat       Static Lung Compliance
            Label:
            NLS_NOM_COMPL_LUNG_STATIC                                0x00025090
            Observed Value:
            NOM_COMPL_LUNG_STATIC                                    0x5090
PIF         Inspiratory Peak Flow
            Label:
            NLS_NOM_FLOW_AWAY_INSP_MAX                               0x000250DD
            Observed Value:
            NOM_FLOW_AWAY_INSP_MAX                                   0x50DD
PEF         Expiratory Peak Flow
            Label:
            NLS_NOM_FLOW_AWAY_EXP_MAX                                0x000250D9
            Observed Value:
            NOM_FLOW_AWAY_EXP_MAX                                    0x50D9
BIS         Bispectral Index
            Label:
            NLS_NOM_EEG_BISPECTRAL_INDEX                             0x0002F04E
            Observed Value:
            NOM_EEG_BISPECTRAL_INDEX                                 0xF04E
```

	Units:	
	NOM_DIM_DIMLESS	0x0200
SQI	Signal Quality Index	
	Label:	
	NLS_NOM_EEG_BIS_SIG_QUAL_INDEX	0x0002F04D
	Observed Value:	
	NOM_EEG_BIS_SIG_QUAL_INDEX	0xF04D
	Units:	
	NOM_DIM_PERCENT	0x0220
EMG	Electromyography	
	Label:	
	NLS_NOM_EMG_ELEC_POTL_MUSCL	0x0002593C
	Observed Value:	
	NOM_EMG_ELEC_POTL_MUSCL	0x593C
	Units:	
	NOM_DIM_DECIBEL	0x1920
TP	Total Power	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_TOT	0x000259B8
	Observed Value:	
	NOM_EEG_PWR_SPEC_TOT	0x59B8
	Units:	
	NOM_DIM_DECIBEL	0x1920
TP1	Total Power channel 1	
	Label:	
	NLS_EEG_NAMES_CHAN_TP1	0x800F5403
	Observed Value:	
	NOM_EEG_PWR_SPEC_TOT	0x59B8
	Units:	
	NOM_DIM_NANO_WATT	0x0FD4
TP2	Total Power channel 2	
	Label:	
	NLS_EEG_NAMES_CHAN_TP2	0x800F5404
	Observed Value:	
	NOM_EEG_PWR_SPEC_TOT	0x59B8
	Units:	
	NOM_DIM_NANO_WATT	0x0FD4
SR	Suppression Ratio	
	Label:	
	NLS_NOM_EEG_RATIO_SUPPRN	0x0002F04A
	Observed Value:	
	NOM_EEG_RATIO_SUPPRN	0xF04A
	Units:	
	NOM_DIM_PERCENT	0x0220
SEF	Spectral Edge Frequency	
	Label:	
	NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_SPECTRAL_EDGE	0x00025988
	Observed Value:	
	NOM_EEG_FREQ_PWR_SPEC_CRTX_SPECTRAL_EDGE	0x5988
	Units:	
	NOM_DIM_HZ	0x09C0
MDF	Mean Dominant Frequency	
	Label:	
	NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_DOM_MEAN	0x0002597C
	Observed Value:	
	NOM_EEG_FREQ_PWR_SPEC_CRTX_DOM_MEAN	0x597C
	Units:	
	NOM_DIM_HZ	0x09C0
PPF	Peak Power Frequency	
	Label:	
	NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_PEAK	0x00025984
	Observed Value:	
	NOM_EEG_FREQ_PWR_SPEC_CRTX_PEAK	0x5984
	Units:	
	NOM_DIM_HZ	0x09C0

Frequ1	generic label for EEG channel 1	
	Label:	
	NLS_EEG_NAMES_CHAN_FREQ1	0x800F5413
	Compound Observed Value:	
	NOM_EEG_FREQ_PWR_SPEC_CRTX_SPECTRAL_EDGE	0x5988
	NOM_EEG_FREQ_PWR_SPEC_CRTX_DOM_MEAN	0x597C
	NOM_EEG_FREQ_PWR_SPEC_CRTX_PEAK	0x5984
	Units:	
	NOM_DIM_HZ	0x09C0
Frequ2	generic label for EEG channel 2	
	Label:	
	NLS_EEG_NAMES_CHAN_FREQ2	0x800F5414
	Compound Observed Value:	
	NOM_EEG_FREQ_PWR_SPEC_CRTX_SPECTRAL_EDGE	0x5988
	NOM_EEG_FREQ_PWR_SPEC_CRTX_DOM_MEAN	0x597C
	NOM_EEG_FREQ_PWR_SPEC_CRTX_PEAK	0x5984
	Units:	
	NOM_DIM_HZ	0x09C0
Prcnt1	generic label for EEG channel 1	
	Label:	
	NLS_EEG_NAMES_CHAN_PCNT1	0x800F5415
	Compound Observed Value:	
	NOM_EEG_PWR_SPEC_ALPHA_REL	0x59D4
	NOM_EEG_PWR_SPEC_BETA_REL	0x59D8
	NOM_EEG_PWR_SPEC_DELTA_REL	0x59DC
	NOM_EEG_PWR_SPEC_THETA_REL	0x59E0
	Units:	
	NOM_DIM_PERCENT	0x0220
Prcnt2	generic label for EEG channel 2	
	Label:	
	NLS_EEG_NAMES_CHAN_PCNT2	0x800F5416
	Compound Observed Value:	
	NOM_EEG_PWR_SPEC_ALPHA_REL	0x59D4
	NOM_EEG_PWR_SPEC_BETA_REL	0x59D8
	NOM_EEG_PWR_SPEC_DELTA_REL	0x59DC
	NOM_EEG_PWR_SPEC_THETA_REL	0x59E0
	Units:	
	NOM_DIM_PERCENT	0x0220
AAI	A-Line ARX Index	
	Label:	
	NLS_NOM_ELEC_EVOK_POTL_CRTX_ACOUSTIC_AAI	0x0002F873
	Observed Value:	
	NOM_ELEC_EVOK_POTL_CRTX_ACOUSTIC_AAI	0xF873
BSI	Burst Suppression Indicator	
	Label:	
	NLS_NOM_EEG_BURST_SUPPRN_INDEX	0x0002F840
	Observed Value:	
	NOM_EEG_BURST_SUPPRN_INDEX	0xF840
Temp	Unspecific Temperature	
	Label:	
	NLS_NOM_TEMP	0x00024B48
	Observed Value:	
	NOM_TEMP	0x4B48
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Trect	Rectal Temperature	
	Label:	
	NLS_NOM_TEMP_RECT	0x0002E004
	Observed Value:	
	NOM_TEMP_RECT	0xE004
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Tblood	Tblood	

	Label:	
	NLS_NOM_TEMP_BLD	0x0002E014
	Observed Value:	
	NOM_TEMP_BLD	0xE014
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Tcore	Core (Body) Temperature	
	Label:	
	NLS_NOM_TEMP_CORE	0x00024B60
	Observed Value:	
	NOM_TEMP_CORE	0x4B60
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Tskin	Skin Temperature	
	Label:	
	NLS_NOM_TEMP_SKIN	0x00024B74
	Observed Value:	
	NOM_TEMP_SKIN	0x4B74
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Tesoph	Esophagial Temperature	
	Label:	
	NLS_NOM_TEMP_ESOPH	0x00024B64
	Observed Value:	
	NOM_TEMP_ESOPH	0x4B64
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Tnaso	Naso pharyngial Temperature	
	Label:	
	NLS_NOM_TEMP_NASOPH	0x00024B6C
	Observed Value:	
	NOM_TEMP_NASOPH	0x4B6C
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Tart	Areterial Temperature	
	Label:	
	NLS_NOM_TEMP_ART	0x00024B50
	Observed Value:	
	NOM_TEMP_ART	0x4B50
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Tven	Venous Temperature	
	Label:	
	NLS_NOM_TEMP_VEN	0x00024B7C
	Observed Value:	
	NOM_TEMP_VEN	0x4B7C
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Tvesic	Temperature of the Urine fluid	
	Label:	
	NLS_NOM_TEMP_VESICAL	0x0002F0C4
	Observed Value:	
	NOM_TEMP_VESICAL	0xF0C4
Ttymp	Tympanic Temperature	
	Label:	
	NLS_NOM_TEMP_TYMP	0x00024B78
	Observed Value:	
	NOM_TEMP_TYMP	0x4B78

	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Tcereb	Cerebral Temperature	
	Label:	
	NLS_NOM_TEMP_CEREBRAL	0x0002F0C5
	Observed Value:	
	NOM_TEMP_CEREBRAL	0xF0C5
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Tamb	Ambient Temperature	
	Label:	
	NLS_NOM_TEMP_AMBIENT	0x0002F0C6
	Observed Value:	
	NOM_TEMP_AMBIENT	0xF0C6
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Tairwy	Airway Temperature	
	Label:	
	NLS_NOM_TEMP_AWAY	0x00024B54
	Observed Value:	
	NOM_TEMP_AWAY	0x4B54
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Tinj	Injectate Temperature	
	Label:	
	NLS_NOM_TEMP_INJ	0x00024B68
	Observed Value:	
	NOM_TEMP_INJ	0x4B68
T1Core	Core Temperature 1 (generic)	
	Label:	
	NLS_NOM_TEMP_CORE_GEN_1	0x0002F966
	Observed Value:	
	NOM_TEMP_CORE_GEN_1	0xF966
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
T2Core	Core Temperature 2 (generic)	
	Label:	
	NLS_NOM_TEMP_CORE_GEN_2	0x0002F967
	Observed Value:	
	NOM_TEMP_CORE_GEN_2	0xF967
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
DeltaTemp	Difference Temperature	
	Label:	
	NLS_NOM_TEMP_DIFF	0x0002E018
	Observed Value:	
	NOM_TEMP_DIFF	0xE018
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
Tbody	Patient Temperature	
	Label:	
	NLS_NOM_TEMP_BODY	0x00024B5C
	Observed Value:	
	NOM_TEMP	0x4B48
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
pTrect	Predictive Rectal Temperature	

	Label:	
	NLS_NOM_TEMP_RECT_PRED	0x0002F114
	Observed Value:	
	NOM_TEMP_RECT_PRED	0xF114
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
pToral	Predictive Oral Temperature	
	Label:	
	NLS_NOM_TEMP_ORAL_PRED	0x0002F110
	Observed Value:	
	NOM_TEMP_ORAL_PRED	0xF110
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
pTaxil	Predictive Axillary Temperature	
	Label:	
	NLS_NOM_TEMP_AXIL_PRED	0x0002F118
	Observed Value:	
	NOM_TEMP_AXIL_PRED	0xF118
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
T1	Generic Temperature 1 (T1)	
	Label:	
	NLS_NOM_TEMP_GEN_1	0x0002F0C7
	Observed Value:	
	NOM_TEMP_GEN_1	0xF0C7
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
T2	Generic Temperature 2 (T2)	
	Label:	
	NLS_NOM_TEMP_GEN_2	0x0002F0C8
	Observed Value:	
	NOM_TEMP_GEN_2	0xF0C8
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
T3	Generic Temperature 3 (T3)	
	Label:	
	NLS_NOM_TEMP_GEN_3	0x0002F0C9
	Observed Value:	
	NOM_TEMP_GEN_3	0xF0C9
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
T4	Generic Temperature 4 (T4)	
	Label:	
	NLS_NOM_TEMP_GEN_4	0x0002F0CA
	Observed Value:	
	NOM_TEMP_GEN_4	0xF0CA
	Units:	
	NOM_DIM_DEGC	0x17A0
	NOM_DIM_FAHR	0x1140
N2	generic N2 label	
	Label:	
	NLS_NOM_CONC_AWAY_N2	0x0002537C
	Observed Value (from VueLink):	
	NOM_CONC_AWAY_N2	0x537C
	Compound Observed Value:	
	NOM_CONC_AWAY_N2_ET	0x5380
	NOM_CONC_AWAY_N2_INSP	0x5384
	Units:	
	NOM_DIM_MMHG	0x0F20

	NOM_DIM_PERCENT	0x0220
	NOM_DIM_KILO_PASCAL	0x0F03
N2O	generic Nitrous Oxide label	
	Label:	
	NLS_NOM_CONC_AWAY_N2O	0x000251F0
	Observed Value (from VueLink):	
	NOM_CONC_AWAY_N2O	0x51F0
	Compound Observed Value:	
	NOM_CONC_AWAY_N2O_ET	0x522C
	NOM_CONC_AWAY_N2O_INSP	0x5280
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_PERCENT	0x0220
	NOM_DIM_KILO_PASCAL	0x0F03
ISO	generic Isoflurane label	
	Label:	
	NLS_NOM_CONC_AWAY_ISOFL	0x000251E8
	Observed Value (from VueLink):	
	NOM_CONC_AWAY_ISOFL	0x51E8
	Compound Observed Value:	
	NOM_CONC_AWAY_ISOFL_ET	0x5224
	NOM_CONC_AWAY_ISOFL_INSP	0x5278
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_PERCENT	0x0220
	NOM_DIM_KILO_PASCAL	0x0F03
SEV	generic Sevoflurane label	
	Label:	
	NLS_NOM_CONC_AWAY_SEVOFL	0x000251E4
	Observed Value (from VueLink):	
	NOM_CONC_AWAY_SEVOFL	0x51E4
	Compound Observed Value:	
	NOM_CONC_AWAY_SEVOFL_ET	0x5220
	NOM_CONC_AWAY_SEVOFL_INSP	0x5274
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_PERCENT	0x0220
	NOM_DIM_KILO_PASCAL	0x0F03
ENF	generic Enflurane label	
	Label:	
	NLS_NOM_CONC_AWAY_ENFL	0x000251DC
	Observed Value (from VueLink):	
	NOM_CONC_AWAY_ENFL	0x51DC
	Compound Observed Value:	
	NOM_CONC_AWAY_ENFL_ET	0x5218
	NOM_CONC_AWAY_ENFL_INSP	0x526C
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_PERCENT	0x0220
	NOM_DIM_KILO_PASCAL	0x0F03
HAL	generic Halothane label	
	Label:	
	NLS_NOM_CONC_AWAY_HALOTH	0x000251E0
	Observed Value (from VueLink):	
	NOM_CONC_AWAY_HALOTH	0x51E0
	Compound Observed Value:	
	NOM_CONC_AWAY_HALOTH_ET	0x521C
	NOM_CONC_AWAY_HALOTH_INSP	0x5270
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_PERCENT	0x0220
	NOM_DIM_KILO_PASCAL	0x0F03
DES	generic Desflurane label	
	Label:	
	NLS_NOM_CONC_AWAY_DESFL	0x000251D8

	Observed Value (from VueLink):	
	NOM_CONC_AWAY_DESFL	0x51D8
	Compound Observed Value:	
	NOM_CONC_AWAY_DESFL_ET	0x5214
	NOM_CONC_AWAY_DESFL_INSP	0x5268
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_PERCENT	0x0220
	NOM_DIM_KILO_PASCAL	0x0F03
AGT	generic Agent label	
	Label:	
	NLS_NOM_CONC_AWAY_AGENT	0x00025388
	Observed Value (from VueLink):	
	NOM_CONC_AWAY_AGENT	0x5388
	Compound Observed Value:	
	NOM_CONC_AWAY_AGENT_ET	0x538C
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_PERCENT	0x0220
	NOM_DIM_KILO_PASCAL	0x0F03
inAGT	Generic Inspired Agent Concentration	
	Label:	
	NLS_NOM_CONC_AWAY_AGENT_INSP	0x00025390
	Observed Value:	
	NOM_CONC_AWAY_AGENT_INSP	0x5390
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
	NOM_DIM_PERCENT	0x0220
AGT1	generic Agent1 label	
	Label:	
	NLS_GASES_NAMES_CONC_AWAY_AGENT1	0x805A5401
	Compound Observed Value:	
	NOM_CONC_AWAY_AGENT_ET	0x538C
	NOM_CONC_AWAY_AGENT_INSP	0x5390
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_PERCENT	0x0220
	NOM_DIM_KILO_PASCAL	0x0F03
AGT2	generic Agent2 label	
	Label:	
	NLS_GASES_NAMES_CONC_AWAY_AGENT2	0x805A5402
	Compound Observed Value:	
	NOM_CONC_AWAY_AGENT_ET	0x538C
	NOM_CONC_AWAY_AGENT_INSP	0x5390
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_PERCENT	0x0220
	NOM_DIM_KILO_PASCAL	0x0F03
MAC	Minimum Alveolar Concentration	
	Label:	
	NLS_NOM_CONC_AWAY_MAC	0x0002F099
	Observed Value:	
	NOM_CONC_AWAY_MAC	0xF099
MAC	Airway MAC Concentration	
	Label:	
	NLS_NOM_CONC_AWAY_SUM_MAC	0x0002F05D
	Compound Observed Value:	
	NOM_CONC_AWAY_SUM_MAC_ET	0xF05E
	NOM_CONC_AWAY_SUM_MAC_INSP	0xF05F
SVR	Systemic Vascular Resistance	
	Label:	
	NLS_NOM_RES_VASC_SYS	0x00024B28
	Observed Value:	
	NOM_RES_VASC_SYS	0x4B28

	Units:	
	NOM_DIM_X_DYNE_PER_SEC_PER_CM5	0x1020
SVRI	Systemic Vascular Resistance Index	
	Label:	
	NLS_NOM_RES_VASC_SYS_INDEX	0x00024900
	Observed Value:	
	NOM_RES_VASC_SYS_INDEX	0x4900
LVSW	Left Ventricular Stroke Volume	
	Label:	
	NLS_NOM_WK_LV_STROKE	0x00024B9C
	Observed Value:	
	NOM_WK_LV_STROKE	0x4B9C
LVSWI	Left Ventricular Stroke Volume Index	
	Label:	
	NLS_NOM_WK_LV_STROKE_INDEX	0x00024904
	Observed Value:	
	NOM_WK_LV_STROKE_INDEX	0x4904
RVSW	Right Ventricular Stroke Volume	
	Label:	
	NLS_NOM_WK_RV_STROKE	0x00024BA4
	Observed Value:	
	NOM_WK_RV_STROKE	0x4BA4
RVSWI	Right Ventricular Stroke Work Index	
	Label:	
	NLS_NOM_WK_RV_STROKE_INDEX	0x00024908
	Observed Value:	
	NOM_WK_RV_STROKE_INDEX	0x4908
PVR	Pulmonary vascular Resistance	
	Label:	
	NLS_NOM_RES_VASC_PULM	0x00024B24
	Observed Value:	
	NOM_RES_VASC_PULM	0x4B24
PVRI	Pulmonary vascular Resistance PVRI	
	Label:	
	NLS_NOM_RES_VASC_PULM_INDEX	0x0002F067
	Observed Value:	
	NOM_RES_VASC_PULM_INDEX	0xF067
LCW	Left Cardiac Work	
	Label:	
	NLS_NOM_WK_CARD_LEFT	0x00024B90
	Observed Value:	
	NOM_WK_CARD_LEFT	0x4B90
LCWI	Left Cardiac Work Index	
	Label:	
	NLS_NOM_WK_CARD_LEFT_INDEX	0x0002F068
	Observed Value:	
	NOM_WK_CARD_LEFT_INDEX	0xF068
RCW	Right Cardiac Work	
	Label:	
	NLS_NOM_WK_CARD_RIGHT	0x00024B94
	Observed Value:	
	NOM_WK_CARD_RIGHT	0x4B94
RCWI	Right Cardiac Work Index	
	Label:	
	NLS_NOM_WK_CARD_RIGHT_INDEX	0x0002F069
	Observed Value:	
	NOM_WK_CARD_RIGHT_INDEX	0xF069
VO2	Oxygen Consumption VO2	
	Label:	
	NLS_NOM_SAT_O2_CONSUMP	0x00024B00
	Observed Value:	
	NOM_SAT_O2_CONSUMP	0x4B00
GCS	Glasgow Coma Score	
	Label:	
	NLS_NOM_SCORE_GLAS_COMA	0x00025880

	Observed Value:	
	NOM_SCORE_GLAS_COMA	0x5880
EyeRsp	SubScore of the GCS: Eye Response	
	Label:	
	NLS_NOM_SCORE_EYE_SUBSC_GLAS_COMA	0x00025882
	Observed Value:	
	NOM_SCORE_EYE_SUBSC_GLAS_COMA	0x5882
MotRsp	SubScore of the GCS: Motoric Response	
	Label:	
	NLS_NOM_SCORE_MOTOR_SUBSC_GLAS_COMA	0x00025883
	Observed Value:	
	NOM_SCORE_MOTOR_SUBSC_GLAS_COMA	0x5883
VblRsp	SubScore of the GCS: Verbal Response	
	Label:	
	NLS_NOM_SCORE_SUBSC_VERBAL_GLAS_COMA	0x00025884
	Observed Value:	
	NOM_SCORE_SUBSC_VERBAL_GLAS_COMA	0x5884
HC	Head Circumferince	
	Label:	
	NLS_NOM_CIRCUM_HEAD	0x00025900
	Observed Value:	
	NOM_CIRCUM_HEAD	0x5900
PRL	Pupil Reaction Left eye - light reaction of left eye's pupil	
	Label:	
	NLS_NOM_TIME_PD_PUPIL_REACT_LEFT	0x00025924
	Observed Value:	
	NOM_TIME_PD_PUPIL_REACT_LEFT	0x5924
PRR	Pupil Reaction Righteye - light reaction of right eye's pupil	
	Label:	
	NLS_NOM_TIME_PD_PUPIL_REACT_RIGHT	0x00025928
	Observed Value:	
	NOM_TIME_PD_PUPIL_REACT_RIGHT	0x5928
pHa	pH in arterial Blood	
	Label:	
	NLS_NOM_CONC_PH_ART	0x00027004
	Observed Value:	
	NOM_CONC_PH_ART	0x7004
PaCO2	Partial Pressure of arterial Carbon Dioxide	
	Label:	
	NLS_NOM_CONC_PCO2_ART	0x00027008
	Observed Value:	
	NOM_CONC_PCO2_ART	0x7008
PaO2	Partial O2 arterial	
	Label:	
	NLS_NOM_CONC_PO2_ART	0x0002700C
	Observed Value:	
	NOM_CONC_PO2_ART	0x700C
Hb	Hemoglobin in arterial Blood	
	Label:	
	NLS_NOM_CONC_HB_ART	0x00027014
	Observed Value:	
	NOM_CONC_HB_ART	0x7014
CaO2	Arterial Oxygen Content CaO2	
	Label:	
	NLS_NOM_CONC_HB_O2_ART	0x00027018
	Observed Value:	
	NOM_CONC_HB_O2_ART	0x7018
pHv	pH in venous Blood	
	Label:	
	NLS_NOM_CONC_PH_VEN	0x00027034
	Observed Value:	
	NOM_CONC_PH_VEN	0x7034
PvCO2	Partial CO2 in the venous blood	
	Label:	
	NLS_NOM_CONC_PCO2_VEN	0x00027038

	Observed Value:	
	NOM_CONC_PCO2_VEN	0x7038
PvO2	Partial O2 Venous	
	Label:	
	NLS_NOM_CONC_PO2_VEN	0x0002703C
	Observed Value:	
	NOM_CONC_PO2_VEN	0x703C
CvO2	Venous Oxygen Content	
	Label:	
	NLS_NOM_CONC_HB_O2_VEN	0x00027048
	Observed Value:	
	NOM_CONC_HB_O2_VEN	0x7048
UrNa	Natrium in Urine	
	Label:	
	NLS_NOM_CONC_NA_URINE	0x0002706C
	Observed Value:	
	NOM_CONC_NA_URINE	0x706C
SerNa	Natrium in Serum	
	Label:	
	NLS_NOM_CONC_NA_SERUM	0x000270D8
	Observed Value:	
	NOM_CONC_NA_SERUM	0x70D8
pH	pH in the Blood Plasma	
	Label:	
	NLS_NOM_CONC_PH_GEN	0x00027104
	Observed Value:	
	NOM_CONC_PH_GEN	0x7104
HCO3	Hydrocarbon concentration in Blood Plasma	
	Label:	
	NLS_NOM_CONC_HCO3_GEN	0x00027108
	Observed Value:	
	NOM_CONC_HCO3_GEN	0x7108
Na	Natrium (Sodium)	
	Label:	
	NLS_NOM_CONC_NA_GEN	0x0002710C
	Observed Value:	
	NOM_CONC_NA_GEN	0x710C
K	Kalium (Potassium)	
	Label:	
	NLS_NOM_CONC_K_GEN	0x00027110
	Observed Value:	
	NOM_CONC_K_GEN	0x7110
Glu	Glucose	
	Label:	
	NLS_NOM_CONC_GLU_GEN	0x00027114
	Observed Value:	
	NOM_CONC_GLU_GEN	0x7114
PCO2	Partial CO2	
	Label:	
	NLS_NOM_CONC_PCO2_GEN	0x00027140
	Observed Value:	
	NOM_CONC_PCO2_GEN	0x7140
PO2	Partial O2.	
	Label:	
	NLS_NOM_CONC_PO2_GEN	0x00027174
	Observed Value:	
	NOM_CONC_PO2_GEN	0x7174
Hct	Haematocrit	
	Label:	
	NLS_NOM_CONC_HCT_GEN	0x00027184
	Observed Value:	
	NOM_CONC_HCT_GEN	0x7184
BE	Base Excess of Blood	
	Label:	
	NLS_NOM_BASE_EXCESS_BLD_ART	0x0002716C

	Observed Value:	
	NOM_BASE_EXCESS_BLD_ART	0x716C
VO2I	Oxygen Consumption Index VO2I	
	Label:	
	NLS_NOM_SAT_O2_CONSUMP_INDEX	0x0002F06A
	Observed Value:	
	NOM_SAT_O2_CONSUMP_INDEX	0xF06A
PB	Barometric Pressure = Ambient Pressure	
	Label:	
	NLS_NOM_PRESS_AIR_AMBIENT	0x0002F06B
	Observed Value:	
	NOM_PRESS_AIR_AMBIENT	0xF06B
InjVol	Injectate Volume (Cardiac Output)	
	Label:	
	NLS_NOM_VOL_INJ	0x0002F079
	Observed Value:	
	NOM_VOL_INJ	0xF079
ETVI	ExtraVascular Thermo Volume Index. Cardiac Output.	
	Label:	
	NLS_NOM_VOL_THERMO_EXTRA_VASC_INDEX	0x0002F07A
	Observed Value:	
	NOM_VOL_THERMO_EXTRA_VASC_INDEX	0xF07A
CompCt	Generic Numeric Calculation Constant	
	Label:	
	NLS_NOM_NUM_CALC_CONST	0x0002F07B
	Observed Value:	
	NOM_METRIC_NOS	0xEFFF
Cl	Chloride	
	Label:	
	NLS_NOM_CONC_CHLORIDE_GEN	0x00027168
	Observed Value:	
	NOM_CONC_CHLORIDE_GEN	0x7168
BUN	Blood Urea Nitrogen	
	Label:	
	NLS_NOM_CONC_BLD_UREA_NITROGEN	0x0002F08F
	Observed Value:	
	NOM_CONC_BLD_UREA_NITROGEN	0xF08F
BEecf	Base Excess of Extra-Cellular Fluid	
	Label:	
	NLS_NOM_CONC_BASE_EXCESS_ECF	0x0002F090
	Observed Value:	
	NOM_CONC_BASE_EXCESS_ECF	0xF090
Ca-vO2	Arteriovenous Oxygen Difference Ca-vO2	
	Label:	
	NLS_NOM_CONC_DIFF_HB_O2_ATR_VEN	0x0002F092
	Observed Value:	
	NOM_CONC_DIFF_HB_O2_ATR_VEN	0xF092
CathCt	Generic Numeric Calculation Constant	
	Label:	
	NLS_NOM_NUM_CATHETER_CONST	0x0002F07C
	Observed Value:	
	NOM_NUM_CATHETER_CONST	0xF07C
BSA	Body Surface Area	
	Label:	
	NLS_NOM_AREA_BODY_SURFACE	0x0002F071
	Observed Value:	
	NOM_AREA_BODY_SURFACE	0xF071
Weight	Patient Weight	
	Label:	
	NLS_NOM_PAT_WEIGHT	0x0002F093
	Observed Value:	
	NOM_PAT_WEIGHT	0xF093
Height	Patient Height	
	Label:	
	NLS_NOM_PAT_HEIGHT	0x0002F094

	Observed Value:	
	NOM_PAT_HEIGHT	0xF094
P5	Generic Pressure 5 (P5)	
	Label:	
	NLS_NOM_PRESS_GEN_5	0x0002F3F4
	Observed Value (from VueLink):	
	NOM_PRESS_GEN_5	0xF3F4
	Compound Observed Value:	
	NOM_PRESS_GEN_5_SYS	0xF3F5
	NOM_PRESS_GEN_5_DIA	0xF3F6
	NOM_PRESS_GEN_5_MEAN	0xF3F7
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
P6	Generic Pressure 6 (P6)	
	Label:	
	NLS_NOM_PRESS_GEN_6	0x0002F3F8
	Observed Value (from VueLink):	
	NOM_PRESS_GEN_6	0xF3F8
	Compound Observed Value:	
	NOM_PRESS_GEN_6_SYS	0xF3F9
	NOM_PRESS_GEN_6_DIA	0xF3FA
	NOM_PRESS_GEN_6_MEAN	0xF3FB
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
P7	Generic Pressure 7 (P7)	
	Label:	
	NLS_NOM_PRESS_GEN_7	0x0002F3FC
	Observed Value (from VueLink):	
	NOM_PRESS_GEN_7	0xF3FC
	Compound Observed Value:	
	NOM_PRESS_GEN_7_SYS	0xF3FD
	NOM_PRESS_GEN_7	0xF3FC
	NOM_PRESS_GEN_7_MEAN	0xF3FF
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
P8	Generic Pressure 8 (P8)	
	Label:	
	NLS_NOM_PRESS_GEN_8	0x0002F400
	Observed Value (from VueLink):	
	NOM_PRESS_GEN_8	0xF400
	Compound Observed Value:	
	NOM_PRESS_GEN_8_SYS	0xF401
	NOM_PRESS_GEN_8_DIA	0xF402
	NOM_PRESS_GEN_8_MEAN	0xF403
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
BUN/cr	BUN Creatinine Ratio	
	Label:	
	NLS_NOM_RATIO_BUN_CREA	0x0002F88F
	Observed Value:	
	NOM_RATIO_BUN_CREA	0xF88F
TFC	Thoracic Fluid Content	
	Label:	
	NLS_NOM_VOL_FLUID_THORAC	0x0002F8C5
	Observed Value:	
	NOM_VOL_FLUID_THORAC	0xF8C5
TFI	Thoracic Fluid Content Index	
	Label:	
	NLS_NOM_VOL_FLUID_THORAC_INDEX	0x0002F8C6
	Observed Value:	
	NOM_VOL_FLUID_THORAC_INDEX	0xF8C6

ACI	Accelerated Cardiac Index	
	Label:	
	NLS_NOM_OUTPUT_CARD_INDEX_ACCEL	0x0002F889
	Observed Value:	
	NOM_OUTPUT_CARD_INDEX_ACCEL	0xF889
HI	Heart Contractility Index	
	Label:	
	NLS_NOM_CARD_CONTRACT_HEATHER_INDEX	0x0002F81C
	Observed Value:	
	NOM_CARD_CONTRACT_HEATHER_INDEX	0xF81C
CH2O	Free Water Clearance	
	Label:	
	NLS_NOM_FREE_WATER_CLR	0x0002F884
	Observed Value:	
	NOM_FREE_WATER_CLR	0xF884
COsm	Osmolar Clearance	
	Label:	
	NLS_NOM_CREA_OSM	0x0002F83F
	Observed Value:	
	NOM_CREA_OSM	0xF83F
CreaCl	Creatinine Clearance	
	Label:	
	NLS_NOM_CONC_CREA_CLR	0x0002F16C
	Observed Value:	
	NOM_CONC_CREA_CLR	0xF16C
FeNa	Fractional Excretion of Sodium	
	Label:	
	NLS_NOM_FRACT_EXCR_NA	0x0002F194
	Observed Value:	
	NOM_FRACT_EXCR_NA	0xF194
IMV	Intermittent Mandatory Ventilation	
	Label:	
	NLS_NOM_VENT_MODE_MAND_INTERMIT	0x0002D02A
	Observed Value:	
	NOM_VENT_MODE_MAND_INTERMIT	0xD02A
PlOsm	Plasma Osmolarity	
	Label:	
	NLS_NOM_PLASMA_OSM	0x0002F16B
	Observed Value:	
	NOM_PLASMA_OSM	0xF16B
SCrea	Serum Creatinine	
	Label:	
	NLS_NOM_CONC_CREA_SER	0x0002F827
	Observed Value:	
	NOM_CONC_CREA_SER	0xF827
U/POsm	Urine Plasma Osmolarity Ratio	
	Label:	
	NLS_NOM_RATIO_URINE_SER_OSM	0x0002F898
	Observed Value:	
	NOM_RATIO_URINE_SER_OSM	0xF898
U/SCr	Urine Serum Creatinine Ratio	
	Label:	
	NLS_NOM_RATIO_CONC_URINE_CREA_SER	0x0002F892
	Observed Value:	
	NOM_RATIO_CONC_URINE_CREA_SER	0xF892
UrCrea	Urine Creatinine	
	Label:	
	NLS_NOM_CONC_CREA_URINE	0x0002F196
	Observed Value:	
	NOM_CONC_CREA_URINE	0xF196
UrK	Urine Potassium	
	Label:	
	NLS_NOM_CONC_K_URINE	0x0002F197
	Observed Value:	
	NOM_CONC_K_URINE	0xF197

UrKEx	Urinary Potassium Excretion	
	Label:	
	NLS_NOM_CONC_K_URINE_EXCR	0x0002F198
	Observed Value:	
	NOM_CONC_K_URINE_EXCR	0xF198
UrNa/K	Urine Sodium/Potassium Ratio	
	Label:	
	NLS_NOM_RATIO_CONC_URINE_NA_K	0x0002F893
	Observed Value:	
	NOM_RATIO_CONC_URINE_NA_K	0xF893
UrNaEx	Urine Sodium Excretion	
	Label:	
	NLS_NOM_CONC_NA_EXCR	0x0002F830
	Observed Value:	
	NOM_CONC_NA_EXCR	0xF830
UrOsm	Urine Osmolarity	
	Label:	
	NLS_NOM_CONC_OSM_URINE	0x0002F199
	Observed Value:	
	NOM_CONC_OSM_URINE	0xF199
UrVol	Urine Volume	
	Label:	
	NLS_NOM_VOL_URINE_BAL_PD	0x00026824
	Observed Value:	
	NOM_VOL_URINE_BAL_PD	0x6824
NsLoss	Nitrogen Balance	
	Label:	
	NLS_NOM_NSLOSS	0x0002F16D
	Observed Value:	
	NOM_NSLOSS	0xF16D
	Units:	
	NOM_DIM_PERCENT	0x0220
Length	Length for neonatal/pediatric	
	Label:	
	NLS_NOM_BIRTH_LENGTH	0x0002F818
	Observed Value:	
	NOM_BIRTH_LENGTH	0xF818
G.Age	Gestational age for neonatal	
	Label:	
	NLS_NOM_AGE_GEST	0x0002F811
	Observed Value:	
	NOM_AGE_GEST	0xF811
BSA(B)	BSA formula: Boyd	
	Label:	
	NLS_NOM_AREA_BODY_SURFACE_ACTUAL_BOYD	0x0002F812
	Observed Value:	
	NOM_AREA_BODY_SURFACE	0xF071
BSA(D)	BSA formula: Dubois	
	Label:	
	NLS_NOM_AREA_BODY_SURFACE_ACTUAL_DUBOIS	0x0002F813
	Observed Value:	
	NOM_AREA_BODY_SURFACE	0xF071
PVcP	Pressure Ventilation Control Pressure	
	Label:	
	NLS_NOM_VENT_PRESS_AWAY_PV	0x0002F8BC
	Observed Value:	
	NOM_VENT_PRESS_AWAY_PV	0xF8BC
Rdyn	Dynamic Lung Resistance	
	Label:	
	NLS_NOM_RES_AWAY_DYN	0x0002F899
	Observed Value:	
	NOM_RES_AWAY_DYN	0xF899
NgInsP	Negative Inspiratory Pressure	
	Label:	
	NLS_NOM_PRESS_AWAY_NEG_MAX	0x000250F9

	Observed Value:	
	NOM_PRESS_AWAY_NEG_MAX	0x50F9
SpPkFl	Spontaneous Peak Flow	
	Label:	
	NLS_NOM_FLOW_AWAY_MAX_SPONT	0x0002F87D
	Observed Value:	
	NOM_FLOW_AWAY_MAX_SPONT	0xF87D
SpAWRR	Spontaneous Airway Respiration Rate	
	Label:	
	NLS_NOM_AWAY_RESP_RATE_SPONT	0x0002F815
	Observed Value:	
	NOM_AWAY_RESP_RATE_SPONT	0xF815
PlGain	Pleth Gain	
	Label:	
	NLS_NOM_PULS_OXIM_PLETH_GAIN	0x0002F88D
	Observed Value:	
	NOM_PULS_OXIM_PLETH_GAIN	0xF88D
fgAGT	Fresh gas Anesthetic Agent	
	Label:	
	NLS_NOM_FLOW_AWAY_AGENT	0x0002F876
	Observed Value:	
	NOM_CONC_AWAY_AGENT	0x5388
O2EI	Oxygen Extraction Index	
	Label:	
	NLS_NOM_EXTRACT_O2_INDEX	0x0002F875
	Observed Value:	
	NOM_EXTRACT_O2_INDEX	0xF875
REF	Right Heart Ejection Fraction	
	Label:	
	NLS_NOM_RIGHT_HEART_FRACT_EJECT	0x0002F89B
	Observed Value:	
	NOM_RIGHT_HEART_FRACT_EJECT	0xF89B
EDV	End Diastolic Volume	
	Label:	
	NLS_NOM_VOL_VENT_L_END_DIA	0x00024C00
	Observed Value:	
	NOM_VOL_GLOBAL_END_DIA	0xF044
ESV	End Systolic Volume	
	Label:	
	NLS_NOM_VOL_VENT_L_END_SYS	0x00024C04
	Observed Value:	
	NOM_VOL_VENT_L_END_SYS	0x4C04
EDVI	End Diastolic Volume Index	
	Label:	
	NLS_NOM_VOL_VENT_L_END_DIA_INDEX	0x0002F8D0
	Observed Value:	
	NOM_VOL_GLOBAL_END_DIA_INDEX	0xF045
ESVI	End Systolic Volume Index	
	Label:	
	NLS_NOM_VOL_VENT_L_END_SYS_INDEX	0x0002F8D1
	Observed Value:	
	NOM_VOL_VENT_L_END_SYS_INDEX	0xF8D1
RiseTi	Rise Time	
	Label:	
	NLS_NOM_VENT_TIME_PD_RAMP	0x0002F8BD
	Observed Value:	
	NOM_VENT_TIME_PD_RAMP	0xF8BD
HFVAmp	High Frequency Ventilation Amplitude	
	Label:	
	NLS_NOM_VENT_AMPL_HFV	0x0002F8B1
	Observed Value:	
	NOM_VENT_AMPL_HFV	0xF8B1
UrUrea	Urine Urea	
	Label:	
	NLS_NOM_CONC_UREA_URINE	0x0002F195

	Observed Value:	
	NOM_CONC_UREA_URINE	0xF195
UrpH	pH value in the Urine	
	Label:	
	NLS_NOM_CONC_PH_URINE	0x00027064
	Observed Value:	
	NOM_CONC_PH_URINE	0x7064
tCO2	total of CO2 - result of Blood gas Analysis	
	Label:	
	NLS_NOM_CONC_CO2_TOT	0x0002F825
	Observed Value:	
	NOM_CONC_CO2_TOT	0xF825
tBili	total Bilirubin	
	Label:	
	NLS_NOM_CONC_BILI_TOT	0x0002F177
	Observed Value:	
	NOM_CONC_BILI_TOT	0xF177
SerGlu	Glucose in Serum	
	Label:	
	NLS_NOM_CONC_GLU_SER	0x0002F82A
	Observed Value:	
	NOM_CONC_GLU_SER	0xF82A
UrGlu	Glucose in Urine	
	Label:	
	NLS_NOM_CONC_GLU_URINE	0x0002F19F
	Observed Value:	
	NOM_CONC_GLU_URINE	0xF19F
dBili	direct Bilirubin	
	Label:	
	NLS_NOM_CONC_BILI_DIRECT	0x0002F17A
	Observed Value:	
	NOM_CONC_BILI_DIRECT	0xF17A
SerCa	Calcium in Serum	
	Label:	
	NLS_NOM_CONC_CA_SER	0x0002F824
	Observed Value:	
	NOM_CONC_CA_SER	0xF824
tSerCa	total of Calcium in Serum	
	Label:	
	NLS_NOM_CONC_tCA_SER	0x0002F15D
	Observed Value:	
	NOM_CONC_tCA_SER	0xF15D
SerMg	Magnesium in Serum	
	Label:	
	NLS_NOM_CONC_MG_SER	0x0002F15C
	Observed Value:	
	NOM_CONC_MG_SER	0xF15C
SerPho	Phosphat in Serum	
	Label:	
	NLS_NOM_CONC_P_SER	0x0002F15E
	Observed Value:	
	NOM_CONC_P_SER	0xF15E
SerK	Kalium (Potassium) in Serum	
	Label:	
	NLS_NOM_CONC_K_SER	0x0002F82F
	Observed Value:	
	NOM_CONC_K_SER	0xF82F
SerCl	Clorid in Serum	
	Label:	
	NLS_NOM_CONC_CHLOR_SER	0x0002F15F
	Observed Value:	
	NOM_CONC_CHLOR_SER	0xF15F
SerAlb	Albumine in Serum	
	Label:	
	NLS_NOM_CONC_ALB_SER	0x0002F163

	Observed Value:	
	NOM_CONC_ALB_SER	0xF163
UrCl	Clorid in Urine	
	Label:	
	NLS_NOM_CONC_CHLOR_URINE	0x0002F19A
	Observed Value:	
	NOM_CONC_CHLOR_URINE	0xF19A
SerGlo	Globulin in Serum	
	Label:	
	NLS_NOM_CONC_GLO_SER	0x0002F829
	Observed Value:	
	NOM_CONC_GLO_SER	0xF829
SerPro	(Total) Protein in Serum	
	Label:	
	NLS_NOM_CONC_PROT_SER	0x0002F178
	Observed Value:	
	NOM_CONC_PROT_SER	0xF178
SrUrea	Serum Urea	
	Label:	
	NLS_NOM_UREA_SER	0x0002F8AD
	Observed Value:	
	NOM_UREA_SER	0xF8AD
WBC	White Blood Count (leucocyte count)	
	Label:	
	NLS_NOM_WB_CNT	0x0002F168
	Observed Value:	
	NOM_WB_CNT	0xF168
RBC	Red Blood Count (erithrocyte count)	
	Label:	
	NLS_NOM_RB_CNT	0x0002F169
	Observed Value:	
	NOM_RB_CNT	0xF169
Plts	Platelets (thrombocyte count)	
	Label:	
	NLS_NOM_PLTS_CNT	0x0002F167
	Observed Value:	
	NOM_PLTS_CNT	0xF167
MCV	Mean Corpuscular Volume	
	Label:	
	NLS_NOM_VOL_CORP_MEAN	0x0002F8C4
	Observed Value:	
	NOM_VOL_CORP_MEAN	0xF8C4
MCH	Mean Corpuscular Hemoglobin. Is the erithrocyte hemoglobin content	
	Label:	
	NLS_NOM_HB_CORP_MEAN	0x0002F885
	Observed Value:	
	NOM_HB_CORP_MEAN	0xF885
MCHC	Mean Corpuscular Hemoglobin Concentration	
	Label:	
	NLS_NOM_CONC_HB_CORP_MEAN	0x0002F82C
	Observed Value:	
	NOM_CONC_HB_CORP_MEAN	0xF82C
PTT	Partial Thromboplastin Time	
	Label:	
	NLS_NOM_TIME_PD_PTT	0x0002F8A5
	Observed Value:	
	NOM_TIME_PD_PTT	0xF8A5
PT	Prothrombin Time	
	Label:	
	NLS_NOM_TIME_PD_PT	0x0002F18B
	Observed Value:	
	NOM_TIME_PD_PT	0xF18B
TT	Thrombin Time	
	Label:	
	NLS_NOM_TIME_PD_THROMBIN	0x0002F191

	Observed Value:	
	NOM_TIME_PD_THROMBIN	0xF191
AP	Alkalische Phosphatase	
	Label:	
	NLS_NOM_CONC_AP	0x0002F185
	Observed Value:	
	NOM_CONC_AP	0xF185
alphaA	Alpha Amylase	
	Label:	
	NLS_NOM_CONC_ALPHA_AMYLASE	0x0002F186
	Observed Value:	
	NOM_CONC_ALPHA_AMYLASE	0xF186
CHE	Cholesterinesterase	
	Label:	
	NLS_NOM_CONC_CHE	0x0002F182
	Observed Value:	
	NOM_CONC_CHE	0xF182
SerCK	Creatinin Kinase	
	Label:	
	NLS_NOM_CONC_CREA_KIN_SER	0x0002F180
	Observed Value:	
	NOM_CONC_CREA_KIN_SER	0xF180
CK-MB	Creatine Cinase of type muscle-brain	
	Label:	
	NLS_NOM_CONC_CREA_KIN_MB	0x0002F181
	Observed Value:	
	NOM_CONC_CREA_KIN_MB	0xF181
CK-MM	Creatine Cinase of type muscle	
	Label:	
	NLS_NOM_CONC_CREA_KIN_MM	0x0002F17F
	Observed Value:	
	NOM_CONC_CREA_KIN_MM	0xF17F
GGT	Gamma GT = Gamma Glutamyltranspeptidase	
	Label:	
	NLS_NOM_CONC_GGT	0x0002F189
	Observed Value:	
	NOM_CONC_GGT	0xF189
GOT	Glutamic Oxaloacetic Transaminase	
	Label:	
	NLS_NOM_CONC_GOT	0x0002F188
	Observed Value:	
	NOM_CONC_GOT	0xF188
GPT	Glutamic-Pyruvic-Transaminase	
	Label:	
	NLS_NOM_CONC_GPT	0x0002F187
	Observed Value:	
	NOM_CONC_GPT	0xF187
Fe	Ferrum	
	Label:	
	NLS_NOM_CONC_FE_GEN	0x0002F160
	Observed Value:	
	NOM_CONC_FE_GEN	0xF160
Chol	Cholesterin	
	Label:	
	NLS_NOM_CONC_CHOLESTEROL	0x0002F16E
	Observed Value:	
	NOM_CONC_CHOLESTEROL	0xF16E
TGL	Triglyzeride	
	Label:	
	NLS_NOM_CONC_TGL	0x0002F16F
	Observed Value:	
	NOM_CONC_TGL	0xF16F
UrPro	(Total) Protein in Urine	
	Label:	
	NLS_NOM_CONC_PRO_URINE	0x0002F19B

	Observed Value:	
	NOM_CONC_PRO_URINE	0xF19B
UrCa	Calzium in Urine	
	Label:	
	NLS_NOM_CONC_CA_URINE	0x0002F19C
	Observed Value:	
	NOM_CONC_CA_URINE	0xF19C
CO-Hb	Carboxy Hemoglobin	
	Label:	
	NLS_NOM_CONC_HB_CO_GEN	0x00027180
	Observed Value:	
	NOM_CONC_HB_CO_GEN	0x7180
HbF	Fetal Hemoglobin	
	Label:	
	NLS_NOM_CONC_HB_FETAL	0x0002F165
	Observed Value:	
	NOM_CONC_HB_FETAL	0xF165
Met-Hb	MetHemoglobin	
	Label:	
	NLS_NOM_CONC_HB_MET_GEN	0x0002717C
	Observed Value:	
	NOM_CONC_HB_MET_GEN	0x717C
tPro	Total Protein	
	Label:	
	NLS_NOM_CONC_PROT_TOT	0x0002F179
	Observed Value:	
	NOM_CONC_PROT_TOT	0xF179
LDH	Lactate Dehydrogenase	
	Label:	
	NLS_NOM_CONC_LDH	0x0002F17B
	Observed Value:	
	NOM_CONC_LDH	0xF17B
AST	Aspartin - Aminotransferase	
	Label:	
	NLS_NOM_CONC_AST	0x0002F184
	Observed Value:	
	NOM_CONC_AST	0xF184
ALP	Alveolarproteinose Rosen-Castleman-Liebow- Syndrom	
	Label:	
	NLS_NOM_CONC_ALP	0x0002F81D
	Observed Value:	
	NOM_CONC_ALP	0xF81D
RC	Reticulocyte Count	
	Label:	
	NLS_NOM_RET_CNT	0x0002F16A
	Observed Value:	
	NOM_RET_CNT	0xF16A
CT	Coagulation Time	
	Label:	
	NLS_NOM_TIME_PD_COAGULATION	0x0002F192
	Observed Value:	
	NOM_TIME_PD_COAGULATION	0xF192
ESR	Erithrocyte sedimentation rate	
	Label:	
	NLS_NOM_ES_RATE	0x0002F17C
	Observed Value:	
	NOM_ES_RATE	0xF17C
KCT	Kaolin cephalin time	
	Label:	
	NLS_NOM_TIME_PD_KAOLIN_CEPHALINE	0x0002F8A4
	Observed Value:	
	NOM_TIME_PD_KAOLIN_CEPHALINE	0xF8A4
Rexp	Expiratory Resistance	
	Label:	
	NLS_NOM_RES_AWAY_EXP	0x00025124

	Observed Value:	
	NOM_RES_AWAY_EXP	0x5124
ExpTi	Expiratory Time	
	Label:	
	NLS_NOM_TIME_PD_EXP	0x0002F8A1
	Observed Value:	
	NOM_TIME_PD_EXP	0xF8A1
Rinsp	Inspiratory Resistance	
	Label:	
	NLS_NOM_RES_AWAY_INSP	0x00025128
	Observed Value:	
	NOM_RES_AWAY_INSP	0x5128
eeFlow	Expiratory Peak Flow	
	Label:	
	NLS_NOM_FLOW_AWAY_EXP_ET	0x0002F87A
	Observed Value:	
	NOM_FLOW_AWAY_EXP_ET	0xF87A
Pmax	Maximum Pressure during a breathing cycle	
	Label:	
	NLS_NOM_VENT_PRESS_AWAY_INSP_MAX	0x0002F8BB
	Observed Value:	
	NOM_PRESS_AWAY_INSP_MAX	0x5109
AccVol	Infusion Pump Accumulated volume. Measured value	
	Label:	
	NLS_NOM_VOL_INFUS_ACTUAL_TOTAL	0x000268FC
	Observed Value:	
	NOM_VOL_INFUS_ACTUAL_TOTAL	0x68FC
i-eN2O	Inspired - EndTidal N2O	
	Label:	
	NLS_NOM_VENT_CONC_AWAY_N2O_DELTA	0x0002F8B7
	Observed Value:	
	NOM_VENT_CONC_AWAY_N2O_DELTA	0xF8B7
i-eHAL	Inspired - EndTidal Halothane	
	Label:	
	NLS_NOM_VENT_CONC_AWAY_HALOTH_DELTA	0x0002F8B5
	Observed Value:	
	NOM_VENT_CONC_AWAY_HALOTH_DELTA	0xF8B5
i-eENF	Inspired - EndTidal Enfluran	
	Label:	
	NLS_NOM_VENT_CONC_AWAY_ENFL_DELTA	0x0002F8B4
	Observed Value:	
	NOM_VENT_CONC_AWAY_ENFL_DELTA	0xF8B4
i-eISO	Inspired - EndTidal Isofluran	
	Label:	
	NLS_NOM_VENT_CONC_AWAY_ISOFL_DELTA	0x0002F8B6
	Observed Value:	
	NOM_VENT_CONC_AWAY_ISOFL_DELTA	0xF8B6
i-eSEV	Inspired - EndTidal Sevofluran	
	Label:	
	NLS_NOM_VENT_CONC_AWAY_SEVOFL_DELTA	0x0002F8B9
	Observed Value:	
	NOM_VENT_CONC_AWAY_SEVOFL_DELTA	0xF8B9
i-eDES	Inspired - EndTidal Desfluran	
	Label:	
	NLS_NOM_VENT_CONC_AWAY_DESFL_DELTA	0x0002F8B3
	Observed Value:	
	NOM_VENT_CONC_AWAY_DESFL_DELTA	0xF8B3
i-eAGT	Inspired - EndTidal Agent	
	Label:	
	NLS_NOM_VENT_CONC_AWAY_AGENT_DELTA	0x0002F8B2
	Observed Value:	
	NOM_VENT_CONC_AWAY_AGENT_DELTA	0xF8B2
cktO2	O2 measured in the Patient Circuit	
	Label:	
	NLS_NOM_VENT_CONC_AWAY_O2_CIRCUIT	0x0002F8B8

	Observed Value:	
	NOM_VENT_CONC_AWAY_O2_CIRCUIT	0xF8B8
MMV	Mandatory Minute Volume	
	Label:	
	NLS_NOM_VENT_VOL_MINUTE_AWAY_MAND	0x000251CC
	Observed Value:	
	NOM_VENT_VOL_MINUTE_AWAY_MAND	0x51CC
RRaw	Airway Respiration Rate. Used by the Ohmeda Ventilator.	
	Label:	
	NLS_NOM_VENT_RESP_RATE	0x00025022
	Observed Value:	
	NOM_AWAY_RESP_RATE	0x5012
HFMVin	Inspired High Frequency Mandatory Minute Volume	
	Label:	
	NLS_NOM_VOL_MINUTE_AWAY_INSP_HFV	0x0002F8CD
	Observed Value:	
	NOM_VOL_MINUTE_AWAY_INSP_HFV	0xF8CD
DCO2	High Frequency Gas Transport Coefficient value	
	Label:	
	NLS_NOM_COEF_GAS_TRAN	0x000251D4
	Observed Value:	
	NOM_COEF_GAS_TRAN	0x51D4
SpTVex	Spontaenous Expired Tidal Volume	
	Label:	
	NLS_NOM_VOL_AWAY_EXP_TIDAL_SPONT	0x0002F8C2
	Observed Value:	
	NOM_VOL_AWAY_EXP_TIDAL_SPONT	0xF8C2
SpTV	Spontaneuous Tidal Volume	
	Label:	
	NLS_NOM_VENT_VOL_TIDAL_SPONT	0x0002F0F3
	Observed Value:	
	NOM_VENT_VOL_TIDAL_SPONT	0xF0F3
MTV	Mandatory Tidal Volume	
	Label:	
	NLS_NOM_VENT_VOL_TIDAL_MAND	0x0002F0F2
	Observed Value:	
	NOM_VENT_VOL_TIDAL_MAND	0xF0F2
HFTVin	Inspired High Frequency Tidal Volume	
	Label:	
	NLS_NOM_VENT_VOL_AWAY_INSP_TIDAL_HFV	0x0002F8BE
	Observed Value:	
	NOM_VENT_VOL_AWAY_INSP_TIDAL_HFV	0xF8BE
HFVTV	High Frequency Fraction Ventilation Tidal Volume	
	Label:	
	NLS_NOM_VENT_VOL_TIDAL_HFV	0x0002F8BF
	Observed Value:	
	NOM_VENT_VOL_TIDAL_HFV	0xF8BF
extHR	denotes a Heart Rate received from an external device	
	Label:	
	NLS_NOM_CARD_BEAT_RATE_EXT	0x0002F81B
	Observed Value:	
	NOM_ECG_CARD_BEAT_RATE	0x4182
Rf-I	ST Reference Value for Lead I	
	Label:	
	NLS_NOM_ECG_AMPL_ST_BASELINE_I	0x0002F411
	Observed Value:	
	NOM_ECG_AMPL_ST_BASELINE_I	0xF411
Rf-II	ST Reference Value for Lead II	
	Label:	
	NLS_NOM_ECG_AMPL_ST_BASELINE_II	0x0002F412
	Observed Value:	
	NOM_ECG_AMPL_ST_BASELINE_II	0xF412
Rf-III	ST Reference Value for Lead III	
	Label:	
	NLS_NOM_ECG_AMPL_ST_BASELINE_III	0x0002F44D

	Observed Value:	
	NOM_ECG_AMPL_ST_BASELINE_III	0xF44D
Rf-aVR	ST Reference Value for Lead aVR	
	Label:	
	NLS_NOM_ECG_AMPL_ST_BASELINE_AVR	0x0002F44E
	Observed Value:	
	NOM_ECG_AMPL_ST_BASELINE_AVR	0xF44E
Rf-aVL	ST Reference Value for Lead aVL	
	Label:	
	NLS_NOM_ECG_AMPL_ST_BASELINE_AVL	0x0002F44F
	Observed Value:	
	NOM_ECG_AMPL_ST_BASELINE_AVL	0xF44F
Rf-aVF	ST Reference Value for Lead aVF	
	Label:	
	NLS_NOM_ECG_AMPL_ST_BASELINE_AVF	0x0002F450
	Observed Value:	
	NOM_ECG_AMPL_ST_BASELINE_AVF	0xF450
Rf-V1	ST Reference Value for Lead V1	
	Label:	
	NLS_NOM_ECG_AMPL_ST_BASELINE_V1	0x0002F413
	Observed Value:	
	NOM_ECG_AMPL_ST_BASELINE_V1	0xF413
Rf-V2	ST Reference Value for Lead V2	
	Label:	
	NLS_NOM_ECG_AMPL_ST_BASELINE_V2	0x0002F414
	Observed Value:	
	NOM_ECG_AMPL_ST_BASELINE_V2	0xF414
Rf-V3	ST Reference Value for Lead V3	
	Label:	
	NLS_NOM_ECG_AMPL_ST_BASELINE_V3	0x0002F415
	Observed Value:	
	NOM_ECG_AMPL_ST_BASELINE_V3	0xF415
Rf-V4	ST Reference Value for Lead V4	
	Label:	
	NLS_NOM_ECG_AMPL_ST_BASELINE_V4	0x0002F416
	Observed Value:	
	NOM_ECG_AMPL_ST_BASELINE_V4	0xF416
Rf-V5	ST Reference Value for Lead V5	
	Label:	
	NLS_NOM_ECG_AMPL_ST_BASELINE_V5	0x0002F417
	Observed Value:	
	NOM_ECG_AMPL_ST_BASELINE_V5	0xF417
Rf-V6	ST Reference Value for Lead V6	
	Label:	
	NLS_NOM_ECG_AMPL_ST_BASELINE_V6	0x0002F418
	Observed Value:	
	NOM_ECG_AMPL_ST_BASELINE_V6	0xF418
LT %AL	Percent Alpha - Left (LT) Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_ALPHA_REL_LEFT	0x0002F859
	Observed Value:	
	NOM_EEG_PWR_SPEC_ALPHA_REL	0x59D4
LT %BE	Percent Beta - Left Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_BETA_REL_LEFT	0x0002F85F
	Observed Value:	
	NOM_EEG_PWR_SPEC_BETA_REL	0x59D8
LT %DL	Percent Delta - Left Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_DELTA_REL_LEFT	0x0002F867
	Observed Value:	
	NOM_EEG_PWR_SPEC_DELTA_REL	0x59DC
LT %TH	Percent Theta - Left Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_THETA_REL_LEFT	0x0002F86D

	Observed Value:	
	NOM_EEG_PWR_SPEC_THETA_REL	0x59E0
LT AL	Absolute Alpha - Left Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_ALPHA_ABS_LEFT	0x0002F855
	Observed Value:	
	NOM_EEG_PWR_SPEC_ALPHA_ABS_LEFT	0xF855
LT BE	Absolute Beta - Left Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_BETA_ABS_LEFT	0x0002F85B
	Observed Value:	
	NOM_EEG_PWR_SPEC_BETA_ABS_LEFT	0xF85B
LT DL	Absolute Delta - Left Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_DELTA_ABS_LEFT	0x0002F863
	Observed Value:	
	NOM_EEG_PWR_SPEC_DELTA_ABS_LEFT	0xF863
LT TH	Absolute Theta - Left Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_THETA_ABS_LEFT	0x0002F869
	Observed Value:	
	NOM_EEG_PWR_SPEC_THETA_ABS_LEFT	0xF869
LT MDF	Mean Dominant Frequency - Left Side	
	Label:	
	NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_DOM_MEAN_LEFT	0x0002F849
	Observed Value:	
	NOM_EEG_FREQ_PWR_SPEC_CRTX_DOM_MEAN	0x597C
LT MPF	Median Power Frequency - Left Side	
	Label:	
	NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_MEDIAN_LEFT	0x0002F84B
	Observed Value:	
	NOM_EEG_FREQ_PWR_SPEC_CRTX_MEDIAN_LEFT	0xF84B
LT PPF	Peak Power Frequency - Left Side	
	Label:	
	NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_PEAK_LEFT	0x0002F84F
	Observed Value:	
	NOM_EEG_FREQ_PWR_SPEC_CRTX_PEAK	0x5984
LT SEF	Spectral Edge Frequency - Left Side	
	Label:	
	NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_SPECTRAL_EDGE_LEFT	0x0002F853
	Observed Value:	
	NOM_EEG_FREQ_PWR_SPEC_CRTX_SPECTRAL_EDGE	0x5988
LT TP	Total Power - Left Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_TOT_LEFT	0x0002F871
	Observed Value:	
	NOM_EEG_PWR_SPEC_TOT	0x59B8
LSCALE	Scale of the Left Channel EEG wave	
	Label:	
	NLS_NOM_EEG_ELEC_POTL_CRTX_GAIN_LEFT	0x0002F841
	Observed Value:	
	NOM_EEG_ELEC_POTL_CRTX_GAIN_LEFT	0xF841
RT %AL	Percent Alpha - Right (RT) Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_ALPHA_REL_RIGHT	0x0002F85A
	Observed Value:	
	NOM_EEG_PWR_SPEC_ALPHA_REL	0x59D4
RT %BE	Percent Beta - Right Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_BETA_REL_RIGHT	0x0002F860
	Observed Value:	
	NOM_EEG_PWR_SPEC_BETA_REL	0x59D8
RT %DL	Percent Delta - Right Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_DELTA_REL_RIGHT	0x0002F868

	Observed Value:	
	NOM_EEG_PWR_SPEC_DELTA_REL	0x59DC
RT %TH	Percent Theta - Right Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_THETA_REL_RIGHT	0x0002F86E
	Observed Value:	
	NOM_EEG_PWR_SPEC_THETA_REL	0x59E0
RT AL	Absolute Alpha - Right Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_ALPHA_ABS_RIGHT	0x0002F856
	Observed Value:	
	NOM_EEG_PWR_SPEC_ALPHA_ABS_RIGHT	0xF856
RT BE	Absolute Beta - Right Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_BETA_ABS_RIGHT	0x0002F85C
	Observed Value:	
	NOM_EEG_PWR_SPEC_BETA_ABS_RIGHT	0xF85C
RT DL	Absolute Delta - Right Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_DELTA_ABS_RIGHT	0x0002F864
	Observed Value:	
	NOM_EEG_PWR_SPEC_DELTA_ABS_RIGHT	0xF864
RT TH	Absolute Theta - Right Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_THETA_ABS_RIGHT	0x0002F86A
	Observed Value:	
	NOM_EEG_PWR_SPEC_THETA_ABS_RIGHT	0xF86A
RT MDF	Mean Dominant Frequency - Right Side	
	Label:	
	NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_DOM_MEAN_RIGHT	0x0002F84A
	Observed Value:	
	NOM_EEG_FREQ_PWR_SPEC_CRTX_DOM_MEAN	0x597C
RT MPF	Median Power Frequency - Right Side	
	Label:	
	NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_MEDIAN_RIGHT	0x0002F84C
	Observed Value:	
	NOM_EEG_FREQ_PWR_SPEC_CRTX_MEDIAN_RIGHT	0xF84C
RT PPF	Peak Power Frequency - Right Side	
	Label:	
	NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_PEAK_RIGHT	0x0002F850
	Observed Value:	
	NOM_EEG_FREQ_PWR_SPEC_CRTX_PEAK	0x5984
RT SEF	Spectral Edge Frequency - Right Side	
	Label:	
	NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_SPECTRAL_EDGE_RIGHT	0x0002F854
	Observed Value:	
	NOM_EEG_FREQ_PWR_SPEC_CRTX_SPECTRAL_EDGE	0x5988
RT TP	Total Power - Right Side	
	Label:	
	NLS_NOM_EEG_PWR_SPEC_TOT_RIGHT	0x0002F872
	Observed Value:	
	NOM_EEG_PWR_SPEC_TOT	0x59B8
RSCALE	Scale of the Right Channel EEG wave	
	Label:	
	NLS_NOM_EEG_ELEC_POTL_CRTX_GAIN_RIGHT	0x0002F842
	Observed Value:	
	NOM_EEG_ELEC_POTL_CRTX_GAIN_RIGHT	0xF842
DPosP	Duration Above Base Pressure	
	Label:	
	NLS_NOM_VENT_TIME_PD_PPV	0x00025360
	Observed Value:	
	NOM_VENT_TIME_PD_PPV	0x5360
RRsync	Sync Breath Rate	
	Label:	
	NLS_NOM_RESP_BREATH_ASSIST_CNT	0x0002F89A

	Observed Value:	
	NOM_RESP_BREATH_ASSIST_CNT	0xF89A
fgDES	fresh gas agent for DESflurane	
	Label:	
	NLS_NOM_FLOW_AWAY_DESFL	0x0002F878
	Observed Value:	
	NOM_CONC_AWAY_DESFL	0x51D8
fgSEV	fresh gas agent for SEVoflurane	
	Label:	
	NLS_NOM_FLOW_AWAY_SEVOFL	0x0002F880
	Observed Value:	
	NOM_CONC_AWAY_SEVOFL	0x51E4
fgHAL	fresh gas agent for HALothane	
	Label:	
	NLS_NOM_FLOW_AWAY_HALOTH	0x0002F87B
	Observed Value:	
	NOM_CONC_AWAY_HALOTH	0x51E0
fgENF	fresh gas agent for ENFlurane	
	Label:	
	NLS_NOM_FLOW_AWAY_ENFL	0x0002F879
	Observed Value:	
	NOM_CONC_AWAY_ENFL	0x51DC
fgISO	fresh gas agent for ISOflurane	
	Label:	
	NLS_NOM_FLOW_AWAY_ISOFL	0x0002F87C
	Observed Value:	
	NOM_CONC_AWAY_ISOFL	0x51E8
fgN2O	N2O concentration in the fresh gas line	
	Label:	
	NLS_NOM_FLOW_AWAY_N2O	0x0002F87E
	Observed Value:	
	NOM_CONC_AWAY_N2O	0x51F0
fgO2	Oxygen concentration in the fresh gas line	
	Label:	
	NLS_NOM_FLOW_AWAY_O2	0x0002F87F
	Observed Value:	
	NOM_CONC_AWAY_O2	0x5164
fgAir	Fresh Gas Flow of Air	
	Label:	
	NLS_NOM_FLOW_AWAY_AIR	0x0002F877
	Observed Value:	
	NOM_FLOW_AWAY_AIR	0xF877
fgFlow	Total Fresh Gas Flow	
	Label:	
	NLS_NOM_FLOW_AWAY_TOT	0x0002F881
	Observed Value:	
	NOM_FLOW_AWAY_TOT	0xF881
AGTLev	Liquid level in the anesthetic agent bottle	
	Label:	
	NLS_NOM_VOL_LVL_LIQUID_BOTTLE_AGENT	0x0002F8C7
	Observed Value:	
	NOM_VOL_LVL_LIQUID_BOTTLE_AGENT	0xF8C7
ISOLev	Liquid level in the ISOflurane bottle	
	Label:	
	NLS_NOM_VOL_LVL_LIQUID_BOTTLE_ISOFL	0x0002F8CB
	Observed Value:	
	NOM_VOL_LVL_LIQUID_BOTTLE_ISOFL	0xF8CB
ENFLev	Liquid level in the ENFlurane bottle	
	Label:	
	NLS_NOM_VOL_LVL_LIQUID_BOTTLE_ENFL	0x0002F8C9
	Observed Value:	
	NOM_VOL_LVL_LIQUID_BOTTLE_ENFL	0xF8C9
HALLev	Liquid level in the HALothane bottle	
	Label:	
	NLS_NOM_VOL_LVL_LIQUID_BOTTLE_HALOTH	0x0002F8CA

	Observed Value:	
	NOM_VOL_LVL_LIQUID_BOTTLE_HALOTH	0xF8CA
DESLev	Liquid level in the DESflurane bottle	
	Label:	
	NLS_NOM_VOL_LVL_LIQUID_BOTTLE_DESFL	0x0002F8C8
	Observed Value:	
	NOM_VOL_LVL_LIQUID_BOTTLE_DESFL	0xF8C8
SEVLev	Liquid level in the SEVoflurane bottle	
	Label:	
	NLS_NOM_VOL_LVL_LIQUID_BOTTLE_SEVOFL	0x0002F8CC
	Observed Value:	
	NOM_VOL_LVL_LIQUID_BOTTLE_SEVOFL	0xF8CC
UrVSht	Urimeter - Urine Shift Volume.	
	Label:	
	NLS_NOM_VOL_URINE_SHIFT	0x0002F8CF
	Observed Value:	
	NOM_VOL_URINE_SHIFT	0xF8CF
UrFl	Urimeter - Urine Flow.	
	Label:	
	NLS_NOM_FLOW_URINE_INSTANT	0x0002680C
	Observed Value:	
	NOM_FLOW_URINE_INSTANT	0x680C
iCa	ionized Calcium	
	Label:	
	NLS_NOM_CONC_CA_GEN	0x00027118
	Observed Value:	
	NOM_CONC_CA_GEN	0x7118
'Hb	Calculated Hemoglobin	
	Label:	
	NLS_NOM_CONC_HB_ART_CALC	0x0002F82B
	Observed Value:	
	NOM_CONC_HB_ART	0x7014
pHc	pH value in the capillaries	
	Label:	
	NLS_NOM_CONC_PH_CAP	0x0002F158
	Observed Value:	
	NOM_CONC_PH_CAP	0xF158
&pH	Adjusted pH at &Patient Temperature	
	Label:	
	NLS_NOM_CONC_PH_GEN_ADJ	0x0002F838
	Observed Value:	
	NOM_CONC_PH_GEN_ADJ	0xF838
&pHa	Adjusted pH in the arterial Blood	
	Label:	
	NLS_NOM_CONC_PH_ART_ADJ	0x0002F836
	Observed Value:	
	NOM_CONC_PH_ART	0x7004
&pHv	Adjusted pH value in the venous Blood	
	Label:	
	NLS_NOM_CONC_PH_VEN_ADJ	0x0002F839
	Observed Value:	
	NOM_CONC_PH_VEN	0x7034
&pHc	Adjusted pH value in the capillaries	
	Label:	
	NLS_NOM_CONC_PH_CAP_ADJ	0x0002F837
	Observed Value:	
	NOM_CONC_PH_CAP_ADJ	0xF837
PcO2	Partial O2 in the capillaries	
	Label:	
	NLS_NOM_CONC_PO2_CAP	0x0002F15A
	Observed Value:	
	NOM_CONC_PO2_CAP	0xF15A
&PO2	Adjusted PO2 at Patient Temperature	
	Label:	
	NLS_NOM_CONC_PO2_GEN_ADJ	0x0002F83D

	Observed Value:	
	NOM_CONC_PO2_GEN	0x7174
&PaO2	Adjusted PaO2 at Patient Temperature on the arterial blood	
	Label:	
	NLS_NOM_CONC_PO2_ART_ADJ	0x0002F83B
	Observed Value:	
	NOM_CONC_PO2_ART_ADJ	0xF83B
&PvO2	Adjusted PvO2 at Patient Temperature	
	Label:	
	NLS_NOM_CONC_PO2_VEN_ADJ	0x0002F83E
	Observed Value:	
	NOM_CONC_PO2_VEN	0x703C
&PcO2	Adjusted PcO2 at Patient Temperature	
	Label:	
	NLS_NOM_CONC_PO2_CAP_ADJ	0x0002F83C
	Observed Value:	
	NOM_CONC_PO2_CAP_ADJ	0xF83C
PcCO2	Partial CO2 in the capillaries	
	Label:	
	NLS_NOM_CONC_PCO2_CAP	0x0002F159
	Observed Value:	
	NOM_CONC_PCO2_CAP	0xF159
&PCO2	Computed PCO2 at Patient Temperature	
	Label:	
	NLS_NOM_CONC_PCO2_GEN_ADJ	0x0002F834
	Observed Value:	
	NOM_CONC_PCO2_GEN	0x7140
&PaCO2	Computed PaCO2 at Patient Temperature on the arterial blood	
	Label:	
	NLS_NOM_CONC_PCO2_ART_ADJ	0x0002F832
	Observed Value:	
	NOM_CONC_PCO2_ART_ADJ	0xF832
&PvCO2	Computed PvCO2 at Patient Temperature	
	Label:	
	NLS_NOM_CONC_PCO2_VEN_ADJ	0x0002F835
	Observed Value:	
	NOM_CONC_PCO2_VEN	0x7038
&PcCO2	Computed PcO2 at Patient Temperature	
	Label:	
	NLS_NOM_CONC_PCO2_CAP_ADJ	0x0002F833
	Observed Value:	
	NOM_CONC_PCO2_CAP_ADJ	0xF833
'tCO2	Calculated total CO2	
	Label:	
	NLS_NOM_CONC_CO2_TOT_CALC	0x0002F826
	Observed Value:	
	NOM_CONC_CO2_TOT_CALC	0xF826
'SO2	Calculated SO2	
	Label:	
	NLS_NOM_SAT_O2_CALC	0x0002F89C
	Observed Value:	
	NOM_SAT_O2_ART	0x4B34
'SaO2	Calculated SaO2	
	Label:	
	NLS_NOM_SAT_O2_ART_CALC	0x0002F164
	Observed Value:	
	NOM_SAT_O2_ART_CALC	0xF164
'SvO2	Calculated SvO2	
	Label:	
	NLS_NOM_SAT_O2_VEN_CALC	0x0002F166
	Observed Value:	
	NOM_SAT_O2_VEN	0x4B3C
'ScO2	Calculated ScO2	
	Label:	
	NLS_NOM_SAT_O2_CAP_CALC	0x0002F1A0

	Observed Value:	
	NOM_SAT_O2_CAP_CALC	0xF1A0
'HCO3	Calculated HCO3	
	Label:	
	NLS_NOM_CONC_HCO3_GEN_CALC	0x0002F82E
	Observed Value:	
	NOM_CONC_HCO3_GEN	0x7108
'BEecf	Calculated Base Excess	
	Label:	
	NLS_NOM_CONC_BASE_EXCESS_ECF_CALC	0x0002F821
	Observed Value:	
	NOM_CONC_BASE_EXCESS_ECF	0xF090
'AnGap	Calculated AnionGap	
	Label:	
	NLS_NOM_CONC_AN_GAP_CALC	0x0002F1A1
	Observed Value:	
	NOM_CONC_AN_GAP_CALC	0xF1A1
Urea	Urea used by the i-Stat	
	Label:	
	NLS_NOM_CONC_UREA_GEN	0x0002F172
	Observed Value:	
	NOM_CONC_UREA_GEN	0xF172
'BE,B	Calculated Base Excess in Blood	
	Label:	
	NLS_NOM_BASE_EXCESS_BLD_ART_CALC	0x0002F817
	Observed Value:	
	NOM_BASE_EXCESS_BLD_ART	0x716C
iMg	ionized Magnesium	
	Label:	
	NLS_NOM_CONC_MG_ION	0x0002F15B
	Observed Value:	
	NOM_CONC_MG_ION	0xF15B
Crea	Creatinine - Measured Value by the i-Stat Module	
	Label:	
	NLS_NOM_CONC_CREA	0x0002F173
	Observed Value:	
	NOM_CONC_CREA	0xF173
'B/Cre	Ratio BUN/Creatinine. Calculated value by the i-Stat module	
	Label:	
	NLS_NOM_RATIO_CONC_BLD_UREA_NITROGEN_CREA_CALC	0x0002F890
	Observed Value:	
	NOM_RATIO_CONC_BLD_UREA_NITROGEN_CREA_CALC	0xF890
'U/Cre	Ratio Urea/Creatinine. Calculated value by the i-Stat module	
	Label:	
	NLS_NOM_RATIO_CONC_URINE_CREA_CALC	0x0002F891
	Observed Value:	
	NOM_RATIO_CONC_URINE_CREA_CALC	0xF891
Lact	Lactate. SMeasured value by the i-Stat module	
	Label:	
	NLS_NOM_CONC_LACT	0x0002F174
	Observed Value:	
	NOM_CONC_LACT	0xF174
Elapse	Time to Elapse Counter	
	Label:	
	NLS_NOM_TIME_PD_FROM_LAST_MSMT	0x0002F8A2
	Observed Value:	
	NOM_TIME_PD_FROM_LAST_MSMT	0xF8A2
	Units:	
	NOM_DIM_SEC	0x0880
Air T	Air Temperature in the Incubator	
	Label:	
	NLS_NOM_TEMP_AIR_INCUB	0x0002F12A
	Observed Value:	
	NOM_TEMP_AIR_INCUB	0xF12A
Hum	Humidity in the Incubator	

	Label:	
	NLS_NOM_HUMID	0x0002F103
	Observed Value:	
	NOM_HUMID	0xF103
Power	Power requ'd to set the Air&Pat Temp in the incubator	
	Label:	
	NLS_NOM_HEATING_PWR_INCUBATOR	0x0002F886
	Observed Value:	
	NOM_HEATING_PWR_INCUBATOR	0xF886
BagWgt	Weight of the Urine Disposable Bag	
	Label:	
	NLS_NOM_WEIGHT_URINE_COL	0x0002F8D3
	Observed Value:	
	NOM_WEIGHT_URINE_COL	0xF8D3
tUrVol	Total Urine Volume of the current measurement period	
	Label:	
	NLS_NOM_VOL_URINE_BAL_PD_INSTANT	0x0002F8CE
	Observed Value:	
	NOM_VOL_URINE_BAL_PD_INSTANT	0xF8CE
UrDens	Density of the Urine fluid	
	Label:	
	NLS_NOM_FLUID_DENS_URINE	0x0002F19D
	Observed Value:	
	NOM_FLUID_DENS_URINE	0xF19D
Age	actual patient age. measured in years	
	Label:	
	NLS_NOM_AGE	0x0002F810
	Observed Value:	
	NOM_AGE	0xF810
U/O	Daily Urine output	
	Label:	
	NLS_NOM_FLOW_URINE_PREV_24HR	0x0002F883
	Observed Value:	
	NOM_FLOW_URINE_PREV_24HR	0xF883
BagVol	Current fluid (Urine) in the Urine Bag	
	Label:	
	NLS_NOM_VOL_URINE_COL	0x00026830
	Observed Value:	
	NOM_VOL_URINE_COL	0x6830
PtVent	Parameter which informs whether the Patient is ventilated	
	Label:	
	NLS_NOM_VENT_ACTIVE	0x0002F8B0
	Observed Value:	
	NOM_VENT_ACTIVE	0xF8B0
	Units:	
PaFIO2	PaO2 to FIO2 ratio. Expressed in mmHg to % ratio	
	Label:	
	NLS_NOM_RATIO_PaO2_FIO2	0x0002F894
	Observed Value:	
	NOM_RATIO_PaO2_FIO2	0xF894
SpRR	Spontaneous Respiration Rate	
	Label:	
	NLS_NOM_RESP_RATE_SPONT	0x0002F828
	Observed Value:	
	NOM_RESP_RATE_SPONT	0xF828
MRR	Mandatory Respiratory Rate	
	Label:	
	NLS_NOM_VENT_RESP_RATE_MAND	0x0002F0F1
	Observed Value:	
	NOM_VENT_RESP_RATE_MAND	0xF0F1
inAGTs	Inspired secondary Anesthetic Agent	
	Label:	
	NLS_NOM_CONC_AWAY_AGENT_INSP_SEC	0x0002F81F
	Observed Value:	

	NOM_CONC_AWAY_AGENT_INSP	0x5390
etAGTs	EndTidal secondary Anesthetic Agent	
	Label:	
	NLS_NOM_CONC_AWAY_AGENT_ET_SEC	0x0002F81E
	Observed Value:	
	NOM_CONC_AWAY_AGENT_ET	0x538C
TOFcnt	Train Of Four (TOF) count - Number of TOF responses.	
	Label:	
	NLS_NOM_TRAIN_OF_FOUR_CNT	0x0002F8AB
	Observed Value:	
	NOM_TRAIN_OF_FOUR_CNT	0xF8AB
TOFrat	Train Of Four (TOF) ratio	
	Label:	
	NLS_NOM_RATIO_TRAIN_OF_FOUR	0x0002F897
	Observed Value:	
	NOM_RATIO_TRAIN_OF_FOUR	0xF897
Twitch	Twitch height of the 1Hz/0.1Hz stimulation response	
	Label:	
	NLS_NOM_TWITCH_AMPL	0x0002F8AC
	Observed Value:	
	NOM_TWITCH_AMPL	0xF8AC
PTC	Post Tetatic Count stimulation	
	Label:	
	NLS_NOM_PTC_CNT	0x0002F88B
	Observed Value:	
	NOM_PTC_CNT	0xF88B
RemTi	Remaining Time until next stimulation	
	Label:	
	NLS_NOM_TIME_PD_EVOK_REMAIN	0x0002F8A0
	Observed Value:	
	NOM_TIME_PD_EVOK_REMAIN	0xF8A0
TOF1	TrainOf Four (TOF) first response value TOF1	
	Label:	
	NLS_NOM_TRAIN_OF_FOUR_1	0x0002F8A7
	Observed Value:	
	NOM_TRAIN_OF_FOUR_1	0xF8A7
TOF2	TrainOf Four (TOF) first response value TOF2	
	Label:	
	NLS_NOM_TRAIN_OF_FOUR_2	0x0002F8A8
	Observed Value:	
	NOM_TRAIN_OF_FOUR_2	0xF8A8
TOF3	TrainOf Four (TOF) first response value TOF3	
	Label:	
	NLS_NOM_TRAIN_OF_FOUR_3	0x0002F8A9
	Observed Value:	
	NOM_TRAIN_OF_FOUR_3	0xF8A9
TOF4	TrainOf Four (TOF) first response value TOF4	
	Label:	
	NLS_NOM_TRAIN_OF_FOUR_4	0x0002F8AA
	Observed Value:	
	NOM_TRAIN_OF_FOUR_4	0xF8AA
sRepTi	Setting: Preset Train Of Four (Slow TOF) repetition time	
	Label:	
	NLS_NOM_SETT_TIME_PD_TRAIN_OF_FOUR	0x0402F8A6
	Observed Value:	
	NOM_SETT_TIME_PD_TRAIN_OF_FOUR	0xF8A6
ACT	Activated Clotting Time. Measured value by the i-Stat module	
	Label:	
	NLS_NOM_TIME_PD_ACT	0x0002F18A
	Observed Value:	
	NOM_TIME_PD_ACT	0xF18A
aPTTWB	aPTT Whole Blood	
	Label:	
	NLS_NOM_TIME_PD_aPTT_WB	0x0002F18D
	Observed Value:	

	NOM_TIME_PD_aPTT_WB	0xF18D
	Units:	
	NOM_DIM_SEC	0x0880
aPTTPE	aPTT Plasma Equivalent Time	
	Label:	
	NLS_NOM_TIME_PD_aPTT_PE	0x0002F18E
	Observed Value:	
	NOM_TIME_PD_aPTT_PE	0xF18E
	Units:	
	NOM_DIM_SEC	0x0880
PTTrat	Activated Partial Thromboplastin Time Ratio	
	Label:	
	NLS_NOM_RATIO_TIME_PD_PTT	0x0002F896
	Observed Value:	
	NOM_RATIO_TIME_PD_PTT	0xF896
PT WB	Prothrombin Time (Blood)	
	Label:	
	NLS_NOM_TIME_PD_PT_WB	0x0002F18F
	Observed Value:	
	NOM_TIME_PD_PT_WB	0xF18F
	Units:	
	NOM_DIM_SEC	0x0880
PT PE	Prothrombin Time (Plasma)	
	Label:	
	NLS_NOM_TIME_PD_PT_PE	0x0002F190
	Observed Value:	
	NOM_TIME_PD_PT_PE	0xF190
	Units:	
	NOM_DIM_SEC	0x0880
PTrat	Prothrombin Time Ratio	
	Label:	
	NLS_NOM_RATIO_TIME_PD_PT	0x0002F895
	Observed Value:	
	NOM_RATIO_TIME_PD_PT	0xF895
	Units:	
PT INR	Prothrombin Time - International Normalized Ratio	
	Label:	
	NLS_NOM_PT_INTL_NORM_RATIO	0x0002F18C
	Observed Value:	
	NOM_PT_INTL_NORM_RATIO	0xF18C
cTnI	Cardiac Troponin I	
	Label:	
	NLS_NOM_CARDIAC_TROPONIN_I	0x0002F0F4
	Observed Value:	
	NOM_CARDIAC_TROPONIN_I	0xF0F4
CPB	Cardio Pulmonary Bypass Flag	
	Label:	
	NLS_NOM_CARDIO_PULMONARY_BYPASS_MODE	0x0002F0F5
	Observed Value:	
	NOM_CARDIO_PULMONARY_BYPASS_MODE	0xF0F5
BNP	Cardiac Brain Natriuretic Peptide	
	Label:	
	NLS_NOM_BNP	0x0002F0F6
	Observed Value:	
	NOM_BNP	0xF0F6
InsTi	Spontaneous Inspiration Time	
	Label:	
	NLS_NOM_TIME_PD_INSP	0x0002F8A3
	Observed Value:	
	NOM_TIME_PD_INSP	0xF8A3
C20/C	Overdistension Index	
	Label:	
	NLS_NOM_C20_PER_C_INDEX	0x0002F81A
	Observed Value:	

	NOM_C20_PER_C_INDEX	0xF81A
TC	Time Constant	
	Label:	
	NLS_NOM_AWAY_TC	0x0002F816
	Observed Value:	
	NOM_AWAY_TC	0xF816
r	Correlation Coefficient	
	Label:	
	NLS_NOM_AWAY_CORR_COEF	0x0002F814
	Observed Value:	
	NOM_AWAY_CORR_COEF	0xF814
RVrat	Rate Volume Ratio	
	Label:	
	NLS_NOM_RATIO_AWAY_RATE_VOL_AWAY	0x0002F88E
	Observed Value:	
	NOM_RATIO_AWAY_RATE_VOL_AWAY	0xF88E
iCa(N)	ionized Calcium Normalized	
	Label:	
	NLS_NOM_CONC_CA_GEN_NORM	0x0002F822
	Observed Value:	
	NOM_CONC_CA_GEN_NORM	0xF822
TVPSV	Tidal Volume (TV) in Pressure Support Ventilation mode	
	Label:	
	NLS_NOM_VOL_AWAY_TIDAL_PSV	0x0002F8C3
	Observed Value:	
	NOM_VOL_AWAY_TIDAL_PSV	0xF8C3
RSBI	Rapid Shallow Breathing Index	
	Label:	
	NLS_NOM_BREATH_RAPID_SHALLOW_INDEX	0x0002F819
	Observed Value:	
	NOM_BREATH_RAPID_SHALLOW_INDEX	0xF819
sAWRR	Setting: Airway Respiratory Rate	
	Label:	
	NLS_NOM_SETT_AWAY_RESP_RATE	0x04025012
	Observed Value:	
	NOM_AWAY_RESP_RATE	0x5012
sTV	Setting: Tidal Volume	
	Label:	
	NLS_NOM_SETT_VOL_AWAY_TIDAL	0x0402513C
	Observed Value:	
	NOM_VOL_AWAY_TIDAL	0x513C
sPIF	Setting: Peak Inspiratory Flow	
	Label:	
	NLS_NOM_SETT_FLOW_AWAY_INSP_MAX	0x040250DD
	Observed Value:	
	NOM_PRESS_AWAY_INSP_MAX	0x5109
sFIO2	Setting: Inspired Oxygen Concentration	
	Label:	
	NLS_NOM_SETT_VENT_CONC_AWAY_O2_INSP	0x04027498
	Observed Value:	
	NOM_VENT_CONC_AWAY_O2_INSP	0x7498
sPltTi	Setting: Plateau Time	
	Label:	
	NLS_NOM_SETT_TIME_PD_RESP_PLAT	0x0402F0FF
	Observed Value:	
	NOM_SETT_TIME_PD_RESP_PLAT	0xF0FF
sSghR	Setting: Sigh Rate	
	Label:	
	NLS_NOM_SETT_VENT_SIGH_RATE	0x0402F93C
	Observed Value:	
	NOM_SETT_VENT_SIGH_RATE	0xF93C
sSghTV	Setting: Sigh Tidal Volume	
	Label:	
	NLS_NOM_SETT_VENT_VOL_TIDAL_SIGH	0x0402F8C0
	Observed Value:	

	NOM_SETT_VENT_VOL_TIDAL_SIGH	0xF8C0
sSghNr	Setting: Multiple Sigh Number	
	Label:	
	NLS_NOM_SETT_VENT_SIGH_MULT_RATE	0x0402F93B
	Observed Value:	
	NOM_SETT_VENT_SIGH_MULT_RATE	0xF93B
sATV	Setting: Apnea Tidal Volume	
	Label:	
	NLS_NOM_SETT_VOL_AWAY_TIDAL_APNEA	0x0402F951
	Observed Value:	
	NOM_SETT_VOL_AWAY_TIDAL_APNEA	0xF951
sARR	Setting: Apnea Respiration Rate	
	Label:	
	NLS_NOM_SETT_AWAY_RESP_RATE_APNEA	0x0402F8DE
	Observed Value:	
	NOM_SETT_AWAY_RESP_RATE_APNEA	0xF8DE
sAPkFl	Setting: Apnea Peak Flow	
	Label:	
	NLS_NOM_SETT_FLOW_AWAY_INSP_APNEA	0x0402F8ED
	Observed Value:	
	NOM_SETT_FLOW_AWAY_INSP_APNEA	0xF8ED
sAFIO2	Setting: Apnea Inspired O2 Concentration	
	Label:	
	NLS_NOM_SETT_VENT_CONC_AWAY_O2_INSP_APNEA	0x0402F917
	Observed Value:	
	NOM_SETT_VENT_CONC_AWAY_O2_INSP_APNEA	0xF917
sPSV	Setting: Pressure Support Ventilation	
	Label:	
	NLS_NOM_SETT_VENT_PRESS_AWAY_PV	0x0402F8BC
	Observed Value:	
	NOM_SETT_VENT_PRESS_AWAY_PV	0xF8BC
sEnSgh	Setting: Enable Sigh	
	Label:	
	NLS_NOM_SETT_VENT_MODE_SIGH	0x0402F923
	Observed Value:	
	NOM_SETT_VENT_MODE_SIGH	0xF923
sO2Suc	Setting: Suction Oxygen Concentration	
	Label:	
	NLS_NOM_SETT_VENT_O2_SUCTION_MODE	0x0402F928
	Observed Value:	
	NOM_SETT_VENT_O2_SUCTION_MODE	0xF928
sBasFl	Setting: Flow-by Base Flow	
	Label:	
	NLS_NOM_SETT_VENT_AWAY_FLOW_BASE	0x0402F910
	Observed Value:	
	NOM_SETT_VENT_AWAY_FLOW_BASE	0xF910
sSenFl	Setting: Flow-by Sensitivity Flow	
	Label:	
	NLS_NOM_SETT_VENT_AWAY_FLOW_SENSE	0x0402F911
	Observed Value:	
	NOM_SETT_VENT_AWAY_FLOW_SENSE	0xF911
sPVinT	Setting: Pressure Ventilation Inspiratory Time	
	Label:	
	NLS_NOM_SETT_VENT_TIME_PD_INSP_PV	0x0402F943
	Observed Value:	
	NOM_SETT_VENT_TIME_PD_INSP_PV	0xF943
sAPVcP	Setting: Apnea Pressure Ventilation Control Pressure	
	Label:	
	NLS_NOM_SETT_VENT_PRESS_AWAY_PV_APNEA	0x0402F933
	Observed Value:	
	NOM_SETT_VENT_PRESS_AWAY_PV_APNEA	0xF933
sAPVRR	Setting: Apnea Pressure Ventilation Respiration Rate	
	Label:	
	NLS_NOM_SETT_VENT_RESP_RATE_PV_APNEA	0x0402F93A
	Observed Value:	

```
            NOM_SETT_VENT_RESP_RATE_PV_APNEA                             0xF93A
sAPVTi      Setting: Apnea Pressure Ventilation Inspiratory Time
            Label:
            NLS_NOM_SETT_VENT_TIME_PD_INSP_PV_APNEA                      0x0402F944
            Observed Value:
            NOM_SETT_VENT_TIME_PD_INSP_PV_APNEA                          0xF944
sAPVO2      Setting: Apnea Pressure Ventilation Oxygen Concentration
            Label:
            NLS_NOM_SETT_VENT_CONC_AWAY_O2_INSP_PV_APNEA                 0x0402F918
            Observed Value:
            NOM_SETT_VENT_CONC_AWAY_O2_INSP_PV_APNEA                     0xF918
sAPVhP      Setting: Apnea Pressure Ventilation High Airway Pressure
            Label:
            NLS_NOM_SETT_VENT_PRESS_AWAY_MAX_PV_APNEA                    0x0402F931
            Observed Value:
            NOM_SETT_VENT_PRESS_AWAY_MAX_PV_APNEA                        0xF931
sPVI        Setting: Pressure Ventilation I component of I:E Ratio
            Label:
            NLS_NOM_SETT_RATIO_IE_INSP_PV                                0x0402F902
            Observed Value:
            NOM_SETT_RATIO_IE_INSP_PV                                    0xF902
sPVE        Setting: Pressure Ventilation E component of I:E Ratio
            Label:
            NLS_NOM_SETT_RATIO_IE_EXP_PV                                 0x0402F900
            Observed Value:
            NOM_SETT_RATIO_IE_EXP_PV                                     0xF900
sAPVI       Setting: Apnea Pressure Ventilation I component of I:E Ratio
            Label:
            NLS_NOM_SETT_RATIO_IE_INSP_PV_APNEA                          0x0402F903
            Observed Value:
            NOM_SETT_RATIO_IE_INSP_PV_APNEA                              0xF903
sAPVE       Setting: Apnea Pressure Ventilation E component of I:E Ratio
            Label:
            NLS_NOM_SETT_RATIO_IE_EXP_PV_APNEA                           0x0402F901
            Observed Value:
            NOM_SETT_RATIO_IE_EXP_PV_APNEA                               0xF901
sCycTi      Setting: Cycle Time
            Label:
            NLS_NOM_SETT_TIME_PD_MSMT                                    0x0402F909
            Observed Value:
            NOM_SETT_TIME_PD_MSMT                                        0xF909
sIPPV       Setting: Ventilation Frequency in IPPV Mode
            Label:
            NLS_NOM_SETT_VENT_RESP_RATE_MODE_PPV_INTERMIT_PAP            0x0402F939
            Observed Value:
            NOM_SETT_VENT_RESP_RATE_MODE_PPV_INTERMIT_PAP                0xF939
sIMV        Setting: Ventilation Frequency in IMV Mode
            Label:
            NLS_NOM_SETT_VENT_RESP_RATE_MODE_MAND_INTERMITT              0x0402F938
            Observed Value:
            NOM_VENT_MODE_MAND_INTERMIT                                  0xD02A
sPEEP       Setting: PEEP/CPAP
            Label:
            NLS_NOM_SETT_VENT_PRESS_AWAY_END_EXP_POS                     0x040251A8
            Observed Value:
            NOM_VENT_PRESS_AWAY_END_EXP_POS                              0x51A8
sSPEEP      Setting: Pressure Support PEEP
            Label:
            NLS_NOM_SETT_VENT_PRESS_AWAY_END_EXP_POS_INTERMIT            0x0402F92C
            Observed Value:
            NOM_SETT_VENT_PRESS_AWAY_END_EXP_POS_INTERMIT                0xF92C
sMV         Setting: Minute Volume
            Label:
            NLS_NOM_SETT_VOL_MINUTE_AWAY                                 0x04025148
            Observed Value:
```

	NOM_VOL_MINUTE_AWAY	0x5148
sO2Mon	Setting: O2 Monitoring	
	Label:	
	NLS_NOM_SETT_VENT_ANALY_CONC_GAS_O2_MODE	0x0402F90E
	Observed Value:	
	NOM_SETT_VENT_ANALY_CONC_GAS_O2_MODE	0xF90E
sO2Cal	Setting: O2 Calibration	
	Label:	
	NLS_NOM_SETT_VENT_O2_CAL_MODE	0x0402F926
	Observed Value:	
	NOM_SETT_VENT_O2_CAL_MODE	0xF926
sPmax	Setting: Maximum Pressure	
	Label:	
	NLS_NOM_SETT_VENT_PRESS_AWAY_INSP_MAX	0x0402F8BB
	Observed Value:	
	NOM_PRESS_AWAY_INSP_MAX	0x5109
sInsTi	Setting: Inspiratory Time	
	Label:	
	NLS_NOM_SETT_VENT_TIME_PD_INSP	0x0402F941
	Observed Value:	
	NOM_SETT_VENT_TIME_PD_INSP	0xF941
sExpTi	Setting: Exhaled Time	
	Label:	
	NLS_NOM_SETT_VENT_TIME_PD_EXP	0x0402F93F
	Observed Value:	
	NOM_SETT_VENT_TIME_PD_EXP	0xF93F
sIE 1:	Setting: Inspiration to Expiration Ratio.	
	Label:	
	NLS_NOM_SETT_RATIO_IE	0x04025118
	Observed Value:	
	NOM_RATIO_IE	0x5118
sALMRT	Setting: Alarm Percentage on Rise Time.	
	Label:	
	NLS_NOM_SETT_VENT_TIME_PD_RAMP_AL	0x0402F946
	Observed Value:	
	NOM_SETT_VENT_TIME_PD_RAMP_AL	0xF946
sCPAP	Setting: Continuous Positive Airway Pressure Value	
	Label:	
	NLS_NOM_SETT_PRESS_AWAY_CTS_POS	0x040250F4
	Observed Value:	
	NOM_PRESS_AWAY_CTS_POS	0x50F4
sFlow	Setting: Flow	
	Label:	
	NLS_NOM_SETT_VENT_FLOW	0x0402F91B
	Observed Value:	
	NOM_SETT_VENT_FLOW	0xF91B
sPIP	Setting: Positive Inspiratory Pressure	
	Label:	
	NLS_NOM_SETT_PRESS_AWAY_INSP_MAX	0x04025109
	Observed Value:	
	NOM_PRESS_AWAY_INSP_MAX	0x5109
sPmin	Setting: Low Inspiratory Pressure	
	Label:	
	NLS_NOM_SETT_PRESS_AWAY_MIN	0x040250F2
	Observed Value:	
	NOM_SETT_PRESS_AWAY_MIN	0x50F2
sHFVFl	Setting: High Freqyency Ventilation Flow	
	Label:	
	NLS_NOM_SETT_FLOW_AWAY_HFV	0x0402F8EB
	Observed Value:	
	NOM_SETT_FLOW_AWAY_HFV	0xF8EB
sHFVRR	Setting: High Frequency Ventilation Respiration Rate	
	Label:	
	NLS_NOM_SETT_AWAY_RESP_RATE_HFV	0x0402F8DF
	Observed Value:	

	NOM_SETT_AWAY_RESP_RATE_HFV	0xF8DF
sO2	Enumeration Type - denotes type of Instrument.	
	Label:	
	NLS_NOM_SETT_CONC_AWAY_O2	0x04025164
	Observed Value:	
	NOM_CONC_AWAY_O2	0x5164
sCMV	Setting: Controlled mechanical ventilation	
	Label:	
	NLS_NOM_SETT_VENT_MODE_MAND_CTS_ONOFF	0x0402F922
	Observed Value:	
	NOM_SETT_VENT_MODE_MAND_CTS_ONOFF	0xF922
sSIMV	Setting: Synchronized intermittent mandatory ventilation	
	Label:	
	NLS_NOM_SETT_VENT_MODE_SYNC_MAND_INTERMIT	0x0402F924
	Observed Value:	
	NOM_SETT_VENT_MODE_SYNC_MAND_INTERMIT	0xF924
sMMV	Setting: Mandatory Minute Volume	
	Label:	
	NLS_NOM_SETT_VENT_VOL_MINUTE_AWAY_MAND	0x040251CC
	Observed Value:	
	NOM_SETT_VENT_VOL_MINUTE_AWAY_MAND	0x51CC
sDRate	Setting: Infusion Pump Delivery Rate	
	Label:	
	NLS_NOM_SETT_FLOW_FLUID_PUMP	0x04026858
	Observed Value:	
	NOM_SETT_FLOW_FLUID_PUMP	0x6858
sPin	Setting: Pressure Ventilation Control Pressure	
	Label:	
	NLS_NOM_SETT_PRESS_AWAY_INSP	0x04025108
	Observed Value:	
	NOM_SETT_PRESS_AWAY_INSP	0x5108
sRRaw	Setting: Airway Respiration Rate. Used by the Ohmeda Ventilator.	
	Label:	
	NLS_NOM_SETT_VENT_RESP_RATE	0x04025022
	Observed Value:	
	NOM_AWAY_RESP_RATE	0x5012
sInsFl	Setting: Inspiratory Flow.	
	Label:	
	NLS_NOM_SETT_FLOW_AWAY_INSP	0x0402F8EC
	Observed Value:	
	NOM_SETT_FLOW_AWAY_INSP	0xF8EC
sExpFl	Setting: Expiratory Flow	
	Label:	
	NLS_NOM_SETT_FLOW_AWAY_EXP	0x0402F8EA
	Observed Value:	
	NOM_SETT_FLOW_AWAY_EXP	0xF8EA
sTrVol	Setting: Trigger Flow/Volume	
	Label:	
	NLS_NOM_SETT_VENT_VOL_LUNG_TRAPD	0x040251B8
	Observed Value:	
	NOM_SETT_VENT_VOL_LUNG_TRAPD	0x51B8
sAADel	Setting: Apnea Ventilation Delay	
	Label:	
	NLS_NOM_SETT_APNEA_ALARM_DELAY	0x0402F8D9
	Observed Value:	
	NOM_SETT_APNEA_ALARM_DELAY	0xF8D9
sHFVAm	Setting: HFV Amplitude (Peak to Peak Pressure)	
	Label:	
	NLS_NOM_SETT_HFV_AMPL	0x0402F8F3
	Observed Value:	
	NOM_SETT_HFV_AMPL	0xF8F3
sMVDel	Setting: Minute Volume Alarm Delay	
	Label:	
	NLS_NOM_SETT_VOL_MINUTE_ALARM_DELAY	0x0402F953
	Observed Value:	

	NOM_SETT_VOL_MINUTE_ALARM_DELAY	0xF953
sTrgFl	Setting: Flow Trigger - delivered by the Evita 2 Vuelink Driver	
	Label:	
	NLS_NOM_SETT_VENT_FLOW_INSP_TRIG	0x0402F91D
	Observed Value:	
	NOM_SETT_VENT_FLOW_INSP_TRIG	0xF91D
sPincR	Setting: Pressure Increase Rate	
	Label:	
	NLS_NOM_SETT_VENT_AWAY_PRESS_RATE_INCREASE	0x0402F912
	Observed Value:	
	NOM_SETT_VENT_AWAY_PRESS_RATE_INCREASE	0xF912
sVmax	Setting: Volume Warning - delivered by the Evita 2 Vuelink Driver	
	Label:	
	NLS_NOM_SETT_VENT_VOL_LIMIT_AL_HI_ONOFF	0x0402F949
	Observed Value:	
	NOM_SETT_VENT_VOL_LIMIT_AL_HI_ONOFF	0xF949
loPmax	Setting: Low Maximum Airway Pressure Alarm Setting.	
	Label:	
	NLS_NOM_SETT_PRESS_AWAY_INSP_MAX_LIMIT_LO	0x0402F8FB
	Observed Value:	
	NOM_SETT_PRESS_AWAY_INSP_MAX_LIMIT_LO	0xF8FB
sTVap	Setting: Applied Tidal Volume.	
	Label:	
	NLS_NOM_SETT_VOL_AWAY_TIDAL_APPLIED	0x0402F952
	Observed Value:	
	NOM_SETT_VOL_AWAY_TIDAL_APPLIED	0xF952
sSens	Setting: Assist Sensitivity. Used by the Bear 1000 ventilator.	
	Label:	
	NLS_NOM_SETT_SENS_LEVEL	0x0402F904
	Observed Value:	
	NOM_SETT_SENS_LEVEL	0xF904
sBkgFl	Setting: Background Flow Setting. Range is 2 - 30 l/min	
	Label:	
	NLS_NOM_SETT_VENT_AWAY_FLOW_BACKGROUND	0x0402F90F
	Observed Value:	
	NOM_SETT_VENT_AWAY_FLOW_BACKGROUND	0xF90F
sAGT	Setting: Vaporizer concentration.	
	Label:	
	NLS_NOM_SETT_FLOW_AWAY_AGENT	0x0402F876
	Observed Value:	
	NOM_CONC_AWAY_AGENT	0x5388
sISO	Setting: Vaporizer concentration for ISOflurane	
	Label:	
	NLS_NOM_SETT_CONC_AWAY_ISOFL	0x040251E8
	Observed Value:	
	NOM_CONC_AWAY_ISOFL	0x51E8
sENF	Setting: Vaporizer concentration for ENFlurane	
	Label:	
	NLS_NOM_SETT_CONC_AWAY_ENFL	0x040251DC
	Observed Value:	
	NOM_CONC_AWAY_ENFL	0x51DC
sHAL	Setting: Vaporizer concentration for HALothane	
	Label:	
	NLS_NOM_SETT_CONC_AWAY_HALOTH	0x040251E0
	Observed Value:	
	NOM_CONC_AWAY_HALOTH	0x51E0
sDES	Setting: Vaporizer concentration for DESflurane	
	Label:	
	NLS_NOM_SETT_CONC_AWAY_DESFL	0x040251D8
	Observed Value:	
	NOM_CONC_AWAY_DESFL	0x51D8
sSEV	Setting: Vaporizer concentration for SEVoflurane	
	Label:	
	NLS_NOM_SETT_CONC_AWAY_SEVOFL	0x040251E4
	Observed Value:	

```
            NOM_CONC_AWAY_SEVOFL                                        0x51E4
sfgAir      Setting: Total fresh gas Air flow on the mixer
            Label:
            NLS_NOM_SETT_FLOW_AWAY_AIR                                  0x0402F877
            Observed Value:
            NOM_SETT_FLOW_AWAY_AIR                                      0xF877
sfgO2       Setting: Fresh gas oxygen Flow on the mixer
            Label:
            NLS_NOM_SETT_FLOW_AWAY_O2                                   0x0402F87F
            Observed Value:
            NOM_CONC_AWAY_O2                                            0x5164
sfgFl       Setting: Total fresh gas Flow on the mixer
            Label:
            NLS_NOM_SETT_FLOW_AWAY_TOT                                  0x0402F881
            Observed Value:
            NOM_SETT_FLOW_AWAY_TOT                                      0xF881
sfgN2O      Setting: fresh gas N2O flow on the mixer
            Label:
            NLS_NOM_SETT_FLOW_AWAY_N2O                                  0x0402F87E
            Observed Value:
            NOM_CONC_AWAY_N2O                                           0x51F0
sGasPr      Setting: Gas Sample point for the oxygen measurement
            Label:
            NLS_NOM_SETT_VENT_GAS_PROBE_POSN                            0x0402F920
            Observed Value:
            NOM_SETT_VENT_GAS_PROBE_POSN                                0xF920
sO2Pr       Setting: Gas sample point for oxygen measurement
            Label:
            NLS_NOM_SETT_VENT_O2_PROBE_POSN                             0x0402F927
            Observed Value:
            NOM_SETT_VENT_O2_PROBE_POSN                                 0xF927
sTVin       Setting: inspired Tidal Volume
            Label:
            NLS_NOM_SETT_VOL_AWAY_INSP_TIDAL                            0x0402F0E0
            Observed Value:
            NOM_SETT_VOL_AWAY_INSP_TIDAL                                0xF0E0
sTemp       Desired Environmental Temperature
            Label:
            NLS_NOM_SETT_TEMP                                           0x04024B48
            Observed Value:
            NOM_SETT_TEMP                                               0x4B48
sUrTi       Setting: Preset period of time for the UrVol numeric
            Label:
            NLS_NOM_SETT_URINE_BAL_PD                                   0x0402F8AF
            Observed Value:
            NOM_SETT_URINE_BAL_PD                                       0xF8AF
sTlow       Setting: part of the Evita 4 Airway Pressure Release Ventilation Mode
            Label:
            NLS_NOM_SETT_VENT_TIME_PD_EXP_APRV                          0x0402F940
            Observed Value:
            NOM_SETT_VENT_TIME_PD_EXP_APRV                              0xF940
sThigh      Setting: part of the Evita 4 Airway Pressure Release Ventilation Mode
            Label:
            NLS_NOM_SETT_VENT_TIME_PD_INSP_APRV                         0x0402F942
            Observed Value:
            NOM_SETT_VENT_TIME_PD_INSP_APRV                             0xF942
sPlow       Setting: part of the Evita 4 Airway Pressure Release Ventilation Mode
            Label:
            NLS_NOM_SETT_VENT_PRESS_AWAY_EXP_APRV                       0x0402F92D
            Observed Value:
            NOM_SETT_VENT_PRESS_AWAY_EXP_APRV                           0xF92D
sPhigh      Setting: part of the Evita 4 Airway Pressure Release Ventilation Mode
            Label:
            NLS_NOM_SETT_VENT_PRESS_AWAY_INSP_APRV                      0x0402F92E
            Observed Value:
```

	NOM_SETT_VENT_PRESS_AWAY_INSP_APRV	0xF92E
sVolas	Setting: Volume Assist level for the CPAP mode	
	Label:	
	NLS_NOM_SETT_VENT_VOL_AWAY_ASSIST	0x0402F948
	Observed Value:	
	NOM_SETT_VENT_VOL_AWAY_ASSIST	0xF948
sFlas	Setting: Flow Assist level for the CPAP mode	
	Label:	
	NLS_NOM_SETT_VENT_FLOW_AWAY_ASSIST	0x0402F91C
	Observed Value:	
	NOM_SETT_VENT_FLOW_AWAY_ASSIST	0xF91C
sCurnt	Setting: Preset stimulation current	
	Label:	
	NLS_NOM_SETT_EVOK_CURR	0x0402F8E7
	Observed Value:	
	NOM_SETT_EVOK_CURR	0xF8E7
sChrge	Setting: Preset stimulation charge	
	Label:	
	NLS_NOM_SETT_EVOK_CHARGE	0x0402F8E6
	Observed Value:	
	NOM_SETT_EVOK_CHARGE	0xF8E6
sPulsD	Setting: Preset stimulation impulse duration	
	Label:	
	NLS_NOM_SETT_TIME_PD_EVOK	0x0402F908
	Observed Value:	
	NOM_SETT_TIME_PD_EVOK	0xF908
sfmax	Setting: Panting Limit	
	Label:	
	NLS_NOM_SETT_VENT_RESP_RATE_LIMIT_HI_PANT	0x0402F937
	Observed Value:	
	NOM_SETT_VENT_RESP_RATE_LIMIT_HI_PANT	0xF937
highP	Alarm Limit: High Pressure	
	Label:	
	NLS_NOM_SETT_VENT_PRESS_AWAY_LIMIT_HI	0x0402F930
	Observed Value:	
	NOM_SETT_VENT_PRESS_AWAY_LIMIT_HI	0xF930
loPEEP	Alarm Limit: Low PEEP/CPAP	
	Label:	
	NLS_NOM_VENT_PRESS_AWAY_END_EXP_POS_LIMIT_LO	0x0002F8BA
	Observed Value:	
	NOM_VENT_PRESS_AWAY_END_EXP_POS_LIMIT_LO	0xF8BA
sustP	Alarm Limit: Sustained Pressure Alarm Limit.	
	Label:	
	NLS_NOM_SETT_VENT_PRESS_AWAY_SUST_LIMIT_HI	0x0402F935
	Observed Value:	
	NOM_SETT_VENT_PRESS_AWAY_SUST_LIMIT_HI	0xF935
lowMV	Alarm Limit: Low Minute Volume Alarm Limit	
	Label:	
	NLS_NOM_SETT_VENT_VOL_MINUTE_AWAY_LIMIT_LO	0x0402F94C
	Observed Value:	
	NOM_SETT_VENT_VOL_MINUTE_AWAY_LIMIT_LO	0xF94C
lowO2	Alarm Limit: Low Oxygen (O2) Alarm Limit	
	Label:	
	NLS_NOM_SETT_VENT_CONC_AWAY_O2_LIMIT_LO	0x0402F91A
	Observed Value:	
	NOM_SETT_VENT_CONC_AWAY_O2_LIMIT_LO	0xF91A
highO2	Alarm Limit. High Oxygen (O2) Alarm Limit	
	Label:	
	NLS_NOM_SETT_VENT_CONC_AWAY_O2_LIMIT_HI	0x0402F919
	Observed Value:	
	NOM_SETT_VENT_CONC_AWAY_O2_LIMIT_HI	0xF919
highMV	Alarm Limit: High Minute Volume Alarm Limit	
	Label:	
	NLS_NOM_SETT_VENT_VOL_MINUTE_AWAY_LIMIT_HI	0x0402F94B
	Observed Value:	

	NOM_SETT_VENT_VOL_MINUTE_AWAY_LIMIT_HI	0xF94B
lowTV	Alarm Limit: Low Tidal Volume Alarm Limit	
	Label:	
	NLS_NOM_SETT_VENT_VOL_TIDAL_LIMIT_LO	0x0402F94E
	Observed Value:	
	NOM_SETT_VENT_VOL_TIDAL_LIMIT_LO	0xF94E
highTV	Alarm Limit: High Tidal Volume Alarm Limit	
	Label:	
	NLS_NOM_SETT_VENT_VOL_TIDAL_LIMIT_HI	0x0402F94D
	Observed Value:	
	NOM_SETT_VENT_VOL_TIDAL_LIMIT_HI	0xF94D
Num 1	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM1	0x80AAF064
	depends on configuration	
Num 2	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM2	0x80AAF066
	depends on configuration	
Num 3	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM3	0x80AAF068
	depends on configuration	
Num 4	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM4	0x80AAF06A
	depends on configuration	
Num 5	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM5	0x80AAF06C
	depends on configuration	
Num 6	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM6	0x80AAF06E
	depends on configuration	
Num 7	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM7	0x80AAF070
	depends on configuration	
Num 8	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM8	0x80AAF072
	depends on configuration	
Num 9	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM9	0x80AAF074
	depends on configuration	
Num 10	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM10	0x80AAF076
	depends on configuration	
Num 11	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM11	0x80AAF078
	depends on configuration	
Num 12	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM12	0x80AAF07A
	depends on configuration	
Num 13	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM13	0x80AAF07C
	depends on configuration	
Num 14	Placeholder for Vuelink Flex Text	
	Label:	

	NLS_VUELINK_FLX1_NPS_TEXT_NUM14	0x80AAF07E
	depends on configuration	
Num 15	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM15	0x80AAF080
	depends on configuration	
Num 16	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM16	0x80AAF082
	depends on configuration	
Num 17	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM17	0x80AAF084
	depends on configuration	
Num 18	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM18	0x80AAF086
	depends on configuration	
Num 19	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM19	0x80AAF088
	depends on configuration	
Num 20	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM20	0x80AAF08A
	depends on configuration	
Num 21	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM21	0x80AAF08C
	depends on configuration	
Num 22	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM22	0x80AAF08E
	depends on configuration	
Num 23	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM23	0x80AAF090
	depends on configuration	
Num 24	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_NUM24	0x80AAF092
	depends on configuration	
PCT	Procalcitonin	
	Label:	
	NLS_NOM_CONC_PCT	0x0002F17D
	Observed Value:	
	NOM_CONC_PCT	0xF17D
	Units:	
	NOM_DIM_PICO_G_PER_ML	0x0875
	NOM_DIM_NANO_G_PER_L	0x0814
Quick	Thromboplastine Time	
	Label:	
	NLS_NOM_TIME_PD_THROMBOPLAS	0x0002F193
	Observed Value:	
	NOM_TIME_PD_THROMBOPLAS	0xF193
	Units:	
	NOM_DIM_SEC	0x0880
HDL	High Density Lipoprotein	
	Label:	
	NLS_NOM_CONC_HDL	0x0002F170
	Observed Value:	
	NOM_CONC_HDL	0xF170
	Units:	
	NOM_DIM_MILLI_MOLE_PER_L	0x1272
	NOM_DIM_MILLI_G_PER_DL	0x0852

LDL	Low Density Lipoprotein	
	Label:	
	NLS_NOM_CONC_LDL	0x0002F171
	Observed Value:	
	NOM_CONC_LDL	0xF171
	Units:	
	NOM_DIM_MILLI_MOL_PER_L	
	NOM_DIM_MILLI_G_PER_DL	0x0852
CRP	C-reactive Protein	
	Label:	
	NLS_NOM_CONC_CRP	0x0002F183
	Observed Value:	
	NOM_CONC_CRP	0xF183
	Units:	
	NOM_DIM_MILLI_G_PER_L	0x0812
	NOM_DIM_MILLI_G_PER_DL	0x0852
UrHb	Hemoglobin (Urine)	
	Label:	
	NLS_NOM_CONC_HB_URINE	0x0002F19E
	Observed Value:	
	NOM_CONC_HB_URINE	0xF19E
	Units:	
	NOM_DIM_X_G_PER_DL	0x0840
	NOM_DIM_X_G_PER_L	0x0800
	NOM_DIM_MILLI_MOLE_PER_L	0x1272
ApneaD	Apnea Time	
	Label:	
	NLS_NOM_TIME_PD_APNEA	0x00025130
	Observed Value:	
	NOM_TIME_PD_APNEA	0x5130
FICO2	Airway CO2 inspiration	
	Label:	
	NLS_NOM_VENT_CONC_AWAY_CO2_INSP	0x00025160
	Observed Value:	
	NOM_VENT_CONC_AWAY_CO2_INSP	0x5160
	Units:	
	NOM_DIM_PERCENT	
HLMfl		
	Label:	
	NLS_NOM_FLOW_PUMP_HEART_LUNG_MAIN	0x0002F974
	Observed Value:	
	NOM_FLOW_PUMP_HEART_LUNG_MAIN	0xF974
	Units:	
SlvPfl		
	Label:	
	NLS_NOM_FLOW_PUMP_HEART_LUNG_SLAVE	0x0002F975
	Observed Value:	
	NOM_FLOW_PUMP_HEART_LUNG_SLAVE	0xF975
	Units:	
SucPfl		
	Label:	
	NLS_NOM_FLOW_PUMP_HEART_LUNG_SUCTION	0x0002F976
	Observed Value:	
	NOM_FLOW_PUMP_HEART_LUNG_SUCTION	0xF976
	Units:	
AuxPfl		
	Label:	
	NLS_NOM_FLOW_PUMP_HEART_LUNG_AUX	0x0002F977
	Observed Value:	
	NOM_FLOW_PUMP_HEART_LUNG_AUX	0xF977
	Units:	

PlePfl		
	Label:	
	NLS_NOM_FLOW_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN	0x0002F978
	Observed Value:	
	NOM_FLOW_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN	0xF978
	Units:	
SplPfl		
	Label:	
	NLS_NOM_FLOW_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE	0x0002F979
	Observed Value:	
	NOM_FLOW_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE	0xF979
	Units:	
DeltaP	Blood Pressure difference	
	Label:	
	NLS_NOM_PRESS_BLD_DIFF	0x0002F968
	Observed Value:	
	NOM_PRESS_BLD_DIFF	0xF968
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
DeltaP1	Blood Pressure difference 1 (generic)	
	Label:	
	NLS_NOM_PRESS_BLD_DIFF_GEN_1	0x0002F96C
	Observed Value:	
	NOM_PRESS_BLD_DIFF_GEN_1	0xF96C
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
DeltaP2	Blood Pressure difference 2 (generic)	
	Label:	
	NLS_NOM_PRESS_BLD_DIFF_GEN_2	0x0002F970
	Observed Value:	
	NOM_PRESS_BLD_DIFF_GEN_2	0xF970
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
AxOnTi		
	Label:	
	NLS_NOM_TIME_PD_PUMP_HEART_LUNG_AUX_SINCE_START	0x0002F97A
	Observed Value:	
	NOM_TIME_PD_PUMP_HEART_LUNG_AUX_SINCE_START	0xF97A
	Units:	
AxOffT		
	Label:	
	NLS_NOM_TIME_PD_PUMP_HEART_LUNG_AUX_SINCE_STOP	0x0002F97B
	Observed Value:	
	NOM_TIME_PD_PUMP_HEART_LUNG_AUX_SINCE_STOP	0xF97B
	Units:	
AxDVol		
	Label:	
	NLS_NOM_VOL_DELIV_PUMP_HEART_LUNG_AUX	0x0002F97C
	Observed Value:	
	NOM_VOL_DELIV_PUMP_HEART_LUNG_AUX	0xF97C
	Units:	
AxTVol		
	Label:	
	NLS_NOM_VOL_DELIV_TOTAL_PUMP_HEART_LUNG_AUX	0x0002F97D
	Observed Value:	
	NOM_VOL_DELIV_TOTAL_PUMP_HEART_LUNG_AUX	0xF97D
	Units:	

```
AxPlTi
            Label:
            NLS_NOM_TIME_PD_PLEGIA_PUMP_HEART_LUNG_AUX                   0x0002F97E
            Observed Value:
            NOM_TIME_PD_PLEGIA_PUMP_HEART_LUNG_AUX                       0xF97E
            Units:

CpOnTi
            Label:
            NLS_NOM_TIME_PD_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN_SINCE_START 0x0002F97F
            Observed Value:
            NOM_TIME_PD_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN_SINCE_START    0xF97F
            Units:

CpOffT
            Label:
            NLS_NOM_TIME_PD_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN_SINCE_STOP 0x0002F980
            Observed Value:
            NOM_TIME_PD_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN_SINCE_STOP     0xF980
            Units:

CpDVol
            Label:
            NLS_NOM_VOL_DELIV_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN          0x0002F981
            Observed Value:
            NOM_VOL_DELIV_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN              0xF981
            Units:

CpTVol
            Label:
            NLS_NOM_VOL_DELIV_TOTAL_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN    0x0002F982
            Observed Value:
            NOM_VOL_DELIV_TOTAL_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN        0xF982
            Units:

CpPlTi
            Label:
            NLS_NOM_TIME_PD_PLEGIA_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN     0x0002F983
            Observed Value:
            NOM_TIME_PD_PLEGIA_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN         0xF983
            Units:

CsOnTi
            Label:
            NLS_NOM_TIME_PD_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE_SINCE_START 0x0002F984
            Observed Value:
            NOM_TIME_PD_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE_SINCE_START   0xF984
            Units:

CsOffT
            Label:
            NLS_NOM_TIME_PD_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE_SINCE_STOP 0x0002F985
            Observed Value:
            NOM_TIME_PD_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE_SINCE_STOP    0xF985
            Units:

CsDVol
            Label:
            NLS_NOM_VOL_DELIV_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE         0x0002F986
            Observed Value:
            NOM_VOL_DELIV_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE             0xF986
            Units:

CsTVol
```

	Label:	
	NLS_NOM_VOL_DELIV_TOTAL_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE	0x0002F987
	Observed Value:	
	NOM_VOL_DELIV_TOTAL_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE	0xF987
	Units:	
CsPlTi		
	Label:	
	NLS_NOM_TIME_PD_PLEGIA_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE	0x0002F988
	Observed Value:	
	NOM_TIME_PD_PLEGIA_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE	0xF988
	Units:	
StO2	O2 Saturation (tissue)	
	Label:	
	NLS_NOM_SAT_O2_TISSUE	0x0002F960
	Observed Value:	
	NOM_SAT_O2_TISSUE	0xF960
	Units:	
	NOM_DIM_PERCENT	0x0220
CSI		
	Label:	
	NLS_NOM_CEREB_STATE_INDEX	0x0002F961
	Observed Value:	
	NOM_CEREB_STATE_INDEX	0xF961
	Units:	
Tin/Tt		
	Label:	
	NLS_NOM_RATIO_INSP_TOTAL_BREATH_SPONT	0x0002F990
	Observed Value:	
	NOM_RATIO_INSP_TOTAL_BREATH_SPONT	0xF990
	Units:	
PEinsp	Respiration Pressure Plateau	
	Label:	
	NLS_NOM_VENT_PRESS_RESP_PLAT	0x00025368
	Observed Value:	
	NOM_VENT_PRESS_RESP_PLAT	0x5368
	Units:	
tPEEP		
	Label:	
	NLS_NOM_VENT_PRESS_AWAY_END_EXP_POS_TOTAL	0x0002F991
	Observed Value:	
	NOM_VENT_PRESS_AWAY_END_EXP_POS_TOTAL	0xF991
	Units:	
	NOM_DIM_MILLI_BAR	0x0F72
Cpav		
	Label:	
	NLS_NOM_COMPL_LUNG_PAV	0x0002F992
	Observed Value:	
	NOM_COMPL_LUNG_PAV	0xF992
	Units:	
Epav		
	Label:	
	NLS_NOM_ELAS_LUNG_PAV	0x0002F995
	Observed Value:	
	NOM_ELAS_LUNG_PAV	0xF995
	Units:	
Rpav		
	Label:	
	NLS_NOM_RES_AWAY_PAV	0x0002F993

```
                Observed Value:
                NOM_RES_AWAY_PAV                                           0xF993
                Units:

Rtot
                Label:
                NLS_NOM_RES_AWAY_EXP_TOTAL                                 0x0002F994
                Observed Value:
                NOM_RES_AWAY_EXP_TOTAL                                     0xF994
                Units:

RSBInm
                Label:
                NLS_NOM_BREATH_RAPID_SHALLOW_INDEX_NORM                    0x0002F996
                Observed Value:
                NOM_BREATH_RAPID_SHALLOW_INDEX_NORM                        0xF996
                Units:
```

Enumerations

```
EctSta          ECG Ectopic Status label
                Label:
                NLS_NOM_ECG_STAT_ECT                                  0x0002D006
                Values:
```

Nomen	Description	Value
NOM_ECG_V_P_C_RUN	Run PVCs	0x4290
NOM_ECG_PACING_NON_CAPT	Pacer not capture	0x40C0
NOM_ECG_PACER_NOT_PACING	pacer not paced	0x41E0
NOM_ECG_BEAT_MISSED	missed beat	0x4058
NOM_ECG_SV_P_C_FREQUENT	frequent SVPB's	0x42F0
NOM_ECG_SV_P_C	SVPB	0x4220
NOM_ECG_SV_BEAT	SV beats	0x4208
NOM_ECG_PACED_BEAT	paced beats	0x40A8
NOM_ECG_V_P_C_PAIR	pair PVC's	0x4280
NOM_ECG_V_P_C_MULTIFOCAL	multiform PVC's	0x4278
NOM_ECG_V_P_C_RonT	R on T PVC's	0x42A0
NOM_ECG_ECT_ABSENT	no ectopic status	0x4308

```
RytSta          ECG Rhythm Status label
                Label:
                NLS_NOM_ECG_STAT_RHY                                  0x0002D007
                Values:
```

Nomen	Description	Value
NOM_ECG_ASY_RHY	Asystole	0x4003
NOM_ECG_V_FIB_TACHY_RHY	Vent Fib/Tach	0x4020
NOM_ECG_LEARN_RHY	Learning Rhythm	0x4002
NOM_ECG_LEARN	Learning ECG	0x4528
NOM_ECG_V_TACHY_RHY	Vtach	0x401A

Nomen	Description	Value
NOM_ECG_V_TACHY_RHY_SUST	Sustained VT	0x401B
NOM_ECG_V_RHY	Vent Rhythm	0x4018
NOM_ECG_V_BIGEM_RHY	Vent Bigeminy	0x4017
NOM_ECG_V_TRIGEM_RHY	Vent Trigeminy	0x401C
NOM_ECG_PACED_RHY	Paced Rhythm	0x4009
NOM_ECG_RHY_IRREG	Irregular HR	0x400D
NOM_ECG_SINUS_BRADY_RHY	Sinus Brady	0x4013
NOM_ECG_SINUS_RHY	Sinus Rhythm	0x4012
NOM_ECG_SINUS_TACHY_RHY	Sinus Tach	0x4014
NOM_ECG_SV_BRADY_RHY	SV Brady	0x4210
NOM_ECG_SV_RHY	SV Rhythm	0x4015
NOM_ECG_SV_TACHY_RHY	SV Tach	0x4016
NOM_ECG_RHY_UNK	Unknown ECG Rhythm	0x4010
NOM_ECG_RHY_UNANALYZEABLE	Cannot Analyze ECG	0x4011
NOM_ECG_RHY_ABSENT		0x400B
NOM_ECG_RHY_NOS		0x403F
NOM_ECG_RHY	ECG Rhythm	0x400A

Waves

ECG	Unspecific ECG wave	
	Label:	
	NLS_NOM_ECG_ELEC_POTL	0x00020100
	Observed Value:	
	NOM_ECG_ELEC_POTL	0x0100
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
I	ECG Lead I	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_I	0x00020101
	Observed Value:	
	NOM_ECG_ELEC_POTL_I	0x0101
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
II	ECG Lead II	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_II	0x00020102
	Observed Value:	
	NOM_ECG_ELEC_POTL_II	0x0102
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
III	ECG Lead III	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_III	0x0002013D
	Observed Value:	
	NOM_ECG_ELEC_POTL_III	0x013D
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
aVR	ECG Lead AVR	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_AVR	0x0002013E
	Observed Value:	
	NOM_ECG_ELEC_POTL_AVR	0x013E

	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
aVL	ECG Lead AVL	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_AVL	0x0002013F
	Observed Value:	
	NOM_ECG_ELEC_POTL_AVL	0x013F
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
aVF	ECG Lead AVF	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_AVF	0x00020140
	Observed Value:	
	NOM_ECG_ELEC_POTL_AVF	0x0140
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
V	ECG Lead V	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_V	0x00020143
	Observed Value:	
	NOM_ECG_ELEC_POTL_V	0x0143
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
MCL	ECG Lead MCL	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_MCL	0x0002014B
	Observed Value:	
	NOM_ECG_ELEC_POTL_MCL	0x014B
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
V1	ECG Lead V1	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_V1	0x00020103
	Observed Value:	
	NOM_ECG_ELEC_POTL_V1	0x0103
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
V2	ECG Lead V1	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_V2	0x00020104
	Observed Value:	
	NOM_ECG_ELEC_POTL_V2	0x0104
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
V3	ECG Lead V1	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_V3	0x00020105
	Observed Value:	
	NOM_ECG_ELEC_POTL_V3	0x0105
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
V4	ECG Lead V1	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_V4	0x00020106
	Observed Value:	
	NOM_ECG_ELEC_POTL_V4	0x0106
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
V5	ECG Lead V1	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_V5	0x00020107
	Observed Value:	
	NOM_ECG_ELEC_POTL_V5	0x0107
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2

V6	ECG Lead V1	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_V6	0x00020108
	Observed Value:	
	NOM_ECG_ELEC_POTL_V6	0x0108
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
MCL1	ECG Lead MCL1	
	Label:	
	NLS_NOM_ECG_ELEC_POTL_MCL1	0x0002014C
	Observed Value:	
	NOM_ECG_ELEC_POTL_MCL1	0x014C
	Units:	
	NOM_DIM_MILLI_VOLT	0x10B2
Pleth	PLETH wave label	
	Label:	
	NLS_NOM_PULS_OXIM_PLETH	0x00024BB4
	Observed Value:	
	NOM_PLETH	0x4BB4
	Units:	
	NOM_DIM_DIMLESS	0x0200
PlethT	Pleth wave from Telemetry	
	Label:	
	NLS_NOM_PULS_OXIM_PLETH_TELE	0x0002F09B
	Observed Value:	
	NOM_PULS_OXIM_PLETH_TELE	0xF09B
	Units:	
	NOM_DIM_DIMLESS	0x0200
PLETHl	PLETH wave (left)	
	Label:	
	NLS_NOM_PULS_OXIM_PLETH_LEFT	0x0002F08D
	Observed Value:	
	NOM_PULS_OXIM_PLETH_LEFT	0xF08D
	Units:	
	NOM_DIM_DIMLESS	0x0200
PLETHr	PLETH wave (right)	
	Label:	
	NLS_NOM_PULS_OXIM_PLETH_RIGHT	0x0002F08C
	Observed Value:	
	NOM_PULS_OXIM_PLETH_RIGHT	0xF08C
	Units:	
	NOM_DIM_DIMLESS	0x0200
ABP	Arterial Blood Pressure (ABP)	
	Label:	
	NLS_NOM_PRESS_BLD_ART_ABP	0x00024A14
	Observed Value:	
	NOM_PRESS_BLD_ART_ABP	0x4A14
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
ART	Arterial Blood Pressure (ART)	
	Label:	
	NLS_NOM_PRESS_BLD_ART	0x00024A10
	Observed Value:	
	NOM_PRESS_BLD_ART	0x4A10
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Ao	Arterial Blood Pressure in the Aorta (Ao)	
	Label:	
	NLS_NOM_PRESS_BLD_AORT	0x00024A0C
	Observed Value:	
	NOM_PRESS_BLD_AORT	0x4A0C
	Units:	
	NOM_DIM_MMHG	0x0F20

	NOM_DIM_KILO_PASCAL	0x0F03
PAP	Pulmonary Arterial Pressure (PAP)	
	Label:	
	NLS_NOM_PRESS_BLD_ART_PULM	0x00024A1C
	Observed Value:	
	NOM_PRESS_BLD_ART_PULM	0x4A1C
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
CVP	Central Venous Pressure (CVP)	
	Label:	
	NLS_NOM_PRESS_BLD_VEN_CENT	0x00024A44
	Observed Value:	
	NOM_PRESS_BLD_VEN_CENT	0x4A44
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
RAP	Right Atrial Pressure (RAP)	
	Label:	
	NLS_NOM_PRESS_BLD_ATR_RIGHT	0x00024A34
	Observed Value:	
	NOM_PRESS_BLD_ATR_RIGHT	0x4A34
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
LAP	Left Atrial Pressure (LAP)	
	Label:	
	NLS_NOM_PRESS_BLD_ATR_LEFT	0x00024A30
	Observed Value:	
	NOM_PRESS_BLD_ATR_LEFT	0x4A30
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
ICP	Intra-cranial Pressure (ICP)	
	Label:	
	NLS_NOM_PRESS_INTRA_CRAN	0x00025808
	Observed Value:	
	NOM_PRESS_INTRA_CRAN	0x5808
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
UAP	Umbilical Arterial Pressure (UAP)	
	Label:	
	NLS_NOM_PRESS_BLD_ART_UMB	0x00024A28
	Observed Value:	
	NOM_PRESS_BLD_ART_UMB	0x4A28
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
UVP	Umbilical Venous Pressure (UVP)	
	Label:	
	NLS_NOM_PRESS_BLD_VEN_UMB	0x00024A48
	Observed Value:	
	NOM_PRESS_BLD_VEN_UMB	0x4A48
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
FAP	Femoral Arterial Pressure (FAP)	
	Label:	
	NLS_NOM_PRESS_BLD_ART_FEMORAL	0x0002F0BC
	Observed Value:	
	NOM_PRESS_BLD_ART_FEMORAL	0xF0BC
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03

BAP	Brachial Arterial Pressure (BAP)	
	Label:	
	NLS_NOM_PRESS_BLD_ART_BRACHIAL	0x0002F0C0
	Observed Value:	
	NOM_PRESS_BLD_ART_BRACHIAL	0xF0C0
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
IC1	Intracranial Pressure 1 (IC1)	
	Label:	
	NLS_NOM_PRESS_INTRA_CRAN_1	0x0002F0B4
	Observed Value:	
	NOM_PRESS_INTRA_CRAN_1	0xF0B4
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
IC2	Intracranial Pressure 2 (IC2)	
	Label:	
	NLS_NOM_PRESS_INTRA_CRAN_2	0x0002F0B8
	Observed Value:	
	NOM_PRESS_INTRA_CRAN_2	0xF0B8
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
P	unspecific pressure	
	Label:	
	NLS_NOM_PRESS_BLD	0x00024A00
	Observed Value:	
	NOM_PRESS_BLD	0x4A00
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
P1	Generic Pressure 1 (P1)	
	Label:	
	NLS_NOM_PRESS_GEN_1	0x0002F0A4
	Observed Value:	
	NOM_PRESS_GEN_1	0xF0A4
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
P2	Generic Pressure 2 (P2)	
	Label:	
	NLS_NOM_PRESS_GEN_2	0x0002F0A8
	Observed Value:	
	NOM_PRESS_GEN_2	0xF0A8
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
P3	Generic Pressure 3 (P3)	
	Label:	
	NLS_NOM_PRESS_GEN_3	0x0002F0AC
	Observed Value:	
	NOM_PRESS_GEN_3	0xF0AC
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
P4	Generic Pressure 4 (P4)	
	Label:	
	NLS_NOM_PRESS_GEN_4	0x0002F0B0
	Observed Value:	
	NOM_PRESS_GEN_4	0xF0B0
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
CO2	CO2 concentration	

	Label:	
	NLS_NOM_AWAY_CO2	0x000250AC
	Observed Value:	
	NOM_AWAY_CO2	0x50AC
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
O2	Generic oxigen measurement label	
	Label:	
	NLS_NOM_CONC_AWAY_O2	0x00025164
	Observed Value:	
	NOM_CONC_AWAY_O2	0x5164
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
Resp	Imedance RESP wave	
	Label:	
	NLS_NOM_RESP	0x00025000
	Observed Value:	
	NOM_RESP	0x5000
	Units:	
	NOM_DIM_X_OHM	0x10C0
AWF	Airway Flow Wave	
	Label:	
	NLS_NOM_FLOW_AWAY	0x000250D4
	Observed Value:	
	NOM_FLOW_AWAY	0x50D4
AWP	Airway Pressure Wave	
	Label:	
	NLS_NOM_PRESS_AWAY	0x000250F0
	Observed Value:	
	NOM_PRESS_AWAY	0x50F0
AWPin	Airway Pressure Wave - measured in the inspiratory path	
	Label:	
	NLS_NOM_PRESS_AWAY_INSP	0x00025108
	Observed Value:	
	NOM_PRESS_AWAY_INSP	0x5108
AWFin	Airway Flow Wave - measured in the inspiratory path	
	Label:	
	NLS_NOM_VENT_FLOW_INSP	0x0002518C
	Observed Value:	
	NOM_VENT_FLOW_INSP	0x518C
EEG	generic EEG and BIS label	
	Label:	
	NLS_NOM_EEG_ELEC_POTL_CRTX	0x0002592C
	Observed Value:	
	NOM_EEG_ELEC_POTL_CRTX	0x592C
	Units:	
	NOM_DIM_MICRO_VOLT	0x10B3
EEG1	EEG wave channel 1	
	Label:	
	NLS_EEG_NAMES_EEG_CHAN1_LBL	0x800F5401
	Observed Value:	
	NOM_EEG_ELEC_POTL_CRTX	0x592C
	Units:	
	NOM_DIM_MICRO_VOLT	0x10B3
EEG2	EEG wave channel 2	
	Label:	
	NLS_EEG_NAMES_EEG_CHAN2_LBL	0x800F5402
	Observed Value:	
	NOM_EEG_ELEC_POTL_CRTX	0x592C
	Units:	
	NOM_DIM_MICRO_VOLT	0x10B3
Tblood	Tblood	
	Label:	

	NLS_NOM_TEMP_BLD	0x0002E014
	Observed Value:	
	NOM_TEMP_BLD	0xE014
N2	generic N2 label	
	Label:	
	NLS_NOM_CONC_AWAY_N2	0x0002537C
	Observed Value:	
	NOM_CONC_AWAY_N2	0x537C
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
	NOM_DIM_PERCENT	0x0220
N2O	generic Nitrous Oxide label	
	Label:	
	NLS_NOM_CONC_AWAY_N2O	0x000251F0
	Observed Value:	
	NOM_CONC_AWAY_N2O	0x51F0
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
	NOM_DIM_PERCENT	0x0220
ISO	generic Isoflurane label	
	Label:	
	NLS_NOM_CONC_AWAY_ISOFL	0x000251E8
	Observed Value:	
	NOM_CONC_AWAY_ISOFL	0x51E8
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
	NOM_DIM_PERCENT	0x0220
SEV	generic Sevoflurane label	
	Label:	
	NLS_NOM_CONC_AWAY_SEVOFL	0x000251E4
	Observed Value:	
	NOM_CONC_AWAY_SEVOFL	0x51E4
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
	NOM_DIM_PERCENT	0x0220
ENF	generic Enflurane label	
	Label:	
	NLS_NOM_CONC_AWAY_ENFL	0x000251DC
	Observed Value:	
	NOM_CONC_AWAY_ENFL	0x51DC
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
	NOM_DIM_PERCENT	0x0220
HAL	generic Halothane label	
	Label:	
	NLS_NOM_CONC_AWAY_HALOTH	0x000251E0
	Observed Value:	
	NOM_CONC_AWAY_HALOTH	0x51E0
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
	NOM_DIM_PERCENT	0x0220
DES	generic Desflurane label	
	Label:	
	NLS_NOM_CONC_AWAY_DESFL	0x000251D8
	Observed Value:	
	NOM_CONC_AWAY_DESFL	0x51D8
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03

	NOM_DIM_PERCENT	0x0220
AGT	generic Agent label	
	Label:	
	NLS_NOM_CONC_AWAY_AGENT	0x00025388
	Observed Value:	
	NOM_CONC_AWAY_AGENT	0x5388
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
	NOM_DIM_PERCENT	0x0220
AGT1	generic Agent1 label	
	Label:	
	NLS_GASES_NAMES_CONC_AWAY_AGENT1	0x805A5401
	Observed Value:	
	NOM_CONC_AWAY_AGENT	0x5388
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
	NOM_DIM_PERCENT	0x0220
AGT2	generic Agent2 label	
	Label:	
	NLS_GASES_NAMES_CONC_AWAY_AGENT2	0x805A5402
	Observed Value:	
	NOM_CONC_AWAY_AGENT	0x5388
	Units:	
	NOM_DIM_MMHG	0x0F20
	NOM_DIM_KILO_PASCAL	0x0F03
	NOM_DIM_PERCENT	0x0220
P_1	non-specific label for Pressure 1	
	Label:	
	NLS_NOM_EMFC_P1	0x04010030
	Observed Value:	
	NOM_PRESS_BLD	0x4A00
P_2	non-specific label for Pressure 2	
	Label:	
	NLS_NOM_EMFC_P2	0x04010034
	Observed Value:	
	NOM_PRESS_BLD	0x4A00
P_3	non-specific label for Pressure 3	
	Label:	
	NLS_NOM_EMFC_P3	0x04010038
	Observed Value:	
	NOM_PRESS_BLD	0x4A00
P_4	non-specific label for Pressure 4	
	Label:	
	NLS_NOM_EMFC_P4	0x0401003C
	Observed Value:	
	NOM_PRESS_BLD	0x4A00
P_5	non-specific label for Pressure 5	
	Label:	
	NLS_NOM_EMFC_P5	0x04010400
	Observed Value:	
	NOM_PRESS_BLD	0x4A00
P_6	non-specific label for Pressure 6	
	Label:	
	NLS_NOM_EMFC_P6	0x04010404
	Observed Value:	
	NOM_PRESS_BLD	0x4A00
P_7	non-specific label for Pressure 7	
	Label:	
	NLS_NOM_EMFC_P7	0x04010408
	Observed Value:	
	NOM_PRESS_BLD	0x4A00
P_8	non-specific label for Pressure 8	
	Label:	

	NLS_NOM_EMFC_P8	0x0401040C
	Observed Value:	
	NOM_PRESS_BLD	0x4A00
IUP	Intra-Uterine Pressure	
	Label:	
	NLS_NOM_EMFC_IUP	0x04010054
	Observed Value:	
	NOM_PRESS_BLD	0x4A00
AUX	Auxiliary Wave/Parameter	
	Label:	
	NLS_NOM_EMFC_AUX	0x040100B4
	Observed Value:	
	NOM_METRIC_NOS	0xEFFF
vECG	Vector ECG taken from ICG	
	Label:	
	NLS_NOM_EMFC_vECG	0x0401119C
	Observed Value:	
	NOM_METRIC_NOS	0xEFFF
ICG	Impedance Cardiography	
	Label:	
	NLS_NOM_EMFC_ICG	0x040111A0
	Observed Value:	
	NOM_METRIC_NOS	0xEFFF
AWV	Airway Volume Wave	
	Label:	
	NLS_NOM_EMFC_AWV	0x04010668
	Observed Value:	
	NOM_METRIC_NOS	0xEFFF
L V1	Lead V1 - ECG wave label	
	Label:	
	NLS_NOM_EMFC_L_V1	0x04010764
	Observed Value:	
	NOM_ECG_ELEC_POTL_V1	0x0103
L V2	Lead V2 - ECG wave label	
	Label:	
	NLS_NOM_EMFC_L_V2	0x04010768
	Observed Value:	
	NOM_ECG_ELEC_POTL_V2	0x0104
L V3	Lead V3 - ECG wave label	
	Label:	
	NLS_NOM_EMFC_L_V3	0x0401076C
	Observed Value:	
	NOM_ECG_ELEC_POTL_V3	0x0105
L V4	Lead V4 - ECG wave label	
	Label:	
	NLS_NOM_EMFC_L_V4	0x04010770
	Observed Value:	
	NOM_ECG_ELEC_POTL_V4	0x0106
L V5	Lead V5 - ECG wave label	
	Label:	
	NLS_NOM_EMFC_L_V5	0x04010774
	Observed Value:	
	NOM_ECG_ELEC_POTL_V5	0x0107
L V6	Lead V6 - ECG wave label	
	Label:	
	NLS_NOM_EMFC_L_V6	0x04010778
	Observed Value:	
	NOM_ECG_ELEC_POTL_V6	0x0108
L I	Lead I - ECG wave label	
	Label:	
	NLS_NOM_EMFC_L_I	0x0401077C
	Observed Value:	
	NOM_ECG_ELEC_POTL_I	0x0101
L II	Lead II - ECG wave label	
	Label:	

	NLS_NOM_EMFC_L_II	0x04010780
	Observed Value:	
	NOM_ECG_ELEC_POTL_II	0x0102
L III	Lead III - ECG wave label	
	Label:	
	NLS_NOM_EMFC_L_III	0x04010784
	Observed Value:	
	NOM_ECG_ELEC_POTL_III	0x013D
L aVR	Lead aVR - ECG wave label	
	Label:	
	NLS_NOM_EMFC_L_aVR	0x04010788
	Observed Value:	
	NOM_ECG_ELEC_POTL_AVR	0x013E
L aVL	Lead aVL - ECG wave label	
	Label:	
	NLS_NOM_EMFC_L_aVL	0x0401078C
	Observed Value:	
	NOM_ECG_ELEC_POTL_AVL	0x013F
L aVF	Lead aVF - ECG wave label	
	Label:	
	NLS_NOM_EMFC_L_aVF	0x04010790
	Observed Value:	
	NOM_ECG_ELEC_POTL_AVF	0x0140
AWVex	Expiratory Airway Volume Wave. Measured in l.	
	Label:	
	NLS_NOM_EMFC_AWVex	0x04010794
	Observed Value:	
	NOM_METRIC_NOS	0xEFFF
PLETH2	PLETH from the second SpO2/PLETH module	
	Label:	
	NLS_NOM_EMFC_PLETH2	0x0401079C
	Observed Value:	
	NOM_PLETH	0x4BB4
LT EEG	Left channel EEG wave	
	Label:	
	NLS_NOM_EMFC_LT_EEG	0x040107F0
	Observed Value:	
	NOM_EEG_ELEC_POTL_CRTX	0x592C
RT EEG	Right channel EEG wave	
	Label:	
	NLS_NOM_EMFC_RT_EEG	0x0401082C
	Observed Value:	
	NOM_EEG_ELEC_POTL_CRTX	0x592C
BP	Unspecified Blood Pressure	
	Label:	
	NLS_NOM_EMFC_BP	0x04010888
	Observed Value:	
	NOM_PRESS_BLD	0x4A00
AGTs	Anesthetic Agent - secondary agent	
	Label:	
	NLS_NOM_EMFC_AGTs	0x04010CE4
	Observed Value:	
	NOM_CONC_AWAY_AGENT	0x5388
Wave 1	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_WAVE1	0x80AAF001
	Observed Value:	
	depends on configuration	
Wave 2	Placeholder for Vuelink Flex Text	
	Label:	
	NLS_VUELINK_FLX1_NPS_TEXT_WAVE2	0x80AAF003
	Observed Value:	
	depends on configuration	
Wave 3	Placeholder for Vuelink Flex Text	
	Label:	

	NLS_VUELINK_FLX1_NPS_TEXT_WAVE3 Observed Value: depends on configuration	0x80AAF005
Wave 4	Placeholder for Vuelink Flex Text Label: NLS_VUELINK_FLX1_NPS_TEXT_WAVE4 Observed Value: depends on configuration	0x80AAF007
Wave 5	Placeholder for Vuelink Flex Text Label: NLS_VUELINK_FLX1_NPS_TEXT_WAVE5 Observed Value: depends on configuration	0x80AAF009
Wave 6	Placeholder for Vuelink Flex Text Label: NLS_VUELINK_FLX1_NPS_TEXT_WAVE6 Observed Value: depends on configuration	0x80AAF00B
Wave 7	Placeholder for Vuelink Flex Text Label: NLS_VUELINK_FLX1_NPS_TEXT_WAVE7 Observed Value: depends on configuration	0x80AAF00D
Wave 8	Placeholder for Vuelink Flex Text Label: NLS_VUELINK_FLX1_NPS_TEXT_WAVE8 Observed Value: depends on configuration	0x80AAF00F

Attribute IDs

The Attribute ID specifies the type of an attribute in the AttributeList. The IDs are taken from the Object Oriented Elements partition. Unknown attributes should be ignored.

Device P-Alarm List	NOM_ATTR_AL_MON_P_AL_LIST	0x0902
Device T-Alarm List	NOM_ATTR_AL_MON_T_AL_LIST	0x0904
Altitude	NOM_ATTR_ALTITUDE	0x090C
Application Area	NOM_ATTR_AREA_APPL	0x090D
Color	NOM_ATTR_COLOR	0x0911
Device Alert Condition	NOM_ATTR_DEV_AL_COND	0x0916
Display Resolution	NOM_ATTR_DISP_RES	0x0917
Visual Grid	NOM_ATTR_GRID_VIS_I16	0x091A
Association Invoke Id	NOM_ATTR_ID_ASSOC_NO	0x091D
Bed Label	NOM_ATTR_ID_BED_LABEL	0x091E
Object Handle	NOM_ATTR_ID_HANDLE	0x0921
Label	NOM_ATTR_ID_LABEL	0x0924
Label String		

	NOM_ATTR_ID_LABEL_STRING	0x0927
System Model	NOM_ATTR_ID_MODEL	0x0928
Product Specification	NOM_ATTR_ID_PROD_SPECN	0x092D
Object Type	NOM_ATTR_ID_TYPE	0x092F
Line Frequency	NOM_ATTR_LINE_FREQ	0x0935
System Localization	NOM_ATTR_LOCALIZN	0x0937
Metric Info Label	NOM_ATTR_METRIC_INFO_LABEL	0x093C
Metric Info Label String	NOM_ATTR_METRIC_INFO_LABEL_STR	0x093D
Metric Specification	NOM_ATTR_METRIC_SPECN	0x093F
Metric State	NOM_ATTR_METRIC _STAT	0x0940
Measure Mode	NOM_ATTR_MODE_MSMT	0x0945
Operating Mode	NOM_ATTR_MODE_OP	0x0946
Nomenclature Version	NOM_ATTR_NOM_VERS	0x0948
Compound Numeric Observed Value	NOM_ATTR_NU_CMPD_VAL_OBS	0x094B
Numeric Observed Value	NOM_ATTR_NU_VAL_OBS	0x0950
Patient BSA	NOM_ATTR_PT_BSA	0x0956
Pat Demo State	NOM_ATTR_PT_DEMOG_ST	0x0957
Patient Date of Birth	NOM_ATTR_PT_DOB	0x0958
Patient ID	NOM_ATTR_PT_ID	0x095A
Family Name	NOM_ATTR_PT_NAME_FAMILY	0x095C
Given Name	NOM_ATTR_PT_NAME_GIVEN	0x095D
Patient Sex	NOM_ATTR_PT_SEX	0x0961
Patient Type	NOM_ATTR_PT_TYPE	0x0962
Sample Array Calibration Specification	NOM_ATTR_SA_CALIB_I16	0x0964
Compound Sample Array Observed Value	NOM_ATTR_SA_CMPD_VAL_OBS	0x0967
Sample Array Physiological Range	NOM_ATTR_SA_RANGE_PHYS_I16	0x096A
Sample Array Specification	NOM_ATTR_SA_SPECN	0x096D
Sample Array Observed Value	NOM_ATTR_SA_VAL_OBS	0x096E
Scale and Range Specification		

	NOM_ATTR_SCALE_SPECN_I16	0x096F
Safety Standard	NOM_ATTR_STD_SAFETY	0x0982
System ID	NOM_ATTR_SYS_ID	0x0984
System Specification	NOM_ATTR_SYS_SPECN	0x0985
System Type	NOM_ATTR_SYS_TYPE	0x0986
Date and Time	NOM_ATTR_TIME_ABS	0x0987
Sample Period	NOM_ATTR_TIME_PD_SAMP	0x098D
Relative Time	NOM_ATTR_TIME_REL	0x098F
Absolute Time Stamp	NOM_ATTR_TIME_STAMP_ABS	0x0990
Relative Time Stamp	NOM_ATTR_TIME_STAMP_REL	0x0991
Unit Code	NOM_ATTR_UNIT_CODE	0x0996
Enumeration Observed Value	NOM_ATTR_VAL_ENUM_OBS	0x099E
MDS Status	NOM_ATTR_VMS_MDS_STAT	0x09A7
Patient Age	NOM_ATTR_PT_AGE	0x09D8
Patient Height	NOM_ATTR_PT_HEIGHT	0x09DC
Patient Weight	NOM_ATTR_PT_WEIGHT	0x09DF
Sample Array Fixed Values Specification	NOM_ATTR_SA_FIXED_VAL_SPECN	0x0A16
Patient Paced Mode	NOM_ATTR_PT_PACED_MODE	0x0A1E
Internal Patient ID	NOM_ATTR_PT_ID_INT	0xF001
Private Attribute	NOM_SAT_O2_TONE_FREQ	0xF008
Private Attribute	NOM_ATTR_CMPD_REF_LIST	0xF009
IP Address Information	NOM_ATTR_NET_ADDR_INFO	0xF100
Protocol Support	NOM_ATTR_PCOL_SUPPORT	0xF101
Notes1	NOM_ATTR_PT_NOTES1	0xF129
Notes2	NOM_ATTR_PT_NOTES2	0xF12A
Time for Periodic Polling	NOM_ATTR_TIME_PD_POLL	0xF13E
Patient BSA Formula	NOM_ATTR_PT_BSA_FORMULA	0xF1EC
Mds General System Info	NOM_ATTR_MDS_GEN_INFO	0xF1FA
no of prioritized objects for poll request		

	NOM_ATTR_POLL_OBJ_PRIO_NUM	0xF228
Numeric Object Priority List	NOM_ATTR_POLL_NU_PRIO_LIST	0xF239
Wave Object Priority List	NOM_ATTR_POLL_RTSA_PRIO_LIST	0xF23A
Metric Modality	NOM_ATTR_METRIC_MODALITY	0xF294

The attributes are arranged in the following attribute groups:

Alert Monitor Group	NOM_ATTR_GRP_AL_MON	0x0801
Metric Observed Value Group	NOM_ATTR_GRP_METRIC_VAL_OBS	0x0803
Patient Demographics Attribute Group	NOM_ATTR_GRP_PT_DEMOG	0x0807
System Application Attribute Group	NOM_ATTR_GRP_SYS_APPL	0x080A
System Identification Attribute Group	NOM_ATTR_GRP_SYS_ID	0x080B
System Production Attribute Group	NOM_ATTR_GRP_SYS_PROD	0x080C
VMO Dynamic Attribute Group	NOM_ATTR_GRP_VMO_DYN	0x0810
VMO Static Attribute Group	NOM_ATTR_GRP_VMO_STATIC	0x0811

Component IDs

The Component IDs specify system components such as the entries in the Production Specification attribute of the Medical Device Service object. A Component ID is a PrivateOid and is not assigned to any nomenclature partition.

for the overall product	ID_COMP_PRODUCT	0x0008
for the specific bundle	ID_COMP_CONFIG	0x0010
for the boot code	ID_COMP_BOOT	0x0018
mainboard component	ID_COMP_MAIN_BD	0x0050
application software component	ID_COMP_APPL_SW	0x0058

Unit Codes

The Unit Codes describe the dimension of a physiological measurement. They are grouped in the Units partition.

NOS	(no dimension)	
	NOM_DIM_NOS	0
/	(/)	
	NOM_DIM_DIV	2
-	(no dimension)	
	NOM_DIM_DIMLESS	512
%	(percentage)	
	NOM_DIM_PERCENT	544
ppth	(parts per thousand)	
	NOM_DIM_PARTS_PER_THOUSAND	576
ppm	(parts per million)	
	NOM_DIM_PARTS_PER_MILLION	608
mol/mol	(mole per mole)	
	NOM_DIM_X_MOLE_PER_MOLE	864
ppb	(parts per billion)	
	NOM_DIM_PARTS_PER_BILLION	672
ppt	(parts per trillion)	
	NOM_DIM_PARTS_PER_TRILLION	704
pH	(pH)	
	NOM_DIM_PH	992
drop	(vital signs count drop)	
	NOM_DIM_DROP	1024
rbc	(vital signs count red blood cells)	
	NOM_DIM_RBC	1056
beat	(vital signs count beat)	
	NOM_DIM_BEAT	1088
breath	(vital signs count breath)	
	NOM_DIM_BREATH	1120
cell	(vital signs count cells)	
	NOM_DIM_CELL	1152
cough	(vital signs count cough)	
	NOM_DIM_COUGH	1184
sigh	(vital signs count sigh)	
	NOM_DIM_SIGH	1216
%PCV	(percent of packed cell volume)	
	NOM_DIM_PCT_PCV	1248
m	(meter)	
	NOM_DIM_X_M	1280
cm	(centimeter)	
	NOM_DIM_CENTI_M	1297
mm	(millimeter)	
	NOM_DIM_MILLI_M	1298
µm	(micro-meter)	
	NOM_DIM_MICRO_M	1299
in	(inch)	
	NOM_DIM_X_INCH	1376
ml/m2	(used e.g. for SI and ITBVI)	
	NOM_DIM_MILLI_L_PER_M_SQ	1426
/m	(per meter)	
	NOM_DIM_PER_X_M	1440
/mm	(per millimeter)	

	NOM_DIM_PER_MILLI_M	1458
m2	(used e.g. for BSA calculation)	
	NOM_DIM_SQ_X_M	1472
in2	(used e.g. for BSA calculation)	
	NOM_DIM_SQ_X_INCH	1504
m3	(cubic meter)	
	NOM_DIM_CUBIC_X_M	1568
cm3	(cubic centimeter)	
	NOM_DIM_CUBIC_CENTI_M	1585
l	(liter)	
	NOM_DIM_X_L	1600
ml	(milli-liters used e.g. for EVLW ITBV SV)	
	NOM_DIM_MILLI_L	1618
ml/breath	(milli-liter per breath)	
	NOM_DIM_MILLI_L_PER_BREATH	1650
/cm3	(per cubic centimeter)	
	NOM_DIM_PER_CUBIC_CENTI_M	1681
/l	(per liter)	
	NOM_DIM_PER_X_L	1696
1/nl	(per nano-liter)	
	NOM_DIM_PER_NANO_LITER	1716
g	(gram)	
	NOM_DIM_X_G	1728
kg	(kilo-gram)	
	NOM_DIM_KILO_G	1731
mg	(milli-gram)	
	NOM_DIM_MILLI_G	1746
µg	(micro-gram)	
	NOM_DIM_MICRO_G	1747
ng	(nono-gram)	
	NOM_DIM_NANO_G	1748
lb	(pound)	
	NOM_DIM_X_LB	1760
oz	(ounce)	
	NOM_DIM_X_OZ	1792
/g	(per gram)	
	NOM_DIM_PER_X_G	1824
g-m	(used e.g. for LVSW RVSW)	
	NOM_DIM_X_G_M	1856
kg-m	(used e.g. for RCW LCW)	
	NOM_DIM_KILO_G_M	1859
g-m/m2	(used e.g. for LVSWI and RVSWI)	
	NOM_DIM_X_G_M_PER_M_SQ	1888
kg-m/m2	(used e.g. for LCWI and RCWI)	
	NOM_DIM_KILO_G_M_PER_M_SQ	1891
kg-m2	(gram meter squared)	
	NOM_DIM_KILO_G_M_SQ	1923
kg/m2	(kilo-gram per square meter)	
	NOM_DIM_KG_PER_M_SQ	1955
kg/m3	(kilo-gram per cubic meter)	
	NOM_DIM_KILO_G_PER_M_CUBE	1987
g/cm3	(gram per cubic meter)	
	NOM_DIM_X_G_PER_CM_CUBE	2016
mg/cm3	(milli-gram per cubic centimeter)	
	NOM_DIM_MILLI_G_PER_CM_CUBE	2034
µg/cm3	(micro-gram per cubic centimeter)	

	NOM_DIM_MICRO_G_PER_CM_CUBE	2035
ng/cm3	(nano-gram per cubic centimeter)	
	NOM_DIM_NANO_G_PER_CM_CUBE	2036
g/l	(gram per liter)	
	NOM_DIM_X_G_PER_L	2048
g/dl	(used e.g. for Hb)	
	NOM_DIM_X_G_PER_DL	2112
mg/dl	(milli-gram per deciliter)	
	NOM_DIM_MILLI_G_PER_DL	2130
g/ml	(gram per milli-liter)	
	NOM_DIM_X_G_PER_ML	2144
mg/ml	(milli-gram per milli-liter)	
	NOM_DIM_MILLI_G_PER_ML	2162
µg/ml	(micro-gram per milli-liter)	
	NOM_DIM_MICRO_G_PER_ML	2163
ng/ml	(nano-gram per milli-liter)	
	NOM_DIM_NANO_G_PER_ML	2164
sec	(seconds)	
	NOM_DIM_SEC	2176
msec	(milli-seconds)	
	NOM_DIM_MILLI_SEC	2194
µsec	(micro-seconds)	
	NOM_DIM_MICRO_SEC	2195
min	(minutes)	
	NOM_DIM_MIN	2208
hrs	(hours)	
	NOM_DIM_HR	2240
days	(days)	
	NOM_DIM_DAY	2272
weeks	(weeks)	
	NOM_DIM_WEEKS	2304
months	(months)	
	NOM_DIM_MON	2336
years	(years)	
	NOM_DIM_YR	2368
TOD	(time of day)	
	NOM_DIM_TOD	2400
date	(date)	
	NOM_DIM_DATE	2432
/sec	(per second)	
	NOM_DIM_PER_X_SEC	2464
Hz	(hertz)	
	NOM_DIM_HZ	2496
/min	(per minute used e.g. for the PVC count numerical value)	
	NOM_DIM_PER_MIN	2528
/hour	(per hour)	
	NOM_DIM_PER_HR	2560
/day	(per day)	
	NOM_DIM_PER_DAY	2592
/week	(per week)	
	NOM_DIM_PER_WK	2624
/month	(per month)	
	NOM_DIM_PER_MO	2656
/year	(per year)	
	NOM_DIM_PER_YR	2688
bpm	(beats per minute used e.g. for HR/PULSE)	

	NOM_DIM_BEAT_PER_MIN	2720
puls/min	(puls per minute)	
	NOM_DIM_PULS_PER_MIN	2752
rpm	(respiration breathes per minute)	
	NOM_DIM_RESP_PER_MIN	2784
m/sec	(meter per second)	
	NOM_DIM_X_M_PER_SEC	2816
mm/sec	(speed for recordings)	
	NOM_DIM_MILLI_M_PER_SEC	2834
l/min/m2	(used for CI)	
	NOM_DIM_X_L_PER_MIN_PER_M_SQ	2848
ml/min/m2	(used for DO2I VO2I O2AVI)	
	NOM_DIM_MILLI_L_PER_MIN_PER_M_SQ	2866
m2/sec	(square meter per second)	
	NOM_DIM_SQ_X_M_PER_SEC	2880
cm2/sec	(square centimeter per second)	
	NOM_DIM_SQ_CENTI_M_PER_SEC	2897
m3/sec	(cubic meter per second)	
	NOM_DIM_CUBIC_X_M_PER_SEC	2912
cm3/sec	(cubic centimeter per second)	
	NOM_DIM_CUBIC_CENTI_M_PER_SEC	2929
l/sec	(liter per second)	
	NOM_DIM_X_L_PER_SEC	3040
l/min	(liter per minutes)	
	NOM_DIM_X_L_PER_MIN	3072
dl/min	(deciliter per second)	
	NOM_DIM_DECI_L_PER_MIN	3088
ml/min	(used for DO2 VO2 ALVENT)	
	NOM_DIM_MILLI_L_PER_MIN	3090
l/hour	(liter per hour)	
	NOM_DIM_X_L_PER_HR	3104
ml/hour	(milli-liter per hour)	
	NOM_DIM_MILLI_L_PER_HR	3122
l/day	(liter per day)	
	NOM_DIM_X_L_PER_DAY	3136
ml/day	(milli-liter per day)	
	NOM_DIM_MILLI_L_PER_DAY	3154
ml/kg	(used e.g. for EVLWI)	
	NOM_DIM_MILLI_L_PER_KG	3186
kg/sec	(kilo-gram per second)	
	NOM_DIM_KILO_G_PER_SEC	3299
g/min	(gram per minute)	
	NOM_DIM_X_G_PER_MIN	3328
kg/min	(kilo-gram per minute)	
	NOM_DIM_KILO_G_PER_MIN	3331
mg/min	(milli-gram per minute)	
	NOM_DIM_MILLI_G_PER_MIN	3346
µg/min	(micro-gram per minute)	
	NOM_DIM_MICRO_G_PER_MIN	3347
ng/min	(nano-gram per minute)	
	NOM_DIM_NANO_G_PER_MIN	3348
g/hour	(gram per hour)	
	NOM_DIM_X_G_PER_HR	3360
kg/hour	(kilo-gram per hour)	
	NOM_DIM_KILO_G_PER_HR	3363
mg/hour	(milli-gram per hour)	

		NOM_DIM_MILLI_G_PER_HR	3378
µg/hour	(micro-gram per hour)	NOM_DIM_MICRO_G_PER_HR	3379
ng/hr	(nano-gram per hour)	NOM_DIM_NANO_G_PER_HR	3380
kg/day	(kilo-gram per day)	NOM_DIM_KILO_G_PER_DAY	3395
g/kg/min	(gram per kilo-gram per minute)	NOM_DIM_X_G_PER_KG_PER_MIN	3456
mg/kg/min	(milli-gram per kilo-gram per minute)	NOM_DIM_MILLI_G_PER_KG_PER_MIN	3474
µg/kg/min	(micro-gram per kilo-gram per minute)	NOM_DIM_MICRO_G_PER_KG_PER_MIN	3475
ng/kg/min	(nano-gram per kilo-gram per minute)	NOM_DIM_NANO_G_PER_KG_PER_MIN	3476
g/kg/hour	(gram per kilo-gram per hour)	NOM_DIM_X_G_PER_KG_PER_HR	3488
mg/kg/hour	(mili-gram per kilo-gram per hour)	NOM_DIM_MILLI_G_PER_KG_PER_HR	3506
µg/kg/hour	(micro-gram per kilo-gram per hour)	NOM_DIM_MICRO_G_PER_KG_PER_HR	3507
ng/kg/hour	(nano-gram per kilo-gram per hour)	NOM_DIM_NANO_G_PER_KG_PER_HR	3508
kg/l/sec	(kilo-gram per liter per second)	NOM_DIM_KILO_G_PER_L_SEC	3555
kg/m/sec	(kilo-gram per meter per second)	NOM_DIM_KILO_G_PER_M_PER_SEC	3683
kg-m/sec	(kilo-gram meter per second)	NOM_DIM_KILO_G_M_PER_SEC	3715
N-s	(newton seconds)	NOM_DIM_X_NEWTON_SEC	3744
N	(newton)	NOM_DIM_X_NEWTON	3776
Pa	(pascal)	NOM_DIM_X_PASCAL	3840
hPa	(hekto-pascal)	NOM_DIM_HECTO_PASCAL	3842
kPa	(kilo-pascal)	NOM_DIM_KILO_PASCAL	3843
mmHg	(mm mercury)	NOM_DIM_MMHG	3872
cmH2O	(centimeter H20)	NOM_DIM_CM_H2O	3904
mBar	(milli-bar)	NOM_DIM_MILLI_BAR	3954
J	(Joules)	NOM_DIM_X_JOULES	3968
eV	(electronvolts)	NOM_DIM_EVOLT	4000
W	(watt)	NOM_DIM_X_WATT	4032
mW	(milli-watt)	NOM_DIM_MILLI_WATT	4050
nW	(nano-watt)	NOM_DIM_NANO_WATT	4052
pW	(pico-watt)		

	NOM_DIM_PICO_WATT	4053
Dyn-sec/cm^5	(dyne second per cm^5)	
	NOM_DIM_X_DYNE_PER_SEC_PER_CM5	4128
A	(ampere)	
	NOM_DIM_X_AMPS	4160
mA	(milli-ampereused e.g. for the battery indications)	
	NOM_DIM_MILLI_AMPS	4178
C	(coulomb)	
	NOM_DIM_X_COULOMB	4192
µC	(micro-coulomb)	
	NOM_DIM_MICRO_COULOMB	4211
V	(volts)	
	NOM_DIM_X_VOLT	4256
mV	(milli-volt)	
	NOM_DIM_MILLI_VOLT	4274
µV	(micro-volt)	
	NOM_DIM_MICRO_VOLT	4275
Ohm	(Ohm)	
	NOM_DIM_X_OHM	4288
kOhm	(kilo-ohm)	
	NOM_DIM_OHM_K	4291
F	(farad)	
	NOM_DIM_X_FARAD	4352
°K	(kelvin)	
	NOM_DIM_KELVIN	4384
°F	(degree-fahrenheit)	
	NOM_DIM_FAHR	4416
cd	(candela)	
	NOM_DIM_X_CANDELA	4480
mOsm	(milli-osmole)	
	NOM_DIM_MILLI_OSM	4530
mol	(mole)	
	NOM_DIM_X_MOLE	4544
mmol	(milli-mole)	
	NOM_DIM_MILLI_MOLE	4562
mEq	(milli-equivalents)	
	NOM_DIM_MILLI_EQUIV	4594
mOsm/l	(milli-osmole per liter)	
	NOM_DIM_MILLI_OSM_PER_L	4626
mmol/l	(used for HB)	
	NOM_DIM_MILLI_MOLE_PER_L	4722
µmol/l	(micro-mol per liter)	
	NOM_DIM_MICRO_MOLE_PER_L	4723
mEq/l	(milli-equivalents per liter)	
	NOM_DIM_MILLI_EQUIV_PER_L	4850
mEq/day	(milli-equivalents per day)	
	NOM_DIM_MILLI_EQUIV_PER_DAY	5202
i.u.	(international unit)	
	NOM_DIM_X_INTL_UNIT	5472
mi.u.	(mili-international unit)	
	NOM_DIM_MILLI_INTL_UNIT	5490
i.u./cm3	(international unit per cubic centimeter)	
	NOM_DIM_X_INTL_UNIT_PER_CM_CUBE	5504
mi.u./cm3	(mili-international unit per cubic centimeter)	
	NOM_DIM_MILLI_INTL_UNIT_PER_CM_CUBE	5522
i.u./ml	(international unit per milli-liter)	

	NOM_DIM_X_INTL_UNIT_PER_ML	5600
i.u./min	(international unit per minute)	
	NOM_DIM_X_INTL_UNIT_PER_MIN	5664
mi.u./ml	(milli-international unit per milli-liter)	
	NOM_DIM_MILLI_INTL_UNIT_PER_ML	5618
mi.u./min	(milli-international unit per minute)	
	NOM_DIM_MILLI_INTL_UNIT_PER_MIN	5682
i.u./hour	(international unit per hour)	
	NOM_DIM_X_INTL_UNIT_PER_HR	5696
mi.u./hour	(milli-international unit per hour)	
	NOM_DIM_MILLI_INTL_UNIT_PER_HR	5714
i.u./kg/min	(international unit per kilo-gram per minute)	
	NOM_DIM_X_INTL_UNIT_PER_KG_PER_MIN	5792
mi.u./kg/min	(milli-international unit per kilo-gram per minute)	
	NOM_DIM_MILLI_INTL_UNIT_PER_KG_PER_MIN	5810
i.u./kg/hour	(international unit per kilo-gram per hour)	
	NOM_DIM_X_INTL_UNIT_PER_KG_PER_HR	5824
mi.u./kg/hour	(milli-international unit per kilo-gram per hour)	
	NOM_DIM_MILLI_INTL_UNIT_PER_KG_PER_HR	5842
ml/cmH2O	(milli-liter per centimeter H2O)	
	NOM_DIM_MILLI_L_PER_CM_H2O	5906
cmH2O/l/sec	(centimeter H2O per second)	
	NOM_DIM_CM_H2O_PER_L_PER_SEC	5920
ml2/sec	(milli-liter per second)	
	NOM_DIM_MILLI_L_SQ_PER_SEC	5970
cmH2O/%	(centimeter H2O per percent)	
	NOM_DIM_CM_H2O_PER_PERCENT	5984
DS*m2/cm5	(used for SVRI and PVRI)	
	NOM_DIM_DYNE_SEC_PER_M_SQ_PER_CM_5	6016
°C	(degree-celsius)	
	NOM_DIM_DEGC	6048
cmH2O/l	(centimeter H2O per liter)	
	NOM_DIM_CM_H2O_PER_L	6144
mmHg/%	(milli-meter mercury per percent)	
	NOM_DIM_MM_HG_PER_PERCENT	6176
kPa/%	(kilo-pascal per percent)	
	NOM_DIM_KILO_PA_PER_PERCENT	6211
l/mmHg	(liter per mmHg)	
	NOM_DIM_X_L_PER_MM_HG	6272
ml/mmHg	(milli-liter per milli-meter Hg)	
	NOM_DIM_MILLI_L_PER_MM_HG	6290
mAh	(milli-ampere per hour used e.g. for the battery indications)	
	NOM_DIM_MILLI_AMP_HR	6098
ml/dl	(used for CaO2 CvO2 Ca-vO2)	
	NOM_DIM_MILLI_L_PER_DL	6418
dB	(decibel)	
	NOM_DIM_DECIBEL	6432
g/mg	(gram per milli-gram)	
	NOM_DIM_X_G_PER_MILLI_G	6464
mg/mg	(milli-gram per milli-gram)	
	NOM_DIM_MILLI_G_PER_MILLI_G	6482
bpm/l	(beats per minute per liter)	
	NOM_DIM_BEAT_PER_MIN_PER_X_L	6496
bpm/ml	(beats per minute per milli-liter)	
	NOM_DIM_BEAT_PER_MIN_PER_MILLI_L	6514
1/(min*l)	(per minute per liter)	

	NOM_DIM_PER_X_L_PER_MIN	6528
m/min	(meter per minute)	
	NOM_DIM_X_M_PER_MIN	6560
cm/min	(speed for recordings)	
	NOM_DIM_CENTI_M_PER_MIN	6577
pg/ml	(pico-gram per milli-liter)	
	NOM_DIM_PICO_G_PER_ML	2165
ug/l	(micro-gram per liter)	
	NOM_DIM_MICRO_G_PER_L	2067
ng/l	(nano-gram per liter)	
	NOM_DIM_NANO_G_PER_L	2068
$/mm^3$	(per cubic millimeter)	
	NOM_DIM_PER_CUBIC_MILLI_M	1682
mm^3	(cubic milli-meter)	
	NOM_DIM_CUBIC_MILLI_M	1586
u/l	(intl. units per liter)	
	NOM_DIM_X_INTL_UNIT_PER_L	5568
/l	(10^6 intl. units per liter)	
	NOM_DIM_MEGA_INTL_UNIT_PER_L	5573
mol/kg	(mole per kilo-gram)	
	NOM_DIM_MILLI_MOL_PER_KG	4946
mcg/dl	(micro-gram per deci-liter)	
	NOM_DIM_MICRO_G_PER_DL	2131
mg/l	(milli-gram per liter)	
	NOM_DIM_MILLI_G_PER_L	2066
/ul	(micro-liter)	
	NOM_DIM_PER_MICRO_L	1715
complx	(-)	
	NOM_DIM_COMPLEX	61440
count	(count as a dimension)	
	NOM_DIM_COUNT	61441
part	(part)	
	NOM_DIM_PART	61442
puls	(puls)	
	NOM_DIM_PULS	61443
µV p-p	(micro-volt peak to peak)	
	NOM_DIM_UV_PP	61444
µV2	(micor-volt square)	
	NOM_DIM_UV_SQ	61445
lumen	(lumen)	
	NOM_DIM_LUMEN	61447
lb/in2	(pound per square inch)	
	NOM_DIM_LB_PER_INCH_SQ	61448
mmHg/s	(milli-meter mercury per second)	
	NOM_DIM_MM_HG_PER_SEC	61449
ml/s	(milli-liter per second)	
	NOM_DIM_ML_PER_SEC	61450
bpm/ml	(beat per minute per milli-liter)	
	NOM_DIM_BEAT_PER_MIN_PER_ML_C	61451
J/day	(joule per day)	
	NOM_DIM_X_JOULE_PER_DAY	61536
kJ/day	(kilo joule per day)	
	NOM_DIM_KILO_JOULE_PER_DAY	61539
MJ/day	(mega joule per day)	
	NOM_DIM_MEGA_JOULE_PER_DAY	61540
cal	(calories)	

	NOM_DIM_X_CALORIE	61568
kcal	(kilo calories)	
	NOM_DIM_KILO_CALORIE	61571
10**6 cal	(million calories)	
	NOM_DIM_MEGA_CALORIE	61572
cal/day	(calories per day)	
	NOM_DIM_X_CALORIE_PER_DAY	61600
kcal/day	(kilo-calories per day)	
	NOM_DIM_KILO_CALORIE_PER_DAY	61603
Mcal/day	(mega calories per day)	
	NOM_DIM_MEGA_CALORIE_PER_DAY	61604
cal/ml	(calories per milli-liter)	
	NOM_DIM_X_CALORIE_PER_ML	61632
kcal/ml	(kilo calories per ml)	
	NOM_DIM_KILO_CALORIE_PER_ML	61635
J/ml	(Joule per milli-liter)	
	NOM_DIM_X_JOULE_PER_ML	61664
kJ/ml	(kilo-joules per milli-liter)	
	NOM_DIM_KILO_JOULE_PER_ML	61667
RPM	(revolutions per minute)	
	NOM_DIM_X_REV_PER_MIN	61696
l/(mn*l*kg)	(per minute per liter per kilo)	
	NOM_DIM_PER_L_PER_MIN_PER_KG	61728
l/mbar	(liter per milli-bar)	
	NOM_DIM_X_L_PER_MILLI_BAR	61760
ml/mbar	(milli-liter per milli-bar)	
	NOM_DIM_MILLI_L_PER_MILLI_BAR	61778
l/kg/hr	(liter per kilo-gram per hour)	
	NOM_DIM_X_L_PER_KG_PER_HR	61792
ml/kg/hr	(milli-liter per kilogram per hour)	
	NOM_DIM_MILLI_L_PER_KG_PER_HR	61810
bar/l/s	(bar per liter per sec)	
	NOM_DIM_X_BAR_PER_LITER_PER_SEC	61824
mbar/l/s	(milli-bar per liter per sec)	
	NOM_DIM_MILLI_BAR_PER_LITER_PER_SEC	61842
bar/l	(bar per liter)	
	NOM_DIM_X_BAR_PER_LITER	61856
mbar/l	(bar per liter)	
	NOM_DIM_MILLI_BAR_PER_LITER	61874
V/mV	(volt per milli-volt)	
	NOM_DIM_VOLT_PER_MILLI_VOLT	61888
cmH2O/uV	(cm H2O per micro-volt)	
	NOM_DIM_CM_H2O_PER_MICRO_VOLT	61920
J/l	(joule per liter)	
	NOM_DIM_X_JOULE_PER_LITER	61952
l/bar	(liter per bar)	
	NOM_DIM_X_L_PER_BAR	61984
m/mV	(meter per milli-volt)	
	NOM_DIM_X_M_PER_MILLI_VOLT	62016
mm/mV	(milli-meter per milli-volt)	
	NOM_DIM_MILLI_M_PER_MILLI_VOLT	62034
l/min/kg	(liter per minute per kilo-gram)	
	NOM_DIM_X_L_PER_MIN_PER_KG	62048
ml/min/kg	(milli-liter per minute per kilo-gram)	
	NOM_DIM_MILLI_L_PER_MIN_PER_KG	62066
Pa/l/s	(pascal per liter per sec)	

	NOM_DIM_X_PASCAL_PER_L_PER_SEC	62080
hPa/l/s	(hPa per liter per sec)	
	NOM_DIM_HECTO_PASCAL_PER_L_PER_SEC	62082
kPa/l/s	(kPa per liter per sec)	
	NOM_DIM_KILO_PASCAL_PER_L_PER_SEC	62083
ml/Pa	(milli-liter per pascal)	
	NOM_DIM_MILLI_L_PER_X_PASCAL	62112
ml/hPa	(milli-liter per hecto-pascal)	
	NOM_DIM_MILLI_L_PER_HECTO_PASCAL	62114
ml/kPa	(milli-liter per kilo-pascal)	
	NOM_DIM_MILLI_L_PER_KILO_PASCAL	62115
mmHg/l/s	(mm)	
	NOM_DIM_MM_HG_PER_X_L_PER_SEC	62144

Alert Codes

The first column in the tables below shows the alert source, the second column shows the associated alert code and the third column contains the alert text which would be displayed by the monitor. The XXX in the alert text is a placeholder for the actual alert source. It is filled depending on the alert source. Note that the alert text depends on the localization of your monitor.

The least significant bit of the alert codes listed below is used to identify the source of an alert (refer to "Alert Monitor Object" on page 99). If the alert code is marked with a (*), the associated alert source is from the object oriented nomenclature partition and hence the least significant bit of the alert code is set to 1.

NOTE On monitors with SW Rev. G.0 or lower some alert codes will only be issued correctly, if the connected MMS or FMS has the same or a higher SW revision.

ECG/HR/Arrhy

Alert Source	Alert Code	Alert Text
NOM_ECG_ELEC_POTL	NOM_EVT_EQUIP_MALF	XXXXXX EQUIP MALF
NOM_ECG_ELEC_POTL	NOM_EVT_LEADS_OFF	XXXXXX LEADS OFF
NOM_ECG_ELEC_POTL	NOM_EVT_LEAD_DISCONN_YELLOW	!! ECG LEADS OFF
NOM_ECG_ELEC_POTL	NOM_EVT_LEADS_OFF	!!!ECG LEADS OFF
NOM_ECG_ELEC_POTL	NOM_EVT_NOISY	ECG NOISY SIGNAL
NOM_ECG_LEAD_C	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_RA	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_LA	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_LL	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_RL	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_C1FR	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_C2FR	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_C3FR	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_C4FR	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_C5FR	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_C6FR	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_EASI_S	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_fI	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_fE	NOM_EVT_LEAD_DISCONN	XXX LEAD OFF
NOM_ECG_LEAD_C	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_LEAD_RA	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_LEAD_LA	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_LEAD_LL	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_LEAD_RL	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_LEAD_C1FR	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_LEAD_C2FR	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_LEAD_C3FR	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_LEAD_C4FR	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_LEAD_C5FR	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_LEAD_C6FR	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_LEAD_EASI_S	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_LEAD_fI	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_LEAD_fE	NOM_EVT_NOISY	ECG EL. NOISY XXX
NOM_ECG_ELEC_POTL	NOM_EVT_SIG_UNANALYZEABLE	CANNOT ANALYZE ECG
NOM_ECG_ELEC_POTL	NOM_EVT_UNDEF	XXXXXX UNKN. ALERT
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_ASYSTOLE	*** ASYSTOLE
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_V_FIB_TACHY	*** VENT FIB/TACH
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_BRADY_EXTREME	*** EXTREME BRADY
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_TACHY_EXTREME	*** EXTREME TACHY
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_LO	** XXXXXX LOW

Alert Source	Alert Code	Alert Text
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_PACER_NOT_PACING	** PACER NT PACING
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_PACING_NON_CAPT	** PACER NOT CAPT
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_SV_TACHY	** SVT
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_BEAT_MISSED	** MISSED BEAT
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_PAUSE	** PAUSE
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_CARD_BEAT_RATE_IRREG	** IRREGULAR HR
NOM_ECG_V_P_C_CNT	NOM_EVT_STAT_ECG_AL_SOME_OFF	SOME ECG ALRMS OFF
NOM_ECG_V_P_C_CNT	NOM_EVT_STAT_ECG_AL_ALL_OFF	ALL ECG ALARMS OFF
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_TACHY	*** VTACH
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_P_C_RATE	** PVCs/min HIGH
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_RHY	** VENT RHYTHM
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_P_C_RUN	** RUN PVCs HIGH
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_P_C_PAIR	** PAIR PVCs
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_P_C_RonT	** R-ON-T PVCs
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_BIGEM	** VENT BIGEMINY
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_TRIGEM	** VENT TRIGEMINY
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_TACHY_NON_SUST	** NON-SUSTAIN VT
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_P_C_MULTIFORM	** MULTIFORM PVCs
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_LO	*
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_HI	*
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_PACER_NOT_PACING	* PACER NT PACING
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_PACING_NON_CAPT	* PACER NOT CAPT
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_SV_TACHY	* SVT
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_BEAT_MISSED	* MISSED BEAT
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_PAUSE	* PAUSE
NOM_ECG_CARD_BEAT_RATE	NOM_EVT_ECG_CARD_BEAT_RATE_IRREG	* IRREGULAR HR
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_P_C_RATE	* PVCs/min HIGH
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_RHY	* VENT RHYTHM
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_P_C_RUN	* RUN PVCs HIGH
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_P_C_PAIR	* PAIR PVCs
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_P_C_RonT	* R-ON-T PVCs
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_BIGEM	* VENT BIGEMINY
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_TRIGEM	* VENT TRIGEMINY
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_TACHY_NON_SUST	* NON-SUSTAIN VT
NOM_ECG_V_P_C_CNT	NOM_EVT_ECG_V_P_C_MULTIFORM	* MULTIFORM PVCs

ST

Alert Source	Alert Code	Alert Text
NOM_ECG_AMPL_ST	NOM_EVT_SIG_UNANALYZEABLE	CANNOT ANALYZE ST
NOM_ECG_AMPL_ST_I	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_II	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_III	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_AVF	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_AVL	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_AVR	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_V	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_MCL	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_V1	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_V2	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_V3	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_V4	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_V5	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_V6	NOM_EVT_LO	** XXXXXX LOW
NOM_ECG_AMPL_ST_I	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST_II	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST_III	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST_AVF	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST_AVL	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST_AVR	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST_V	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST_MCL	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST_V1	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST_V2	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST_V3	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST_V4	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST_V5	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST_V6	NOM_EVT_HI	** XXXXXX HIGH
NOM_ECG_AMPL_ST	NOM_EVT_ST_MULTI	**ST MULTI XXX,XXX

QT Analysis

Alert Source	Alert Code	Alert Text
	NOM_EVT_SIG_UNANALYZEABLE	CANNOT ANALYZE QT
NOM_ECG_LEAD_C, NOM_ECG_LEAD_RA. **NOM_ECG_LEAD_LA,** **NOM_ECG_LEAD_LL,** **NOM_ECG_LEAD_RL,** **NOM_ECG_LEAD_C1,** **NOM_ECG_LEAD_C2,** **NOM_ECG_LEAD_C3,** **NOM_ECG_LEAD_C4,** **NOM_ECG_LEAD_C5,** **NOM_ECG_LEAD_C6,** **NOM_ECG_LEAD_A,** **NOM_ECG_LEAD_S,** **NOM_ECG_LEAD_I** **NOM_ECG_LEAD_E**	NOM_EVT_HI	** XXXXXX HIGH

Resp

Alert Source	Alert Code	Alert Text
NOM_RESP	NOM_EVT_LEADS_OFF	XXXXXX LEADS OFF
NOM_RESP	NOM_EVT_ERRATIC	XXXXXX ERRATIC
NOM_RESP_RATE	NOM_EVT_APNEA	*** APNEA
NOM_RESP_RATE	NOM_EVT_LO	** XXXXXX LOW
NOM_RESP_RATE	NOM_EVT_HI	** XXXXXX HIGH

Derived Measurements

Alert Source	Alert Code	Alert Text
NOM_PRESS_CEREB_PERF	NOM_EVT_ADVIS_SRC_CHK	XXXXXX CHK SOURCES
NOM_RES_VASC_SYS	NOM_EVT_ADVIS_SRC_CHK	XXXXXX CHK SOURCES
NOM_RES_VASC_SYS_INDEX	NOM_EVT_ADVIS_SRC_CHK	XXXXXX CHK SOURCES
NOM_TEMP_DIFF	NOM_EVT_ADVIS_SRC_CHK	XXXXXX CHK SOURCES
NOM_SAT_DIFF_O2_ART_VEN	NOM_EVT_ADVIS_SRC_CHK	XXXXXX CHK SOURCES
NOM_PULS_OXIM_SAT_O2_DIFF	NOM_EVT_ADVIS_SRC_CHK	XXXXXX CHK SOURCES
NOM_RATE_DIFF_CARD_BEAT_PULSE	NOM_EVT_ADVIS_SRC_CHK	XXXXXX CHK SOURCES
NOM_PRESS_CEREB_PERF	NOM_EVT_ADVIS_UNIT_CHK	XXXXXX CHK UNITS
NOM_RES_VASC_SYS	NOM_EVT_ADVIS_UNIT_CHK	XXXXXX CHK UNITS
NOM_RES_VASC_SYS_INDEX	NOM_EVT_ADVIS_UNIT_CHK	XXXXXX CHK UNITS
NOM_TEMP_DIFF	NOM_EVT_ADVIS_UNIT_CHK	XXXXXX CHK UNITS
NOM_SAT_DIFF_O2_ART_VEN	NOM_EVT_ADVIS_UNIT_CHK	XXXXXX CHK UNITS
NOM_PULS_OXIM_SAT_O2_DIFF	NOM_EVT_ADVIS_UNIT_CHK	XXXXXX CHK UNITS
NOM_RATE_DIFF_CARD_BEAT_PULSE	NOM_EVT_ADVIS_UNIT_CHK	XXXXXX CHK UNITS
NOM_RES_VASC_SYS	NOM_EVT_ADVIS_PRESUMED_CVP	XXXXXXSET CVP USED
NOM_RES_VASC_SYS_INDEX	NOM_EVT_ADVIS_PRESUMED_CVP	XXXXXXSET CVP USED
NOM_PRESS_CEREB_PERF	NOM_EVT_HI	** XXXXXX HIGH
NOM_PRESS_CEREB_PERF	NOM_EVT_LO	** XXXXXX LOW

C.O./CCO

Alert Source	Alert Code	Alert Text
NOM_VMD_CARD_OUTPUT	NOM_EVT_EQUIP_MALF+1	XXXXXX EQUIP MALF
NOM_OUTPUT_CARD_CTS	NOM_EVT_XDUCR_DISCONN	CCO/Tbl NO TRANSD.
NOM_OUTPUT_CARD	NOM_EVT_XDUCR_DISCONN	XXXXXX NO TRANSDUC
NOM_TEMP_BLOOD	NOM_EVT_RANGE_ERR	XXXXXX OVERRANGE
NOM_TEMP_BLOOD	NOM_EVT_MSMT_RANGE_OVER	XXXXXX OVERRANGE
NOM_TEMP_BLOOD	NOM_EVT_HI	** XXXXXX HIGH
NOM_TEMP_BLOOD	NOM_EVT_LO	** XXXXXX LOW
NOM_OUTPUT_CARD_CTS	NOM_EVT_UNSUPPORTED	CCO NOT SUPPORTED
NOM_OUTPUT_CARD_CTS	NOM_EVT_SRC_ABSENT	CCO NO XXX
NOM_OUTPUT_CARD_CTS	NOM_EVT_ADVIS_SRC_CHK	CCO XXX INVALID
NOM_OUTPUT_CARD_CTS	NOM_EVT_STAT_PULSE_SRC_RANGE_OVER	CCO PULSE OVERRANG
NOM_OUTPUT_CARD_CTS	NOM_EVT_ADVIS_CALIB_REQD	CCO NO CALIBRATION
NOM_OUTPUT_CARD_CTS	NOM_EVT_STAT_PRESS_SRC_RANGE_OVER	CCO PRESS OVERRANG
NOM_OUTPUT_CARD_CTS	NOM_EVT_SIG_UNANALYZEABLE	CCO BAD PRESS SIGN
NOM_OUTPUT_CARD_CTS	NOM_EVT_MSMT_RANGE_OVER	XXXXXX OVERRANGE
NOM_OUTPUT_CARD_CTS	NOM_EVT_ADVIS_CALIB_AND_ZERO_CHK	CCO RECALIBRATE
NOM_OUTPUT_CARD_CTS	NOM_EVT_HI	** XXXXXX HIGH
NOM_OUTPUT_CARD_CTS	NOM_EVT_LO	** XXXXXX LOW
NOM_OUTPUT_CARD_INDEX_CTS	NOM_EVT_ADVIS_BSA_REQD	CCI NO BSA
NOM_OUTPUT_CARD_INDEX_CTS	NOM_EVT_MSMT_RANGE_OVER	XXXXXX OVERRANGE
NOM_OUTPUT_CARD_INDEX_CTS	NOM_EVT_HI	** XXXXXX HIGH
NOM_OUTPUT_CARD_INDEX_CTS	NOM_EVT_LO	** XXXXXX LOW

EEG

Alert Source	Alert Code	Alert Text
NOM_EEG_ELEC_POTL_CRTX	NOM_EVT_EQUIP_MALF	XXXXXX EQUIP MALF
NOM_EEG_ELEC_POTL_CRTX	NOM_EVT_XDUCR_DISCONN	XXXXXX NO TRANSDUC
NOM_EEG_ELEC_POTL_CRTX	NOM_EVT_LEADS_OFF	XXXXXX LEADS OFF
NOM_EEG_ELEC_POTL_CRTX	NOM_EVT_MSMT_RANGE_OVER	XXXXXX OVERRANGE
NOM_EEG_ELEC_POTL_CRTX	NOM_EVT_IMPED_HI	EEG IMPEDANCE HIGH
NOM_EEG_ELEC_POTL_CRTX	NOM_EVT_MUSCLE_NOISE	EEG MUSCLE NOISE
NOM_EEG_ELEC_POTL_CRTX	NOM_EVT_LINE_NOISE	EEG LINE NOISE
NOM_OBJ_CHAN_1	NOM_EVT_LEAD_DISCONN+1	EEG1 LEAD OFF XXX
NOM_OBJ_CHAN_2	NOM_EVT_LEAD_DISCONN+1	EEG2 LEAD OFF XXX
NOM_OBJ_CHAN_1	NOM_EVT_LEADS_OFF+1	XXXXXX LEADS OFF
NOM_OBJ_CHAN_2	NOM_EVT_LEADS_OFF+1	XXXXXX LEADS OFF
NOM_OBJ_CHAN_1	NOM_EVT_MSMT_RANGE_OVER+1	XXXXXX OVERRANGE
NOM_OBJ_CHAN_2	NOM_EVT_MSMT_RANGE_OVER+1	XXXXXX OVERRANGE
NOM_OBJ_CHAN_1	NOM_EVT_MUSCLE_NOISE+1	XXXXXXMUSCLE NOISE
NOM_OBJ_CHAN_2	NOM_EVT_LINE_NOISE+1	XXXXXX LINE NOISE
NOM_OBJ_CHAN_1	NOM_EVT_IMPED_HI+1	EEG1 IMPED. HIGH
NOM_OBJ_CHAN_1	NOM_EVT_IMPEDS_HI+1	EEG1 IMPED. HIGH
NOM_OBJ_CHAN_2	NOM_EVT_IMPED_HI+1	EEG2 IMPED. HIGH
NOM_OBJ_CHAN_2	NOM_EVT_IMPEDS_HI+1	EEG2 IMPED. HIGH

BIS

Alert Source	Alert Code	Alert Text
NOM_DEV_ANALY_BISPECTRAL_INDEX_VMD	NOM_EVT_EQUIP_MALF+1	XXXXXX EQUIP MALF
NOM_DEV_ANALY_BISPECTRAL_INDEX_VMD	NOM_EVT_DISCONN+1	BIS ENGINE DISCONN
NOM_DEV_ANALY_BISPECTRAL_INDEX_VMD	NOM_EVT_VOLTAGE_OUT_OF_RANGE+1	BIS OVERCURRENT
NOM_EEG_BISPECTRAL_INDEX	NOM_EVT_INCOMPAT	BIS ENGINE INCOMPT
NOM_DEV_ANALY_BISPECTRAL_INDEX_VMD	NOM_EVT_MALF+1	BIS ENGINE MALFUNC
NOM_DEV_ANALY_BISPECTRAL_INDEX_VMD	NOM_EVT_XDUCR_DISCONN+1	BIS DSC DISCONN
NOM_DEV_ANALY_BISPECTRAL_INDEX_VMD	NOM_EVT_STAT_FW_UPDATE_IN_PROGRESS+1	BIS DSC UPDATE
NOM_OBJ_XDUCR	NOM_EVT_INCOMPAT+1	BIS DSC INCOMPT
NOM_DEV_ANALY_BISPECTRAL_INDEX_VMD	NOM_EVT_XDUCR_MALF+1	BIS DSC MALFUNC
NOM_DEV_ANALY_BISPECTRAL_INDEX_VMD	NOM_EVT_SENSOR_DISCONN+1	BIS SENSOR DISCONN
NOM_DEV_ANALY_BISPECTRAL_INDEX_VMD	NOM_EVT_SENSOR_MALF+1	BIS SENSOR MALFUNC

Alert Source	Alert Code	Alert Text
NOM_OBJ_SENSOR	NOM_EVT_INCOMPAT+1	BIS SENSOR INCOMPT
NOM_OBJ_SENSOR	NOM_EVT_EXH+1	BIS SENSOR USAGE
NOM_ELECTRODE_IMPED	NOM_EVT_ADVIS_CHK	BIS IMPEDANCE CHCK
NOM_EEG_BISPECTRAL_INDEX	NOM_EVT_LEAD_DISCONN	BIS LEAD OFF
NOM_EEG_BISPECTRAL_INDEX	NOM_EVT_IMPED_HI	BIS HIGH IMPEDANCE
NOM_EEG_BIS_SIG_QUAL_INDEX	NOM_EVT_SIG_LO	BIS SQI < 15%
NOM_EEG_BISPECTRAL_INDEX	NOM_EVT_ADVIS_CHK	BIS IMPEDANCE CHCK
NOM_EEG_BISPECTRAL_INDEX	NOM_EVT_LEAD_DISCONN	BIS LEAD OFF
NOM_EEG_BISPECTRAL_INDEX	NOM_EVT_IMPED_HI	BIS HIGH IMPEDANCE
NOM_EEG_BIS_SIG_QUAL_INDEX	NOM_EVT_LO	BIS SQI < 50%
NOM_EEG_ELEC_POTL_CRTX	NOM_EVT_ABSENT	BIS ISOELECTRC EEG
NOM_EEG_BISPECTRAL_INDEX	NOM_EVT_HI	** XXXXXX HIGH
NOM_EEG_BISPECTRAL_INDEX	NOM_EVT_LO	** XXXXXX LOW
NOM_EEG_BISPECTRAL_INDEX	NOM_EVT_DISCONN	BISx DISCONNECTED
NOM_EEG_BISPECTRAL_INDEX	NOM_EVT_INCOMPAT	BISx INCOMPATIBLE
NOM_EEG_BISPECTRAL_INDEX	NOM_EVT_MALF	BISx MALFUNCTION
NOM_OBJ_SENSOR	NOM_EVT_SENSOR_DISCONN+1	BIS ELECTR. DISC.
NOM_OBJ_CABLE	NOM_EVT_INCOMPAT+1	BIS CABLE INCOMPAT
NOM_OBJ_CABLE	NOM_EVT_EXH+1	BIS CABLE USAGE

Temp

*

Alert Source	Alert Code	Alert Text
* any temperature (e.g.**NOM_TEMP**)	NOM_EVT_EQUIP_MALF	XXXXXX EQUIP MALF
* any temperature (e.g.**NOM_TEMP**)	NOM_EVT_XDUCR_DISCONN	XXXXXX NO TRANSDUC
* any temperature (e.g.**NOM_TEMP**)	NOM_EVT_MSMT_RANGE_OVER	XXXXXX OVERRANGE
* any temperature (e.g.**NOM_TEMP**)	NOM_EVT_HI	** XXXXXX HIGH
* any temperature (e.g.**NOM_TEMP**)	NOM_EVT_LO	** XXXXXX LOW

Invasive Pressure

Alert Source	Alert Code	Alert Text
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_EQUIP_MALF	XXXXXX EQUIP MALF
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_XDUCR_DISCONN	XXX NO TRANSDUCER
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_XDUCR_MALF	XXX TRANSDUC MALF
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_ADVIS_CALIB_AND_ZERO_CHK	XXX ZERO+CHECK CAL
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_MSMT_RANGE_OVER	XXX OVERRANGE
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_WAVE_ARTIF_ERR	XXX ARTIFACT
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_ADVIS_GAIN_DECR	XXXXXX REDUCE SIZE
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_WAVE_OSCIL_ABSENT	XXX NON-PULSATILE
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_NOISY	XXX NOISY SIGNAL
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_HI	** XXXXXX HIGH
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_LO	** XXXXXX LOW
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_HI	** XXXXXX HIGH
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_LO	** XXXXXX LOW
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_HI	** XXXXXX HIGH
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_LO	** XXXXXX LOW
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_MSMT_DISCONN	*** XXX DISCONNECT
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_EXTR_LO	*** XXXXXX LOW
* any pressure (e.g.**NOM_PRESS_BLD**)	NOM_EVT_EXTR_HI	*** XXXXXX HIGH
NOM_PULS_RATE	NOM_EVT_HI	** XXXXXX HIGH
NOM_PULS_RATE	NOM_EVT_LO	** XXXXXX LOW
NOM_PULS_RATE	NOM_EVT_BRADY	*** BRADY (Pulse)
NOM_PULS_RATE	NOM_EVT_TACHY	*** TACHY (Pulse)

SpO_2

Alert Source	Alert Code	Alert Text
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_EQUIP_MALF	XXXXXX EQUIP MALF
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_SENSOR_MALF	XXXXXX SENSOR MALF
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_XDUCR_DISCONN	XXXXXX NO SENSOR
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_MSMT_INTERF_ERR	XXXXXX INTERFERNCE
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_ADVIS_SENSOR_CHK	XXXXXX UNKN.SENSOR
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_NOISY	XXXXXX NOISY SIGN.
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_STAT_FW_UPDATE_IN_PROGRESS	XXXXXX UPGRADE
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_WAVE_OSCIL_ABSENT	XXXXXX NON-PULSAT.

Alert Source	Alert Code	Alert Text
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_ERRATIC	XXXXXX ERRATIC
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_STAT_LEARN	XXXXXX SEARCHING
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_SUST	XXXXXX EXTD.UPDATE
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_MSMT_RANGE_UNDER	XXXXXX PULSE?
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_SENSOR_DISCONN	XXXXXX SENSOR OFF
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_WAVE_SIG_QUAL_ERR	XXXXXX POOR SIGNAL
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_SIG_LO	XXXXXX LOW PERF
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_HI	** XXXXXX HIGH
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_LO	** XXXXXX LOW
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_DESAT	*** DESAT
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_LO	** XXXXXX LOW
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_HI	** XXXXXX HIGH
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_BRADY	*** BRADY (Pulse)
any SpO2 (e.g. **NOM_PULS_OXIM_SAT_O2_***)	NOM_EVT_TACHY	*** TACHY (Pulse)

SvO$_2$

Alert Source	Alert Code	Alert Text
NOM_SAT_O2(_VEN)	NOM_EVT_EQUIP_MALF	XXXXXX EQUIP MALF
NOM_SAT_O2(_VEN)	NOM_EVT_CONFIG_ERR	XXXXXX CONFIG ERROR
NOM_SAT_O2(_VEN)	NOM_EVT_STAT_OPT_MOD_SENSOR_CONN	SvO2 CONNCT OPTMOD
NOM_SAT_O2(_VEN)	NOM_EVT_OPTIC_MODULE_ABSENT	XXXXXX NO OPTMOD
NOM_SAT_O2(_VEN)	NOM_EVT_STAT_CALIB_PREINS_RUNNING	SvO2 PRE-INS CALIB
NOM_SAT_O2(_VEN)	NOM_EVT_CALIB_FAIL	XXXXXX CAL FAILED
NOM_SAT_O2(_VEN)	NOM_EVT_ADVIS_CALIB_REQD	XXXXXX CAL REQUIRED
NOM_SAT_O2(_VEN)	NOM_EVT_STAT_CALIB_MODE	XXXXXX CAL MODE
NOM_SAT_O2(_VEN)	NOM_EVT_SIG_LO	XXXXXX LOW LIGHT
NOM_SAT_O2(_VEN)	NOM_EVT_MSMT_ERR	XXXXXX CANNOT MEAS
NOM_SAT_O2(_VEN)	NOM_EVT_INTENS_LIGHT_ERR	XXXXXX LIGHT INTENS
NOM_SAT_O2(_VEN)	NOM_EVT_STAT_CALIB_LIGHT_RUNNING	XXXXXX LIGHT CALIB
NOM_SAT_O2(_VEN)	NOM_EVT_STAT_CALIB_INVIVO_RUNNING	XXXXXX IN-VIVO CALIB
NOM_SAT_O2(_VEN)	NOM_EVT_STAT_OPT_MOD_SENSOR_WARMING	XXXXXX OPTMOD WARMUP
NOM_SAT_O2(_VEN)	NOM_EVT_STAT_FW_UPDATE_IN_PROGRESS	XXXXXX UPGRADE
NOM_SAT_O2(_VEN)	NOM_EVT_INCOMPAT	XXXXXX INCOMPAT.
NOM_SAT_O2(_VEN)	NOM_EVT_OPTIC_MODULE_DEFECT	XXXXXX OPTMOD MALF
NOM_SAT_O2(_VEN)	NOM_EVT_HI	** XXXXXX HIGH
NOM_SAT_O2(_VEN)	NOM_EVT_LO	** XXXXXX LOW

CO$_2$

Alert Source	Alert Code	Alert Text
NOM_AWAY_CO2	NOM_EVT_EQUIP_MALF	CO2 EQUIP MALF
NOM_AWAY_CO2	NOM_EVT_EQUIP_MALF	CO2 EQUIP MALF
NOM_AWAY_CO2	NOM_EVT_XDUCR_DISCONN	XXXXXX NO TRANSDUC
NOM_AWAY_CO2	NOM_EVT_CALIB_FAIL	CO2 FAILED CAL
NOM_AWAY_CO2	NOM_EVT_WAIT_CAL	CO2 WAIT CAL2
NOM_AWAY_CO2	NOM_EVT_STAT_CALIB_RUNNING	XXXXXX CAL RUNNING
NOM_AWAY_CO2	NOM_EVT_STAT_CALIB_MODE	CO2 CAL MODE
NOM_AWAY_CO2	NOM_EVT_ADVIS_CALIB_AND_ZERO_CHK	CO2 CHECK CAL
NOM_AWAY_CO2	NOM_EVT_STAT_SENSOR_WARMING	CO2 SENSOR WARMUP
NOM_AWAY_CO2	NOM_EVT_ADVIS_CHANGE_SCALE	XXXXXXCHANGE SCALE
NOM_AWAY_CO2	NOM_EVT_SW_VER_UNK	CO2 UPDATE FW
NOM_AWAY_CO2	NOM_EVT_TUBE_DISCONN	CO2 NO TUBING
NOM_AWAY_CO2	NOM_EVT_TUBE_OCCL	CO2 OCCLUSION
NOM_AWAY_CO2	NOM_EVT_MSMT_RANGE_OVER	XXXXXX OVERRANGE
NOM_AWAY_CO2	NOM_EVT_TUBE_OBSTRUC	CO2 PURGING

Alert Source	Alert Code	Alert Text
NOM_AWAY_CO2	NOM_EVT_STAT_ZERO_RUNNING	CO2 AUTO ZERO
NOM_AWAY_CO2_ET	NOM_EVT_HI	** XXXXXX HIGH
NOM_AWAY_CO2_ET	NOM_EVT_LO	** XXXXXX LOW
NOM_AWAY_CO2_INSP_MIN	NOM_EVT_HI	** XXXXXX HIGH
NOM_AWAY_RESP_RATE	NOM_EVT_APNEA	*** APNEA
NOM_AWAY_RESP_RATE	NOM_EVT_LO	** XXXXXX LOW
NOM_AWAY_RESP_RATE	NOM_EVT_HI	** XXXXXX HIGH

AGM

Alert Source	Alert Code	Alert Text
NOM_VMD_GAS_ANALY	NOM_EVT_INCOMPAT+1	XXX INCOMPATIBLE
NOM_VMD_GAS_ANALY	NOM_EVT_MALF+1	XXX MALFUNCTION
NOM_VMD_GAS_ANALY	NOM_EVT_MALF+1	XXX MALFUNCTION
NOM_VMD_GAS_ANALY	NOM_EVT_STAT_STANDBY+1	XXX STANDBY
NOM_VMD_GAS_ANALY	NOM_EVT_STAT_DISCONN+1	XXX NOT AVAILABLE
NOM_VMD_GAS_ANALY	NOM_EVT_STAT_SELFTEST_RUNNING+1	XXX SELFTEST
NOM_VMD_GAS_ANALY	NOM_EVT_OBSTRUC+1	XXX OCCLUSION
NOM_VMD_GAS_ANALY	NOM_EVT_OBSTRUC+1	XXX OCCLUSION
NOM_VMD_GAS_ANALY	NOM_EVT_MSMT_INOP+1	XXX UNABLE TO MEAS
NOM_VMD_GAS_ANALY	NOM_EVT_MSMT_INOP+1	XXX UNABLE TO MEAS
NOM_VMD_GAS_ANALY	NOM_EVT_MSMT_RANGE_OVER+1	XXXXXX OVERRANGE
NOM_VMD_GAS_ANALY	NOM_EVT_STAT_CALIB_RUNNING+1	XXX ZERO RUNNING
NOM_VMD_GAS_ANALY	NOM_EVT_WARMING+1	XXX WARMUP
NOM_VMD_GAS_ANALY	NOM_EVT_CALIB_FAIL+1	XXX ZERO FAILED
NOM_VMD_GAS_ANALY	NOM_EVT_MSMT_ERR+1	XXX ACCURACY?
NOM_VMD_GAS_ANALY	NOM_EVT_STAT_AL_OFF+1	XXX ALARM SUPPRESS
NOM_VMD_GAS_ANALY	NOM_EVT_BREATH_ABSENT+1	XXX NO BREATH
NOM_AWAY_CO2	NOM_EVT_MSMT_INOP	XXX UNABLE TO MEAS
NOM_AWAY_CO2	NOM_EVT_DISTURB	XXX MEAS DISTURBED
NOM_AWAY_CO2	NOM_EVT_ADVIS_CHANGE_SCALE	XXXXXXCHANGE SCALE
NOM_AWAY_O2	NOM_EVT_MALF	O2 SENSOR MALFUNCT
NOM_AWAY_O2	NOM_EVT_CALIB_FAIL	O2 ZERO FAILED
NOM_AWAY_O2	NOM_EVT_MSMT_INOP	XXX UNABLE TO MEAS
NOM_AWAY_O2	NOM_EVT_DISTURB	XXX MEAS DISTURBED
NOM_AWAY_O2	NOM_EVT_ADVIS_CHANGE_SCALE	XXXXXXCHANGE SCALE
NOM_AWAY_N2O	NOM_EVT_MSMT_INOP	XXX UNABLE TO MEAS
NOM_AWAY_N2O	NOM_EVT_DISTURB	XXX MEAS DISTURBED
NOM_AWAY_N2O	NOM_EVT_ADVIS_CHANGE_SCALE	XXXXXXCHANGE SCALE
NOM_AWAY_AGENT	NOM_EVT_GAS_AGENT_IDENT_MALF	AGT ID MALFUNCTION
NOM_AWAY_DESFL	NOM_EVT_GAS_AGENT_IDENT_MALF	AGT ID MALFUNCTION
NOM_AWAY_ENFL	NOM_EVT_GAS_AGENT_IDENT_MALF	AGT ID MALFUNCTION
NOM_AWAY_HALOTH	NOM_EVT_GAS_AGENT_IDENT_MALF	AGT ID MALFUNCTION
NOM_AWAY_SEVOFL	NOM_EVT_GAS_AGENT_IDENT_MALF	AGT ID MALFUNCTION
NOM_AWAY_ISOFL	NOM_EVT_GAS_AGENT_IDENT_MALF	AGT ID MALFUNCTION
NOM_AWAY_AGENT	NOM_EVT_CALIB_FAIL	AGT ID ZERO FAILED
NOM_AWAY_DESFL	NOM_EVT_CALIB_FAIL	AGT ID ZERO FAILED
NOM_AWAY_ENFL	NOM_EVT_CALIB_FAIL	AGT ID ZERO FAILED
NOM_AWAY_HALOTH	NOM_EVT_CALIB_FAIL	AGT ID ZERO FAILED
NOM_AWAY_SEVOFL	NOM_EVT_CALIB_FAIL	AGT ID ZERO FAILED
NOM_AWAY_ISOFL	NOM_EVT_CALIB_FAIL	AGT ID ZERO FAILED

Alert Source	Alert Code	Alert Text
NOM_AWAY_AGENT	NOM_EVT_ADVIS_GAS_AGENT_CHK	CHECK AGENT
NOM_AWAY_DESFL	NOM_EVT_ADVIS_GAS_AGENT_CHK	CHECK AGENT
NOM_AWAY_ENFL	NOM_EVT_ADVIS_GAS_AGENT_CHK	CHECK AGENT
NOM_AWAY_HALOTH	NOM_EVT_ADVIS_GAS_AGENT_CHK	CHECK AGENT
NOM_AWAY_SEVOFL	NOM_EVT_ADVIS_GAS_AGENT_CHK	CHECK AGENT
NOM_AWAY_ISOFL	NOM_EVT_ADVIS_GAS_AGENT_CHK	CHECK AGENT
NOM_AWAY_AGENT	NOM_EVT_MSMT_INOP	XXX UNABLE TO MEAS
NOM_AWAY_DESFL	NOM_EVT_MSMT_INOP	XXX UNABLE TO MEAS
NOM_AWAY_ENFL	NOM_EVT_MSMT_INOP	XXX UNABLE TO MEAS
NOM_AWAY_HALOTH	NOM_EVT_MSMT_INOP	XXX UNABLE TO MEAS
NOM_AWAY_SEVOFL	NOM_EVT_MSMT_INOP	XXX UNABLE TO MEAS
NOM_AWAY_ISOFL	NOM_EVT_MSMT_INOP	XXX UNABLE TO MEAS
NOM_AWAY_AGENT	NOM_EVT_MSMT_RESTART	AGT MEAS RESTARTNG
NOM_AWAY_DESFL	NOM_EVT_MSMT_RESTART	AGT MEAS RESTARTNG
NOM_AWAY_ENFL	NOM_EVT_MSMT_RESTART	AGT MEAS RESTARTNG
NOM_AWAY_HALOTH	NOM_EVT_MSMT_RESTART	AGT MEAS RESTARTNG
NOM_AWAY_SEVOFL	NOM_EVT_MSMT_RESTART	AGT MEAS RESTARTNG
NOM_AWAY_ISOFL	NOM_EVT_MSMT_RESTART	AGT MEAS RESTARTNG
NOM_AWAY_AGENT	NOM_EVT_DISTURB	XXX MEAS DISTURBED
NOM_AWAY_DESFL	NOM_EVT_DISTURB	XXX MEAS DISTURBED
NOM_AWAY_ENFL	NOM_EVT_DISTURB	XXX MEAS DISTURBED
NOM_AWAY_HALOTH	NOM_EVT_DISTURB	XXX MEAS DISTURBED
NOM_AWAY_SEVOFL	NOM_EVT_DISTURB	XXX MEAS DISTURBED
NOM_AWAY_ISOFL	NOM_EVT_DISTURB	XXX MEAS DISTURBED
NOM_AWAY_AGENT	NOM_EVT_CONTAM	GAS CONTAMINANT
NOM_AWAY_DESFL	NOM_EVT_CONTAM	GAS CONTAMINANT
NOM_AWAY_ENFL	NOM_EVT_CONTAM	GAS CONTAMINANT
NOM_AWAY_HALOTH	NOM_EVT_CONTAM	GAS CONTAMINANT
NOM_AWAY_SEVOFL	NOM_EVT_CONTAM	GAS CONTAMINANT
NOM_AWAY_ISOFL	NOM_EVT_CONTAM	GAS CONTAMINANT
NOM_AWAY_AGENT	NOM_EVT_TOO_MANY_AGENTS	TOO MANY AGENTS
NOM_AWAY_DESFL	NOM_EVT_TOO_MANY_AGENTS	TOO MANY AGENTS
NOM_AWAY_ENFL	NOM_EVT_TOO_MANY_AGENTS	TOO MANY AGENTS
NOM_AWAY_HALOTH	NOM_EVT_TOO_MANY_AGENTS	TOO MANY AGENTS
NOM_AWAY_SEVOFL	NOM_EVT_TOO_MANY_AGENTS	TOO MANY AGENTS
NOM_AWAY_ISOFL	NOM_EVT_TOO_MANY_AGENTS	TOO MANY AGENTS
NOM_AWAY_AGENT	NOM_EVT_ADVIS_CHANGE_SCALE	XXXXXXCHANGE SCALE
NOM_AWAY_DESFL	NOM_EVT_ADVIS_CHANGE_SCALE	XXXXXXCHANGE SCALE
NOM_AWAY_ENFL	NOM_EVT_ADVIS_CHANGE_SCALE	XXXXXXCHANGE SCALE
NOM_AWAY_HALOTH	NOM_EVT_ADVIS_CHANGE_SCALE	XXXXXXCHANGE SCALE
NOM_AWAY_SEVOFL	NOM_EVT_ADVIS_CHANGE_SCALE	XXXXXXCHANGE SCALE
NOM_AWAY_ISOFL	NOM_EVT_ADVIS_CHANGE_SCALE	XXXXXXCHANGE SCALE

Alert Source	Alert Code	Alert Text
NOM_AWAY_AGENT	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_DESFL	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_ENFL	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_HALOTH	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_SEVOFL	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_ISOFL	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_N2	NOM_EVT_ADVIS_CHANGE_SCALE	XXXXXXCHANGE SCALE
NOM_AWAY_CO2_ET	NOM_EVT_LO	** XXXXXX LOW
NOM_AWAY_RESP_RATE	NOM_EVT_LO	** XXXXXX LOW
NOM_CONC_AWAY_O2_INSP	NOM_EVT_LO	** XXXXXX LOW
NOM_CONC_AWAY_AGENT_ET	NOM_EVT_LO	** XXXXXX LOW
NOM_CONC_AWAY_AGENT_INSP	NOM_EVT_LO	** XXXXXX LOW
NOM_CONC_AWAY_HALOTH_ET	NOM_EVT_LO	** XXXXXX LOW
NOM_CONC_AWAY_HALOTH_INSP	NOM_EVT_LO	** XXXXXX LOW
NOM_CONC_AWAY_ENFL_ET	NOM_EVT_LO	** XXXXXX LOW
NOM_CONC_AWAY_ENFL_INSP	NOM_EVT_LO	** XXXXXX LOW
NOM_CONC_AWAY_ISOFL_ET	NOM_EVT_LO	** XXXXXX LOW
NOM_CONC_AWAY_ISOFL_INSP	NOM_EVT_LO	** XXXXXX LOW
NOM_CONC_AWAY_SEVOFL_ET	NOM_EVT_LO	** XXXXXX LOW
NOM_CONC_AWAY_SEVOFL_INSP	NOM_EVT_LO	** XXXXXX LOW
NOM_CONC_AWAY_DESFL_ET	NOM_EVT_LO	** XXXXXX LOW
NOM_CONC_AWAY_DESFL_INSP	NOM_EVT_LO	** XXXXXX LOW
NOM_AWAY_CO2_ET	NOM_EVT_HI	** XXXXXX HIGH
NOM_AWAY_CO2_INSP_MIN	NOM_EVT_HI	** XXXXXX HIGH
NOM_AWAY_RESP_RATE	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_O2_INSP	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_N2O_INSP	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_AGENT_ET	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_AGENT_INSP	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_HALOTH_ET	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_HALOTH_INSP	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_ENFL_ET	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_ENFL_INSP	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_ISOFL_ET	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_ISOFL_INSP	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_SEVOFL_ET	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_SEVOFL_INSP	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_DESFL_ET	NOM_EVT_HI	** XXXXXX HIGH
NOM_CONC_AWAY_DESFL_INSP	NOM_EVT_HI	** XXXXXX HIGH
NOM_AWAY_RESP_RATE	NOM_EVT_APNEA	*** APNEA
NOM_CONC_AWAY_O2_INSP	NOM_EVT_O2_SUPPLY_LO	***inO2 LOW OXYGEN

Alert Source	Alert Code	Alert Text
NOM_VMD_GAS_ANALY	NOM_EVT_ADVIS_WATER_TRAP_CHK+1	CHECK WATERTRAP
NOM_VMD_GAS_ANALY	NOM_EVT_STAT_OFF+1	XXX SWITCHED OFF
NOM_VMD_GAS_ANALY	NOM_EVT_COMP_MALF+1	XXX COMPONENT MALF
NOM_AWAY_AGENT	NOM_EVT_AGENT_MEAS_MALF	AGT MEAS MALFUNCT
NOM_AWAY_DESFL	NOM_EVT_AGENT_MEAS_MALF	AGT MEAS MALFUNCT
NOM_AWAY_ENFL	NOM_EVT_AGENT_MEAS_MALF	AGT MEAS MALFUNCT
NOM_AWAY_HALOTH	NOM_EVT_AGENT_MEAS_MALF	AGT MEAS MALFUNCT
NOM_AWAY_SEVOFL	NOM_EVT_AGENT_MEAS_MALF	AGT MEAS MALFUNCT
NOM_AWAY_ISOFL	NOM_EVT_AGENT_MEAS_MALF	AGT MEAS MALFUNCT
NOM_AWAY_AGENT	NOM_EVT_STAT_AGENT_CALC_RUNNING	AGENT CALCULATING
NOM_AWAY_DESFL	NOM_EVT_STAT_AGENT_CALC_RUNNING	AGENT CALCULATING
NOM_AWAY_ENFL	NOM_EVT_STAT_AGENT_CALC_RUNNING	AGENT CALCULATING
NOM_AWAY_HALOTH	NOM_EVT_STAT_AGENT_CALC_RUNNING	AGENT CALCULATING
NOM_AWAY_SEVOFL	NOM_EVT_STAT_AGENT_CALC_RUNNING	AGENT CALCULATING
NOM_AWAY_ISOFL	NOM_EVT_STAT_AGENT_CALC_RUNNING	AGENT CALCULATING
NOM_AWAY_AGENT	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_DESFL	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_ENFL	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_HALOTH	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_SEVOFL	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_ISOFL	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_AGENT	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_DESFL	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_ENFL	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_HALOTH	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_SEVOFL	NOM_EVT_AGENT_MIX	AGENT MIXTURE
NOM_AWAY_ISOFL	NOM_EVT_AGENT_MIX	AGENT MIXTURE

System

Alert Source	Alert Code	Alert Text
NOM_OBJ_NETWORK	NOM_EVT_STAT_DISCONN+1	Unsupported LAN
NOM_OBJ_NETWORK	NOM_EVT_MALF+1	No Central Monit.
NOM_OBJ_QUICKLINK	NOM_EVT_IRREG+1	Bad Server Link
NOM_OBJ_QUICKLINK	NOM_EVT_UNSUPPORTED+1	XXXXXX UNSUPPORTD
NOM_OBJ_SPEAKER	NOM_EVT_MALF+1	Speaker Malfunct.
NOM_OBJ_INPUT_DEV	NOM_EVT_MALF+1	User I/F Malfunct.
NOM_OBJ_HIF_KEY	NOM_EVT_MALF+1	Check Keyboard
NOM_OBJ_HIF_MOUSE	NOM_EVT_MALF+1	Check Mouse Device
NOM_OBJ_HIF_TOUCH	NOM_EVT_MALF+1	Check Touch Input
NOM_OBJ_HIF_SPEEDPOINT	NOM_EVT_MALF+1	Check SpeedPoint
NOM_OBJ_HIF_ALARMBOX	NOM_EVT_MALF+1	Rem.AlarmDev.Malf.
NOM_OBJ_QUICKLINK	NOM_EVT_ADVIS_PWR_HI+1	MSL Power High
NOM_OBJ_QUICKLINK	NOM_EVT_ADVIS_PWR_OFF+1	MSL Power Off
NOM_OBJ_QUICKLINK	NOM_EVT_ADVIS_PWR_OVER+1	MSL Power Overload
NOM_OBJ_BUS_I2C	NOM_EVT_MALF+1	Internal.Comm.Malf
NOM_MOC_VMS_MDS	NOM_EVT_VOLTAGE_OUT_OF_RANGE+1	CheckInternVoltage
NOM_OBJ_QUICKLINK	NOM_EVT_VOLTAGE_OUT_OF_RANGE+1	Check MSL Voltage
NOM_MOC_VMS_MDS	NOM_EVT_TEMP_HI_GT_LIM+1	Check Monitor Temp
NOM_OBJ_SETTING	NOM_EVT_MALF+1	Check Settings
NOM_OBJ_SETTING	NOM_EVT_MALF+1	Settings Malfunc.
NOM_OBJ_CPU_SEC	NOM_EVT_MALF+1	Check Main Board 2
NOM_OBJ_SETTING	NOM_EVT_MALF+1	Check Flex Texts
NOM_OBJ_LED	NOM_EVT_MALF+1	Check Alarm Lamps
NOM_OBJ_NETWORK	NOM_EVT_MALF+1	Check Network Conf
NOM_OBJ_SETTING	NOM_EVT_MALF+1	Check Screen Res.
NOM_OBJ_SETTING	NOM_EVT_MALF+1	Check Waves
NOM_OBJ_DISP_SEC	NOM_EVT_UNSUPPORTED+1	Indep.Dsp NotSupp.
NOM_OBJ_DISP_SEC	NOM_EVT_MALF+1	Indep.Dsp Malfunc.
NOM_OBJ_DISP_THIRD	NOM_EVT_FAIL+1	Chk IndepDsp Cable
NOM_OBJ_DISP_THIRD	NOM_EVT_REVERSED+1	MCC Reversed
NOM_OBJ_DISP_THIRD	NOM_EVT_CONFIG_ERR+1	Check MCC
NOM_OBJ_DISP_THIRD	NOM_EVT_UNSUPPORTED+1	Intell.Dsp Unsupp.
NOM_OBJ_DISP_THIRD	NOM_EVT_MALF+1	Intell.Dsp Malf.
NOM_OBJ_DISP_THIRD	NOM_EVT_DISCONN+1	Intell.Dsp Missing
NOM_OBJ_DISP_THIRD	NOM_EVT_UNAVAIL+1	MCC Unsupported
NOM_OBJ_SETTING	NOM_EVT_SYNCH_UNSUPPORTED+1	Tele Sync Unsupp.
NOM_OBJ_SETTING	NOM_EVT_SYNCH_ERR+1	Check TeleSettings
NOM_OBJ_NETWORK	NOM_EVT_ECG_ADVIS_SRC_CHK+1	!!CHECK ECG SOURCE
NOM_OBJ_NETWORK	NOM_EVT_ECG_ADVIS_SRC_CHK+1	!!CHECK ECG SOURCE
NOM_OBJ_NETWORK	NOM_EVT_DEV_ASSOC_CHK+1	!! CHECK PAIRING

Alert Source	Alert Code	Alert Text
NOM_OBJ_NETWORK	NOM_EVT_DEV_ASSOC_CHK+1	!!!CHECK PAIRING
NOM_OBJ_XMTR	NOM_EVT_UNPLUGGED+1	TELE DISCONNECTED
NOM_ECG_ELEC_POTL	NOM_EVT_TELE_EQUIP_MALF	ECG EQUIP MALF T
NOM_OBJ_XMTR	NOM_EVT_EQUIP_MALF+1	TELE EQUIP MALF
NOM_OBJ_XMTR	NOM_EVT_UNSUPPORTED+1	TELE UNSUPPORTED
NOM_OBJ_SETTING	NOM_EVT_SYNCH_ERR_ECG+1	Check ECG Settings
NOM_OBJ_SETTING	NOM_EVT_SYNCH_ERR_SPO2T+1	Chk SpO2T Settings
NOM_DEV_CALC_VMD	NOM_EVT_ADVIS_SETTINGS_CHK+1	Check DrugSettings
Used by a specific measurement.	NOM_EVT_AGENT_MEAS_MALF+1	XXXXXX MEAS FAILED
NOM_OBJ_MMS_EXT	NOM_EVT_UNPLUGGED+1	MMS Ext. UNPLUGGED
NOM_OBJ_MMS_EXT	NOM_EVT_ADVIS_PWR_OFF + 1	MMS Ext. Unpowered
NOM_OBJ_MMS_EXT	NOM_EVT_MALF + 1	MMS Ext. MALF
NOM_OBJ_MMS_EXT	NOM_EVT_UNSUPPORTED + 1	MMS Ext. Unsupported
NOM_OBJ_ECG_SYNC	NOM_EVT_ADVIS_CABLE_CHK + 1	MMS Ext. Unsupported
NOM_OBJ_ECG_SYNC	NOM_EVT_ADVIS_CABLE_CHK + 1	CHK ECG Sync Cable
NOM_OBJ_MMS	NOM_EVT_ADVIS_DEACT + 1	MSMT DEACTIVATED
NOM_OBJ_TELEMON	NOM_EVT_ALARM_MORE_TECH +1	MORE BED ALARMS
NOM_OBJ_XMTR	NOM_EVT_INCOMPAT + 1	TELE CONFIG UNSUPP
NOM_OBJ_QUICKLINK	NOM_EVT_SYNCH_ERR+1	Chk MSL Connection
NOM_OBJ_TELEMON	NOM_EVT_ALARM_MORE_TECH_YELLOW +1	!!MORE BED ALARMS
NOM_OBJ_TELEMON	NOM_EVT_ALARM_MORE_TECH_RED+1	!!!MORE BED ALARMS

AlarmMgr

Alert Source	Alert Code	Alert Text
source id of the parameter	NOM_EVT_STAT_DISCONN	XXXXXX UNPLUGGED
source id of the parameter	NOM_EVT_ADVIS_DEACT	XXXXXX DEACTIVATED

NBP

Alert Source	Alert Code	Alert Text
NOM_PRESS_BLD_NONINV	NOM_EVT_CUFF_NOT_DEFLATED	CUFF NOT DEFLATED
NOM_PRESS_BLD_NONINV	NOM_EVT_CUFF_NOT_DEFLATED_YELLOW	!! CUFF NOT DEFLAT
NOM_PRESS_BLD_NONINV	NOM_EVT_CUFF_NOT_DEFLATED_RED	!!!CUFF NOT DEFLAT
NOM_PRESS_BLD_NONINV	NOM_EVT_CUFF_INFLAT_OVER	NBP CUFF OVERPRESS
NOM_PRESS_BLD_NONINV	NOM_EVT_CUFF_INFLAT_OVER_YELLOW	!! CUFF OVERPRESS
NOM_PRESS_BLD_NONINV	NOM_EVT_CUFF_INFLAT_OVER_RED	!!!CUFF OVERPRESS
NOM_PRESS_BLD_NONINV	NOM_EVT_EQUIP_MALF	XXXXXX EQUIP MALF
NOM_PRESS_BLD_NONINV	NOM_EVT_MSMT_INTERRUP	NBP INTERRUPTED
NOM_PRESS_BLD_NONINV	NOM_EVT_MSMT_FAIL	NBP MEASURE FAILED
NOM_PRESS_BLD_NONINV_MEAN	NOM_EVT_HI	** XXXXXX HIGH
NOM_PRESS_BLD_NONINV_MEAN	NOM_EVT_LO	** XXXXXX LOW
NOM_PRESS_BLD_NONINV_SYS	NOM_EVT_HI	** XXXXXX HIGH
NOM_PRESS_BLD_NONINV_SYS	NOM_EVT_LO	** XXXXXX LOW
NOM_PRESS_BLD_NONINV_DIA	NOM_EVT_HI	** XXXXXX HIGH
NOM_PRESS_BLD_NONINV_DIA	NOM_EVT_LO	** XXXXXX LOW

TcGas

Alert Source	Alert Code	Alert Text
NOM_O2_TCUT	NOM_EVT_EQUIP_MALF	XXXXXX EQUIP MALF
NOM_CO2_TCUT	NOM_EVT_EQUIP_MALF	XXXXXX EQUIP MALF
NOM_GAS_TCUT	NOM_EVT_EQUIP_MALF	XXXXXX EQUIP MALF
NOM_O2_TCUT	NOM_EVT_SENSOR_DISCONN	XXXXXX NO TRANSDUC
NOM_CO2_TCUT	NOM_EVT_SENSOR_DISCONN	XXXXXX NO TRANSDUC
NOM_GAS_TCUT	NOM_EVT_SENSOR_DISCONN	XXXXXX NO TRANSDUC
NOM_O2_TCUT	NOM_EVT_STAT_CALIB_RUNNING	XXXXXX CAL RUNNING
NOM_CO2_TCUT	NOM_EVT_STAT_CALIB_RUNNING	XXXXXX CAL RUNNING
NOM_GAS_TCUT	NOM_EVT_STAT_CALIB_RUNNING	XXXXXX CAL RUNNING
NOM_O2_TCUT	NOM_EVT_CALIB_FAIL	XXXXXX CAL FAILED
NOM_CO2_TCUT	NOM_EVT_CALIB_FAIL	XXXXXX CAL FAILED
NOM_GAS_TCUT	NOM_EVT_CALIB_FAIL	XXXXXX CAL FAILED
NOM_O2_TCUT	NOM_EVT_ADVIS_CALIB_REQD	XXXXXX CAL REQUIRD
NOM_CO2_TCUT	NOM_EVT_ADVIS_CALIB_REQD	XXXXXX CAL REQUIRD
NOM_GAS_TCUT	NOM_EVT_ADVIS_CALIB_REQD	XXXXXX CAL REQUIRD
NOM_O2_TCUT	NOM_EVT_ADVIS_CHANGE_SITE	XXXXXX CHANGE SITE
NOM_CO2_TCUT	NOM_EVT_ADVIS_CHANGE_SITE	XXXXXX CHANGE SITE
NOM_GAS_TCUT	NOM_EVT_ADVIS_CHANGE_SITE	XXXXXX CHANGE SITE
NOM_O2_TCUT	NOM_EVT_STAT_SENSOR_WARMING	XXXXXX STABILIZING

Alert Source	Alert Code	Alert Text
NOM_CO2_TCUT	NOM_EVT_STAT_SENSOR_WARMING	XXXXXX STABILIZING
NOM_GAS_TCUT	NOM_EVT_STAT_SENSOR_WARMING	XXXXXX STABILIZING
NOM_O2_TCUT	NOM_EVT_ADVIS_CHECK_SITE_TIME	XXXXXX CHECK TIME
NOM_CO2_TCUT	NOM_EVT_ADVIS_CHECK_SITE_TIME	XXXXXX CHECK TIME
NOM_GAS_TCUT	NOM_EVT_ADVIS_CHECK_SITE_TIME	XXXXXX CHECK TIME
NOM_O2_TCUT	NOM_EVT_LO	** XXXXXX LOW
NOM_O2_TCUT	NOM_EVT_HI	** XXXXXX HIGH
NOM_CO2_TCUT	NOM_EVT_LO	** XXXXXX LOW
NOM_CO2_TCUT	NOM_EVT_HI	** XXXXXX HIGH

VueLink

Alert Source	Alert Code	Alert Text
NOM_DEV_SYS_MULTI_MODAL_MDS	NOM_EVT_EQUIP_MALF+1	XXXXXX EQUIP MALF
NOM_DEV_SYS_MULTI_MODAL_MDS	NOM_EVT_CONFIG_ERR+1	XXXXXX NO CONFIG
NOM_DEV_SYS_MULTI_MODAL_MDS	NOM_EVT_ADVIS_SETUP_CHK+1	XXXXXX CHECK SETUP
NOM_DEV_SYS_MULTI_MODAL_MDS	NOM_EVT_ADVIS_CONFIG_CHK+1	XXXXXX CHK CONF.
NOM_DEV_SYS_MULTI_MODAL_MDS	NOM_EVT_ADVIS_CABLE_CHK+1	XXXXXX CHK CABLE
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_1+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_2+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_3+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_4+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_5+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_6+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_7+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_8+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_9+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_10+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_11+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_12+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_13+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_14+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_15+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_16+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_17+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_18+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_19+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_20+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_21+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_22+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_23+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_24+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_25+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_26+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_27+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_28+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_29+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_30+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_31+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_32+1	
depends on configuration	NOM_EVT_EXT_DEV_AL_CODE_33+1	

Battery

Alert Source	Alert Code	Alert Text
NOM_MOC_BATT	NOM_EVT_BATT_PROB+1	BATTERIES MALFUNC.
NOM_MOC_BATT	NOM_EVT_BATT_PROB+1	XXXXXX MALFUNCTION.
NOM_OBJ_BATT_1	NOM_EVT_BATT_PROB+1	XXXXXX MALFUNCTION
NOM_OBJ_BATT_2	NOM_EVT_BATT_PROB+1	XXXXXX MALFUNCTION
NOM_MOC_BATT	NOM_EVT_EMPTY+1	BATTERIES EMPTY
NOM_MOC_BATT	NOM_EVT_EMPTY+1	XXXXXX EMPTY
NOM_OBJ_BATT_1	NOM_EVT_EMPTY+1	XXXXXX EMPTY
NOM_OBJ_BATT_2	NOM_EVT_EMPTY+1	XXXXXX EMPTY
NOM_OBJ_BATT_1	NOM_EVT_ABSENT+1	XXXXXX MISSING
NOM_OBJ_BATT_2	NOM_EVT_ABSENT+1	XXXXXX MISSING
NOM_MOC_BATT	NOM_EVT_BATT_LO+1	BATTERIES LOW
NOM_OBJ_BATT_1	NOM_EVT_EMPTY+1	XXXXXX LOW
NOM_OBJ_BATT_2	NOM_EVT_EMPTY+1	XXXXXX LOW
NOM_OBJ_BATT_CHARGER	NOM_EVT_MALF+1	CHARGER MALFUNC.
NOM_MOC_BATT	NOM_EVT_INCOMPAT+1	BATTERIES INCOMPAT
NOM_OBJ_BATT	NOM_EVT_INCOMPAT+1	XXXXXX INCOMPAT.
NOM_OBJ_BATT_1	NOM_EVT_INCOMPAT+1	XXXXXX INCOMPAT.
NOM_OBJ_BATT_2	NOM_EVT_INCOMPAT+1	XXXXXX INCOMPAT.
NOM_MOC_BATT	NOM_EVT_TEMP_HI_GT_LIM+1	CHECK BATT TEMP
NOM_MOC_BATT	NOM_EVT_STAT_BATT_CHARGING +1	Charge XXXXXX now
NOM_MOC_BATT_1	NOM_EVT_STAT_BATT_CHARGING +1	Charge XXXXXX now
NOM_MOC_BATT_2	NOM_EVT_STAT_BATT_CHARGING +1	Charge XXXXXX now
NOM_OBJ_BATT	NOM_EVT_ABSENT+1	!!INSERT BATTERY

Telemetry

Alert Source	Alert Code	Alert Text
NOM_ECG_ELEC_POTL	NOM_EVT_EQUIP_MALF	XXXXXX EQUIP MALF
NOM_ECG_ELEC_POTL	NOM_EVT_LEADS_OFF	XXXXXX LEADS OFF
NOM_ECG_LEAD_C	NOM_EVT_LEAD_OFF	XXX LEAD OFF
NOM_ECG_LEAD_RA	NOM_EVT_LEAD_OFF	XXX LEAD OFF
NOM_ECG_LEAD_LA	NOM_EVT_LEAD_OFF	XXX LEAD OFF
NOM_ECG_LEAD_LL	NOM_EVT_LEAD_OFF	XXX LEAD OFF
NOM_ECG_LEAD_RL	NOM_EVT_LEAD_OFF	XXX LEAD OFF
NOM_ECG_LEAD_C1FR	NOM_EVT_LEAD_OFF	XXX LEAD OFF
NOM_ECG_LEAD_C2FR	NOM_EVT_LEAD_OFF	XXX LEAD OFF
NOM_ECG_LEAD_C3FR	NOM_EVT_LEAD_OFF	XXX LEAD OFF

Alert Source	Alert Code	Alert Text
NOM_ECG_LEAD_C4FR	NOM_EVT_LEAD_OFF	XXX LEAD OFF
NOM_ECG_LEAD_C5FR	NOM_EVT_LEAD_OFF	XXX LEAD OFF
NOM_ECG_LEAD_C6FR	NOM_EVT_LEAD_OFF	XXX LEAD OFF
NOM_ECG_LEAD_AS	NOM_EVT_LEAD_OFF	XXX LEAD OFF
NOM_ECG_LEAD_AI	NOM_EVT_LEAD_OFF	XXX LEAD OFF
NOM_ECG_LEAD_ES	NOM_EVT_LEAD_OFF	XXX LEAD OFF
NOM_ECG_ELEC_POTL	NOM_EVT_ADVIS_LEAD_CHK	INVALID LEADSET
NOM_PULS_OXIM_SAT_O2_TELE	NOM_EVT_EQUIP_MALF	XXXXXX EQUIP MALF
NOM_PULS_OXIM_SAT_O2_TELE	NOM_EVT_SENSOR_MALF	XXXXXX SENSOR MALF
NOM_PULS_OXIM_SAT_O2_TELE	NOM_EVT_SENSOR_MALF	XXXXXX SENSOR MALF
NOM_PULS_OXIM_SAT_O2_TELE	NOM_EVT_MSMT_INTERF_ERR	XXXXXX INTERFERNCE
NOM_PULS_OXIM_SAT_O2_TELE	NOM_EVT_SIG_NOISY	XXXXXX NOISY SIGN.
NOM_PULS_OXIM_SAT_O2_TELE	NOM_EVT_NON_PULSATILE	XXXXXX NON-PULSAT.
NOM_PULS_OXIM_SAT_O2_TELE	NOM_EVT_ERRATIC	XXXXXX ERRATIC
NOM_PULS_OXIM_SAT_O2_TELE	NOM_EVT_SIG_LO	XXXXXX LOW PERF
NOM_PULS_OXIM_SAT_O2_TELE	NOM_EVT_SUST	XXXXXX EXTD.UPDATE
NOM_PRESS_BLD_NONINV_TELE	NOM_EVT_CUFF_NOT_DEFLATED	CUFF NOT DEFLATED
NOM_PRESS_BLD_NONINV_TELE	NOM_EVT_CUFF_INFLAT_OVER	NBP CUFF OVERPRESS
NOM_PRESS_BLD_NONINV_TELE	NOM_EVT_MSMT_INTERRUP	NBP INTERRUPTED
NOM_PRESS_BLD_NONINV_TELE	NOM_EVT_MSMT_FAIL	NBP MEASURE FAILED
NOM_PRESS_BLD_NONINV_TELE	NOM_EVT_EQUIP_MALF	XXXXXX EQUIP MALF
NOM_OBJ_TELEMON	NOM_EVT_ADVIS_BATT_COND+1	CHARGE MON BATT
NOM_MOC_BATT	NOM_EVT_BATT_LO+1	BATTERY LOW
NOM_OBJ_XMTR	NOM_EVT_EQUIP_MALF+1	TRANSMITTER MALF
NOM_OBJ_XMTR	NOM_EVT_MSMT_INTERRUP+1	TRANSMITTER OFF
NOM_OBJ_XMTR	NOM_EVT_STAT_STANDBY+1	TELEMETRY STANDBY
NOM_OBJ_XMTR	NOM_EVT_ABSENT+1	NO SIGNAL
NOM_OBJ_XMTR	NOM_EVT_ADVIS_NURSE_CALL+1	* NURSE CALL
NOM_OBJ_XMTR	NOM_EVT_WEAK+1	XXXXXX WEAK SIGNAL
NOM_OBJ_XMTR	NOM_EVT_MSMT_INTERF_ERR+1	XXXXXX INTERFERNCE
NOM_OBJ_TELEMETRY_XMTR	NOM_EVT_ALARM_MED_YELLOW_SHORT+1	* TELE ALARM
NOM_OBJ_TELEMETRY_XMTR	NOM_EVT_ALARM_MED_YELLOW+1	** TELE ALARM

Alert Source	Alert Code	Alert Text
NOM_OBJ_TELEMETRY_XMTR	NOM_EVT_ALARM_MED_RED+1	*** TELE ALARM
NOM_OBJ_TELEMETRY_XMTR	NOM_EVT_ALARM_TECH+1	TELE INOP
NOM_OBJ_TELEMETRY_XMTR	NOM_EVT_ALARM_TECH_YELLOW+1	!! TELE INOP
NOM_OBJ_TELEMETRY_XMTR	NOM_EVT_ALARM_TECH_RED+1	!!! TELE INOP
NOM_OBJ_TELEMETRY_XMTR	NOM_EVT_OUT_OF_AREA+1	OUT OF AREA
NOM_ECG_ELEC_POTL	NOM_EVT_LEADS_DISCONN	LEADSET UNPLUGGED
NOM_ECG_ELEC_POTL	NOM_EVT_SRC_ABSENT	NO ECG SOURCE
NOM_OBJ_BATT_TELE	NOM_EVT_BATT_LO+1	BATTERY LOW T
NOM_OBJ_BATT_TELE	NOM_EVT_ADVIS_BATT_REPLACE+1	REPLACE BATTERY T
NOM_ECG_ELEC_POTL	NOM_EVT_LEAD_DISCONN_YELLOW	!! ECG LEADS OFF
NOM_ECG_ELEC_POTL	NOM_EVT_LEAD_DISCONN_RED	!!!ECG LEADS OFF

Spirometry

Alert Source	Alert Code	Alert Text
Spiro	NOM_EVT_MSMT_RANGE_OVER+1	XXXXXX OVERRANGE
Spiro	NOM_EVT_MSMT_UNSUPPORTED+1	XXXXXX UNSUPPORTED
Spiro	NOM_EVT_MALF+1	SPIRO MALFUNCTION
Spiro	NOM_EVT_STAT_FW_UPDATE_IN_PROGRESS+1	SPIRO UPGRADE
Spiro	NOM_EVT_INCOMPAT+1	SPIRO INCOMPATIBLE
Spiro	NOM_EVT_MSMT_INOP+1	XXXXXX CANNOT MEAS
Spiro	NOM_EVT_SENSOR_DISCONN+1	SPIRO NO SENSOR
Spiro	NOM_EVT_SENSOR_PROB+1	SPIRO PATIENT CAT.
Spiro	NOM_EVT_STAT_CALIB_RUNNING+1	SPIRO PURGING
Spiro	NOM_EVT_CALIB_FAIL+1	SPIRO PURGE FAILED
Spiro	NOM_EVT_ADVIS_GAS_AGENT_CHK+1	SPIRO GAS COMPENS?
Spiro	NOM_EVT_ADVIS_SENSOR_CHK+1	SPIRO PATIENT CAT.
Spiro	NOM_EVT_STAT_AL_OFF+1	SPIRO ALARMS SUPPR
Spiro	NOM_EVT_BREATH_ABSENT+1	SPIRO NO BREATH

Alert Source	Alert Code	Alert Text
NOM_AWAY_RESP_RATE_SPIRO	NOM_EVT_MSMT_INOP	XXXXXX CANNOT MEAS
NOM_COMPL_LUNG	NOM_EVT_MSMT_INOP	XXXXXX CANNOT MEAS
NOM_RES_AWAY	NOM_EVT_MSMT_INOP	XXXXXX CANNOT MEAS
NOM_VENT_PRESS_AWAY_END_EXP_POS	NOM_EVT_MSMT_INOP	XXXXXX CANNOT MEAS
NOM_PRESS_AWAY_INSP	NOM_EVT_MSMT_INOP	XXXXXX CANNOT MEAS
NOM_VOL_AWAY_INSP_TIDAL	NOM_EVT_MSMT_INOP	XXXXXX CANNOT MEAS
NOM_VOL_AWAY_EXP_TIDAL	NOM_EVT_MSMT_INOP	XXXXXX CANNOT MEAS
NOM_VOL_MINUTE_AWAY_INSP	NOM_EVT_MSMT_INOP	XXXXXX CANNOT MEAS
NOM_VOL_MINUTE_AWAY_EXP	NOM_EVT_MSMT_INOP	XXXXXX CANNOT MEAS

Predictive Temp

Alert Source	Alert Code	Alert Text
current label e.g. NOM_TEMP_ORAL_PRED	NOM_EVT_MSMT_FAIL	XXXXXX MEAS FAILED
current label e.g. NOM_TEMP_ORAL_PRED	NOM_EVT_MALF	XXXXXX EQUIP MALF
current label e.g. NOM_TEMP_ORAL_PRED	NOM_EVT_INCOMPAT	XXXXXX INCOMPAT.
current label e.g. NOM_TEMP_ORAL_PRED	NOM_EVT_SENSOR_PROB	XXXXXX CHECK PROBE
current label e.g. NOM_TEMP_ORAL_PRED	NOM_EVT_XDUCR_MALF	XXXXXX PROBE MALF
current label e.g. NOM_TEMP_ORAL_PRED	NOM_EVT_XDUCR_DISCONN	XXXXXX NO PROBE
current label e.g. NOM_TEMP_ORAL_PRED	NOM_EVT_MSMT_RANGE_OVER	XXXXXX OVERRANGE
current label e.g. NOM_TEMP_ORAL_PRED	NOM_EVT_UNPLUGGED+1	XXXXXX DEACTIVATED

Protocol Watch

Alert Source	Alert Code	Alert Text
NOM_DEV_PROT_WATCH_CHAN	NOM_EVT_ADVIS_SETTINGS_CHK+1	PW: Check Settings
NOM_DEV_PROT_WATCH_CHAN	NOM_EVT_ADVIS_ACTION_REQD+1	PW:Action Required

Alert Source	Alert Code	Alert Text
NOM_DEV_PROT_WATCH_CHAN	NOM_EVT_ADVIS_ACTION_REQD_YELLOW+1	!!PW:Action Requ'd
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_TECH+1	
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_TECH+1	
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_TECH_YELLOW+1	
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_TECH_YELLOW+1	
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_TECH_RED+1	
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_TECH_RED+1	
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_TECH+1	
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_TECH+1	
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_TECH_YELLOW+1	
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_TECH_YELLOW+1	
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_TECH_RED+1	
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_TECH_RED+1	
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_TECH+1	
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_TECH+1	
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_TECH_YELLOW+1	
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_TECH_YELLOW+1	
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_TECH_RED+1	
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_TECH_RED+1	
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_MED_YELLOW_SHORT+1	
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_MED_YELLOW+1	
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_MED_RED+1	
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_MED_YELLOW_SHORT+1	
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_MED_YELLOW+1	
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_MED_RED+1	
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_MED_YELLOW_SHORT+1	
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_MED_YELLOW+1	
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_MED_RED+1	
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_TECH+1	PW ALARM

Alert Source	Alert Code	Alert Text
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_TECH_YELLOW+1	PW ALARM
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_TECH_RED+1	PW ALARM
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_TECH+1	PW ALARM
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_TECH_YELLOW+1	PW ALARM
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_TECH_RED+1	PW ALARM
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_TECH+1	PW ALARM
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_TECH_YELLOW+1	PW ALARM
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_TECH_RED+1	PW ALARM
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_MED_YELLOW+1	** PW ALARM
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_MED_YELLOW_SHORT+1	** PW ALARM
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_MED_YELLOW+1	** PW ALARM
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_MED_YELLOW_SHORT+1	** PW ALARM
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_MED_YELLOW+1	** PW ALARM
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_MED_YELLOW_SHORT+1	** PW ALARM
NOM_OBJ_PROT_WATCH_1	NOM_EVT_ALARM_MED_RED+1	*** PW ALARM
NOM_OBJ_PROT_WATCH_2	NOM_EVT_ALARM_MED_RED+1	*** PW ALARM
NOM_OBJ_PROT_WATCH_3	NOM_EVT_ALARM_MED_RED+1	*** PW ALARM
NOM_DEV_PROT_WATCH_CHAN	NOM_DEV_PROT_WATCH_CHAN	PW: In Conflict

Metabolics

Alert Source	Alert Code	Alert Text
NOM_DEV_METAB_VMD	NOM_EVT_ADVIS_SRC_CHK+1	METAB CHK SOURCES
NOM_VENT_VOL_LUNG_ALV	NOM_EVT_STAT_AL_OFF	XXXXXX ALARM SUPPR
NOM_DEV_METAB_VMD	NOM_EVT_ADVIS_SRC_CHK+1	EXT VENT CHK SRC
NOM_RATIO_AWAY_DEADSP_TIDAL	NOM_EVT_ADVIS_SRC_CHK	Vd/Vt ENTER PaCO2
	NOM_EVT_MSMT_RANGE_OVER	XXXXXX OVERRANGE
NOM_ENERGY_BAL	NOM_EVT_MSMT_RANGE_OVER	XXXXXX OVERRANGE

Intellibridge

Alert Source	Alert Code	Alert Text
	NOM_EVT_ADVIS_SETUP_CHK+1	DEVICE CHECK SETUP
	NOM_EVT_ADVIS_CONFIG_CHK+1	DEVICE CHECK CONF.
	NOM_EVT_EQUIP_MALF+1	XXXXXX EQUIP MALF
	NOM_EVT_CONFIG_ERR+1	NO DEVICE DATA
	NOM_EVT_CONFIG_ERR+1	!!NO DEVICE DATA
	NOM_EVT_CONFIG_ERR+1	!!!NO DEVICE DATA
	NOM_EVT_UNPLUGGED+1	XXXXXX UNPLUGGED
	NOM_EVT_UNSUPPORTED+1	DEVICE UNSUPPORTED

Short Range Radio

Alert Source	Alert Code	Alert Text
NOM_OBJ_SRR_IF_1	NOM_EVT_SRR_INTERF+1	SRR INTERFERENCE
NOM_OBJ_SRR_IF_1	NOM_EVT_SRR_INVALID_CHAN+1	SRR INVALID CHAN
NOM_OBJ_SRR_IF_1	NOM_EVT_MALF+1	SRR MALFUNCTION

The following code from the SCADA partition are used for the alert source:

NOM_ECG_LEAD_I	1
NOM_ECG_LEAD_II	2
NOM_ECG_LEAD_LA	21
NOM_ECG_LEAD_RA	22
NOM_ECG_LEAD_LL	23
NOM_ECG_LEAD_fI	24
NOM_ECG_LEAD_fE	25
NOM_ECG_LEAD_fA	27
NOM_ECG_LEAD_C	66
NOM_ECG_LEAD_C1FR	82
NOM_ECG_LEAD_C2FR	83
NOM_ECG_LEAD_C3FR	84
NOM_ECG_LEAD_C4FR	85
NOM_ECG_LEAD_C5FR	87
NOM_ECG_LEAD_C6FR	88
NOM_ECG_LEAD_C7FR	89
NOM_ECG_LEAD_C8FR	90
NOM_ECG_LEAD_ES	100
NOM_ECG_LEAD_AS	101
NOM_ECG_LEAD_AI	102
NOM_ECG_LEAD_RL	115
NOM_ECG_LEAD_EASI_S	116
NOM_ECG_ELEC_POTL	256
NOM_ECG_ELEC_POTL_I	257
NOM_ECG_ELEC_POTL_II	258
NOM_ECG_ELEC_POTL_V1	259
NOM_ECG_ELEC_POTL_V2	260
NOM_ECG_ELEC_POTL_V3	261
NOM_ECG_ELEC_POTL_V4	262
NOM_ECG_ELEC_POTL_V5	263
NOM_ECG_ELEC_POTL_V6	264
NOM_ECG_ELEC_POTL_III	317
NOM_ECG_ELEC_POTL_AVR	318
NOM_ECG_ELEC_POTL_AVL	319
NOM_ECG_ELEC_POTL_AVF	320
NOM_ECG_ELEC_POTL_V	323
NOM_ECG_ELEC_POTL_MCL	331
NOM_ECG_ELEC_POTL_MCL1	332
NOM_ECG_AMPL_ST	768
NOM_ECG_AMPL_ST_I	769
NOM_ECG_AMPL_ST_II	770
NOM_ECG_AMPL_ST_V1	771
NOM_ECG_AMPL_ST_V2	772
NOM_ECG_AMPL_ST_V3	773
NOM_ECG_AMPL_ST_V4	774
NOM_ECG_AMPL_ST_V5	775
NOM_ECG_AMPL_ST_V6	776
NOM_ECG_AMPL_ST_III	829
NOM_ECG_AMPL_ST_AVR	830
NOM_ECG_AMPL_ST_AVL	831
NOM_ECG_AMPL_ST_AVF	832
NOM_ECG_AMPL_ST_V	835
NOM_ECG_AMPL_ST_MCL	843
NOM_ECG_AMPL_ST_ES	868
NOM_ECG_AMPL_ST_AS	869

NOM_ECG_AMPL_ST_AI	870
NOM_ECG_TIME_PD_QT_GL	16160
NOM_ECG_TIME_PD_QTc	16164
NOM_ECG_CARD_BEAT_RATE	16770
NOM_ECG_CARD_BEAT_RATE_BTB	16778
NOM_ECG_V_P_C_CNT	16993
NOM_ECG_V_P_C_RATE	16994
NOM_ECG_V_P_C_FREQ	17000
NOM_PULS_RATE	18442
NOM_PLETH_PULS_RATE	18466
NOM_RES_VASC_SYS_INDEX	18688
NOM_WK_LV_STROKE_INDEX	18692
NOM_WK_RV_STROKE_INDEX	18696
NOM_OUTPUT_CARD_INDEX	18700
NOM_PRESS_BLD	18944
NOM_PRESS_BLD_SYS	18945
NOM_PRESS_BLD_DIA	18946
NOM_PRESS_BLD_MEAN	18947
NOM_PRESS_BLD_NONINV	18948
NOM_PRESS_BLD_NONINV_SYS	18949
NOM_PRESS_BLD_NONINV_DIA	18950
NOM_PRESS_BLD_NONINV_MEAN	18951
NOM_PRESS_BLD_AORT	18956
NOM_PRESS_BLD_AORT_SYS	18957
NOM_PRESS_BLD_AORT_DIA	18958
NOM_PRESS_BLD_AORT_MEAN	18959
NOM_PRESS_BLD_ART	18960
NOM_PRESS_BLD_ART_SYS	18961
NOM_PRESS_BLD_ART_DIA	18962
NOM_PRESS_BLD_ART_MEAN	18963
NOM_PRESS_BLD_ART_ABP	18964
NOM_PRESS_BLD_ART_ABP_SYS	18965
NOM_PRESS_BLD_ART_ABP_DIA	18966
NOM_PRESS_BLD_ART_ABP_MEAN	18967
NOM_PRESS_BLD_ART_PULM	18972
NOM_PRESS_BLD_ART_PULM_SYS	18973
NOM_PRESS_BLD_ART_PULM_DIA	18974
NOM_PRESS_BLD_ART_PULM_MEAN	18975
NOM_PRESS_BLD_ART_PULM_WEDGE	18980
NOM_PRESS_BLD_ART_UMB	18984
NOM_PRESS_BLD_ART_UMB_SYS	18985
NOM_PRESS_BLD_ART_UMB_DIA	18986
NOM_PRESS_BLD_ART_UMB_MEAN	18987
NOM_PRESS_BLD_ATR_LEFT	18992
NOM_PRESS_BLD_ATR_LEFT_SYS	18993
NOM_PRESS_BLD_ATR_LEFT_DIA	18994
NOM_PRESS_BLD_ATR_LEFT_MEAN	18995
NOM_PRESS_BLD_ATR_RIGHT	18996
NOM_PRESS_BLD_ATR_RIGHT_SYS	18997
NOM_PRESS_BLD_ATR_RIGHT_DIA	18998
NOM_PRESS_BLD_ATR_RIGHT_MEAN	18999
NOM_PRESS_BLD_VEN_CENT	19012
NOM_PRESS_BLD_VEN_CENT_SYS	19013
NOM_PRESS_BLD_VEN_CENT_DIA	19014
NOM_PRESS_BLD_VEN_CENT_MEAN	19015
NOM_PRESS_BLD_VEN_UMB	19016

NOM_PRESS_BLD_VEN_UMB_SYS	19017
NOM_PRESS_BLD_VEN_UMB_DIA	19018
NOM_PRESS_BLD_VEN_UMB_MEAN	19019
NOM_SAT_O2_CONSUMP	19200
NOM_OUTPUT_CARD	19204
NOM_RES_VASC_PULM	19236
NOM_RES_VASC_SYS	19240
NOM_SAT_O2	19244
NOM_SAT_O2_ART	19252
NOM_SAT_O2_VEN	19260
NOM_SAT_DIFF_O2_ART_ALV	19264
NOM_TEMP	19272
NOM_TEMP_ART	19280
NOM_TEMP_AWAY	19284
NOM_TEMP_CORE	19296
NOM_TEMP_ESOPH	19300
NOM_TEMP_INJ	19304
NOM_TEMP_NASOPH	19308
NOM_TEMP_SKIN	19316
NOM_TEMP_TYMP	19320
NOM_TEMP_VEN	19324
NOM_VOL_BLD_STROKE	19332
NOM_WK_CARD_LEFT	19344
NOM_WK_CARD_RIGHT	19348
NOM_WK_LV_STROKE	19356
NOM_WK_RV_STROKE	19364
NOM_PULS_OXIM_PERF_REL	19376
NOM_PLETH	19380
NOM_PULS_OXIM_SAT_O2	19384
NOM_PULS_OXIM_SAT_O2_DIFF	19396
NOM_PULS_OXIM_SAT_O2_ART_LEFT	19400
NOM_PULS_OXIM_SAT_O2_ART_RIGHT	19404
NOM_OUTPUT_CARD_CTS	19420
NOM_VOL_VENT_L_END_SYS	19460
NOM_GRAD_PRESS_BLD_AORT_POS_MAX	19493
NOM_RESP	20480
NOM_RESP_RATE	20490
NOM_AWAY_RESP_RATE	20498
NOM_CAPAC_VITAL	20608
NOM_COMPL_LUNG	20616
NOM_COMPL_LUNG_DYN	20620
NOM_COMPL_LUNG_STATIC	20624
NOM_CONC_AWAY_CO2	20628
NOM_CONC_AWAY_CO2_ET	20636
NOM_CONC_AWAY_CO2_INSP_MIN	20646
NOM_AWAY_CO2	20652
NOM_AWAY_CO2_ET	20656
NOM_AWAY_CO2_INSP_MIN	20666
NOM_CO2_TCUT	20684
NOM_O2_TCUT	20688
NOM_FLOW_AWAY	20692
NOM_FLOW_AWAY_EXP_MAX	20697
NOM_FLOW_AWAY_INSP_MAX	20701
NOM_FLOW_CO2_PROD_RESP	20704
NOM_IMPED_TTHOR	20708
NOM_PRESS_RESP_PLAT	20712

NOM_PRESS_AWAY	20720
NOM_PRESS_AWAY_MIN	20722
NOM_PRESS_AWAY_CTS_POS	20724
NOM_PRESS_AWAY_NEG_MAX	20729
NOM_PRESS_AWAY_END_EXP_POS_INTRINSIC	20736
NOM_PRESS_AWAY_INSP	20744
NOM_PRESS_AWAY_INSP_MAX	20745
NOM_PRESS_AWAY_INSP_MEAN	20747
NOM_RATIO_IE	20760
NOM_RATIO_AWAY_DEADSP_TIDAL	20764
NOM_RES_AWAY	20768
NOM_RES_AWAY_EXP	20772
NOM_RES_AWAY_INSP	20776
NOM_TIME_PD_APNEA	20784
NOM_VOL_AWAY_TIDAL	20796
NOM_VOL_MINUTE_AWAY	20808
NOM_VOL_MINUTE_AWAY_EXP	20812
NOM_VOL_MINUTE_AWAY_INSP	20816
NOM_CONC_AWAY_O2	20836
NOM_VENT_CONC_AWAY_O2_DELTA	20840
NOM_VENT_CONC_AWAY_O2_EXP	20844
NOM_VENT_AWAY_CO2_EXP	20860
NOM_VENT_PRESS_AWAY_END_EXP_POS	20904
NOM_VENT_VOL_AWAY_DEADSP	20912
NOM_VENT_VOL_LUNG_TRAPD	20920
NOM_VENT_CONC_AWAY_O2_INSP	29848
NOM_VENT_FLOW_RATIO_PERF_ALV_INDEX	20880
NOM_VENT_FLOW_INSP	20876
NOM_VENT_CONC_AWAY_CO2_INSP	20832
NOM_VENT_PRESS_OCCL	20892
NOM_VENT_VOL_AWAY_DEADSP_REL	20916
NOM_VENT_VOL_MINUTE_AWAY_MAND	20940
NOM_COEF_GAS_TRAN	20948
NOM_CONC_AWAY_DESFL	20952
NOM_CONC_AWAY_ENFL	20956
NOM_CONC_AWAY_HALOTH	20960
NOM_CONC_AWAY_SEVOFL	20964
NOM_CONC_AWAY_ISOFL	20968
NOM_CONC_AWAY_N2O	20976
NOM_CONC_AWAY_DESFL_ET	21012
NOM_CONC_AWAY_ENFL_ET	21016
NOM_CONC_AWAY_HALOTH_ET	21020
NOM_CONC_AWAY_SEVOFL_ET	21024
NOM_CONC_AWAY_ISOFL_ET	21028
NOM_CONC_AWAY_N2O_ET	21036
NOM_CONC_AWAY_DESFL_INSP	21096
NOM_CONC_AWAY_ENFL_INSP	21100
NOM_CONC_AWAY_HALOTH_INSP	21104
NOM_CONC_AWAY_SEVOFL_INSP	21108
NOM_CONC_AWAY_ISOFL_INSP	21112
NOM_CONC_AWAY_N2O_INSP	21120
NOM_CONC_AWAY_O2_INSP	21124
NOM_VENT_TIME_PD_PPV	21344
NOM_VENT_PRESS_RESP_PLAT	21352
NOM_VENT_VOL_LEAK	21360
NOM_VENT_VOL_LUNG_ALV	21364

NOM_CONC_AWAY_O2_ET	21368
NOM_CONC_AWAY_N2	21372
NOM_CONC_AWAY_N2_ET	21376
NOM_CONC_AWAY_N2_INSP	21380
NOM_CONC_AWAY_AGENT	21384
NOM_CONC_AWAY_AGENT_ET	21388
NOM_CONC_AWAY_AGENT_INSP	21392
NOM_PRESS_CEREB_PERF	22532
NOM_PRESS_INTRA_CRAN	22536
NOM_PRESS_INTRA_CRAN_SYS	22537
NOM_PRESS_INTRA_CRAN_DIA	22538
NOM_PRESS_INTRA_CRAN_MEAN	22539
NOM_SCORE_GLAS_COMA	22656
NOM_SCORE_EYE_SUBSC_GLAS_COMA	22658
NOM_SCORE_MOTOR_SUBSC_GLAS_COMA	22659
NOM_SCORE_SUBSC_VERBAL_GLAS_COMA	22660
NOM_CIRCUM_HEAD	22784
NOM_TIME_PD_PUPIL_REACT_LEFT	22820
NOM_TIME_PD_PUPIL_REACT_RIGHT	22824
NOM_EEG_ELEC_POTL_CRTX	22828
NOM_EMG_ELEC_POTL_MUSCL	22844
NOM_EEG_FREQ_PWR_SPEC_CRTX_DOM_MEAN	22908
NOM_EEG_FREQ_PWR_SPEC_CRTX_PEAK	22916
NOM_EEG_FREQ_PWR_SPEC_CRTX_SPECTRAL_EDGE	22920
NOM_EEG_PWR_SPEC_TOT	22968
NOM_EEG_PWR_SPEC_ALPHA_REL	22996
NOM_EEG_PWR_SPEC_BETA_REL	23000
NOM_EEG_PWR_SPEC_DELTA_REL	23004
NOM_EEG_PWR_SPEC_THETA_REL	23008
NOM_FLOW_URINE_INSTANT	26636
NOM_VOL_URINE_BAL_PD	26660
NOM_VOL_URINE_COL	26672
NOM_VOL_INFUS_ACTUAL_TOTAL	26876
NOM_CONC_PH_ART	28676
NOM_CONC_PCO2_ART	28680
NOM_CONC_PO2_ART	28684
NOM_CONC_HB_ART	28692
NOM_CONC_HB_O2_ART	28696
NOM_CONC_PO2_VEN	28732
NOM_CONC_PH_VEN	28724
NOM_CONC_PCO2_VEN	28728
NOM_CONC_HB_O2_VEN	28744
NOM_CONC_PH_URINE	28772
NOM_CONC_NA_URINE	28780
NOM_CONC_NA_SERUM	28888
NOM_CONC_PH_GEN	28932
NOM_CONC_HCO3_GEN	28936
NOM_CONC_NA_GEN	28940
NOM_CONC_K_GEN	28944
NOM_CONC_GLU_GEN	28948
NOM_CONC_CA_GEN	28952
NOM_CONC_PCO2_GEN	28992
NOM_CONC_CHLORIDE_GEN	29032
NOM_BASE_EXCESS_BLD_ART	29036
NOM_CONC_PO2_GEN	29044

NOM_CONC_HCT_GEN	29060
NOM_VENT_MODE_MAND_INTERMIT	53290
NOM_TEMP_RECT	57348
NOM_TEMP_BLD	57364
NOM_TEMP_DIFF	57368
NOM_METRIC_NOS	61439
NOM_ECG_AMPL_ST_INDEX	61501
NOM_TIME_TCUT_SENSOR	61502
NOM_TEMP_TCUT_SENSOR	61503
NOM_VOL_BLD_INTRA_THOR	61504
NOM_VOL_BLD_INTRA_THOR_INDEX	61505
NOM_VOL_LUNG_WATER_EXTRA_VASC	61506
NOM_VOL_LUNG_WATER_EXTRA_VASC_INDEX	61507
NOM_VOL_GLOBAL_END_DIA	61508
NOM_VOL_GLOBAL_END_DIA_INDEX	61509
NOM_CARD_FUNC_INDEX	61510
NOM_OUTPUT_CARD_INDEX_CTS	61511
NOM_VOL_BLD_STROKE_INDEX	61512
NOM_VOL_BLD_STROKE_VAR	61513
NOM_EEG_RATIO_SUPPRN	61514
NOM_ELECTRODE_IMPED	61515
NOM_EEG_BIS_SIG_QUAL_INDEX	61517
NOM_EEG_BISPECTRAL_INDEX	61518
NOM_GAS_TCUT	61521
NOM_CONC_AWAY_SUM_MAC	61533
NOM_CONC_AWAY_SUM_MAC_ET	61534
NOM_CONC_AWAY_SUM_MAC_INSP	61535
NOM_RES_VASC_PULM_INDEX	61543
NOM_WK_CARD_LEFT_INDEX	61544
NOM_WK_CARD_RIGHT_INDEX	61545
NOM_SAT_O2_CONSUMP_INDEX	61546
NOM_PRESS_AIR_AMBIENT	61547
NOM_SAT_DIFF_O2_ART_VEN	61548
NOM_SAT_O2_DELIVER	61549
NOM_SAT_O2_DELIVER_INDEX	61550
NOM_RATIO_SAT_O2_CONSUMP_DELIVER	61551
NOM_RATIO_ART_VEN_SHUNT	61552
NOM_AREA_BODY_SURFACE	61553
NOM_INTENS_LIGHT	61554
NOM_HEATING_PWR_TCUT_SENSOR	61558
NOM_RATE_DIFF_CARD_BEAT_PULSE	61560
NOM_VOL_INJ	61561
NOM_VOL_THERMO_EXTRA_VASC_INDEX	61562
NOM_NUM_CATHETER_CONST	61564
NOM_PULS_OXIM_PERF_REL_LEFT	61578
NOM_PULS_OXIM_PERF_REL_RIGHT	61579
NOM_PULS_OXIM_PLETH_RIGHT	61580
NOM_PULS_OXIM_PLETH_LEFT	61581
NOM_CONC_BLD_UREA_NITROGEN	61583
NOM_CONC_BASE_EXCESS_ECF	61584
NOM_VENT_VOL_MINUTE_AWAY_SPONT	61585
NOM_CONC_DIFF_HB_O2_ATR_VEN	61586
NOM_PAT_WEIGHT	61587
NOM_PAT_HEIGHT	61588
NOM_CONC_AWAY_MAC	61593
NOM_PULS_OXIM_PLETH_TELE	61595

NOM_PULS_OXIM_SAT_O2_TELE	61596
NOM_PULS_OXIM_PULS_RATE_TELE	61597
NOM_PRESS_BLD_NONINV_TELE	61600
NOM_PRESS_BLD_NONINV_TELE_SYS	61601
NOM_PRESS_BLD_NONINV_TELE_DIA	61602
NOM_PRESS_BLD_NONINV_TELE_MEAN	61603
NOM_PRESS_GEN_1	61604
NOM_PRESS_GEN_1_SYS	61605
NOM_PRESS_GEN_1_DIA	61606
NOM_PRESS_GEN_1_MEAN	61607
NOM_PRESS_GEN_2	61608
NOM_PRESS_GEN_2_SYS	61609
NOM_PRESS_GEN_2_DIA	61610
NOM_PRESS_GEN_2_MEAN	61611
NOM_PRESS_GEN_3	61612
NOM_PRESS_GEN_3_SYS	61613
NOM_PRESS_GEN_3_DIA	61614
NOM_PRESS_GEN_3_MEAN	61615
NOM_PRESS_GEN_4	61616
NOM_PRESS_GEN_4_SYS	61617
NOM_PRESS_GEN_4_DIA	61618
NOM_PRESS_GEN_4_MEAN	61619
NOM_PRESS_INTRA_CRAN_1	61620
NOM_PRESS_INTRA_CRAN_1_SYS	61621
NOM_PRESS_INTRA_CRAN_1_DIA	61622
NOM_PRESS_INTRA_CRAN_1_MEAN	61623
NOM_PRESS_INTRA_CRAN_2	61624
NOM_PRESS_INTRA_CRAN_2_SYS	61625
NOM_PRESS_INTRA_CRAN_2_DIA	61626
NOM_PRESS_INTRA_CRAN_2_MEAN	61627
NOM_PRESS_BLD_ART_FEMORAL	61628
NOM_PRESS_BLD_ART_FEMORAL_SYS	61629
NOM_PRESS_BLD_ART_FEMORAL_DIA	61630
NOM_PRESS_BLD_ART_FEMORAL_MEAN	61631
NOM_PRESS_BLD_ART_BRACHIAL	61632
NOM_PRESS_BLD_ART_BRACHIAL_SYS	61633
NOM_PRESS_BLD_ART_BRACHIAL_DIA	61634
NOM_PRESS_BLD_ART_BRACHIAL_MEAN	61635
NOM_TEMP_VESICAL	61636
NOM_TEMP_CEREBRAL	61637
NOM_TEMP_AMBIENT	61638
NOM_TEMP_GEN_1	61639
NOM_TEMP_GEN_2	61640
NOM_TEMP_GEN_3	61641
NOM_TEMP_GEN_4	61642
NOM_USOUND_CARD_BEAT_RATE_FETAL	61643
NOM_USOUND_CARD_BEAT_RATE_FETAL_BTB	61644
NOM_USOUND_CARD_BEAT_FETAL_SIG_QUAL_INDEX	61645
NOM_ECG_CARD_BEAT_FETAL	61646
NOM_ECG_CARD_BEAT_RATE_FETAL	61647
NOM_ECG_CARD_BEAT_RATE_FETAL_BTB	61648
NOM_ECG_CARD_BEAT_FETAL_SIG_QUAL_INDEX	61649
NOM_TRIG_BEAT_FETAL	61650
NOM_ECG_ELEC_POTL_FETAL	61651
NOM_TOCO	61652

NOM_STAT_COINCIDENCE	61653
NOM_PRESS_INTRA_UTERAL	61656
NOM_VOL_AWAY	61663
NOM_VOL_AWAY_INSP_TIDAL	61664
NOM_VOL_AWAY_EXP_TIDAL	61665
NOM_AWAY_RESP_RATE_SPIRO	61666
NOM_PULS_PRESS_VAR	61667
NOM_PRESS_BLD_NONINV_PULS_RATE	61669
NOM_RATIO_FETAL_MVMT_TOTAL	61680
NOM_VENT_RESP_RATE_MAND	61681
NOM_VENT_VOL_TIDAL_MAND	61682
NOM_VENT_VOL_TIDAL_SPONT	61683
NOM_CARDIAC_TROPONIN_I	61684
NOM_CARDIO_PULMONARY_BYPASS_MODE	61685
NOM_BNP	61686
NOM_TIME_PD_RESP_PLAT	61695
NOM_SAT_O2_VEN_CENT	61696
NOM_SNR	61697
NOM_HUMID	61699
NOM_FRACT_EJECT	61701
NOM_PERM_VASC_PULM_INDEX	61702
NOM_TEMP_ORAL	61704
NOM_TEMP_AXIL	61708
NOM_TEMP_ORAL_PRED	61712
NOM_TEMP_RECT_PRED	61716
NOM_TEMP_AXIL_PRED	61720
NOM_TEMP_AIR_INCUB	61738
NOM_PULS_OXIM_PERF_REL_TELE	61740
NOM_TEMP_PRED	61760
NOM_SHUNT_RIGHT_LEFT	61770
NOM_ECG_TIME_PD_QT_HEART_RATE	61780
NOM_ECG_TIME_PD_QT_BASELINE	61781
NOM_ECG_TIME_PD_QTc_DELTA	61782
NOM_ECG_TIME_PD_QT_BASELINE_HEART_RATE	61783
NOM_CONC_PH_CAP	61784
NOM_CONC_PCO2_CAP	61785
NOM_CONC_PO2_CAP	61786
NOM_SAT_O2_CAP	61793
NOM_CONC_MG_ION	61787
NOM_CONC_MG_SER	61788
NOM_CONC_tCA_SER	61789
NOM_CONC_P_SER	61790
NOM_CONC_CHLOR_SER	61791
NOM_CONC_FE_GEN	61792
NOM_CONC_AN_GAP	61794
NOM_CONC_AN_GAP_CALC	61857
NOM_CONC_ALB_SER	61795
NOM_SAT_O2_ART_CALC	61796
NOM_SAT_O2_VEN_CALC	61798
NOM_SAT_O2_CAP_CALC	61856
NOM_CONC_HB_CO_GEN	29056
NOM_CONC_HB_FETAL	61797
NOM_CONC_HB_MET_GEN	29052
NOM_PLTS_CNT	61799
NOM_WB_CNT	61800
NOM_RB_CNT	61801

NOM_RET_CNT	61802
NOM_PLASMA_OSM	61803
NOM_CONC_CREA_CLR	61804
NOM_NSLOSS	61805
NOM_CONC_CHOLESTEROL	61806
NOM_CONC_TGL	61807
NOM_CONC_HDL	61808
NOM_CONC_LDL	61809
NOM_CONC_UREA_GEN	61810
NOM_CONC_CREA	61811
NOM_CONC_LACT	61812
NOM_CONC_BILI_TOT	61815
NOM_CONC_PROT_SER	61816
NOM_CONC_PROT_TOT	61817
NOM_CONC_BILI_DIRECT	61818
NOM_CONC_LDH	61819
NOM_ES_RATE	61820
NOM_CONC_PCT	61821
NOM_CONC_CREA_KIN_MM	61823
NOM_CONC_CREA_KIN_SER	61824
NOM_CONC_CREA_KIN_MB	61825
NOM_CONC_CHE	61826
NOM_CONC_CRP	61827
NOM_CONC_AST	61828
NOM_CONC_AP	61829
NOM_CONC_ALPHA_AMYLASE	61830
NOM_CONC_GPT	61831
NOM_CONC_GOT	61832
NOM_CONC_GGT	61833
NOM_TIME_PD_ACT	61834
NOM_TIME_PD_PT	61835
NOM_PT_INTL_NORM_RATIO	61836
NOM_TIME_PD_aPTT_WB	61837
NOM_TIME_PD_aPTT_PE	61838
NOM_TIME_PD_PT_WB	61839
NOM_TIME_PD_PT_PE	61840
NOM_TIME_PD_THROMBIN	61841
NOM_TIME_PD_COAGULATION	61842
NOM_TIME_PD_THROMBOPLAS	61843
NOM_FRACT_EXCR_NA	61844
NOM_CONC_UREA_URINE	61845
NOM_CONC_CREA_URINE	61846
NOM_CONC_K_URINE	61847
NOM_CONC_K_URINE_EXCR	61848
NOM_CONC_OSM_URINE	61849
NOM_CONC_GLU_URINE	61855
NOM_CONC_CHLOR_URINE	61850
NOM_CONC_PRO_URINE	61851
NOM_CONC_CA_URINE	61852
NOM_FLUID_DENS_URINE	61853
NOM_CONC_HB_URINE	61854
NOM_ENERGY_BAL	61861
NOM_PULS_OXIM_SAT_O2_PRE_DUCTAL	61888
NOM_PULS_OXIM_PERF_REL_PRE_DUCTAL	61996
NOM_PULS_OXIM_SAT_O2_POST_DUCTAL	61908
NOM_PULS_OXIM_PERF_REL_POST_DUCTAL	61916

NOM_PRESS_GEN_5	62452
NOM_PRESS_GEN_5_SYS	62453
NOM_PRESS_GEN_5_DIA	62454
NOM_PRESS_GEN_5_MEAN	62455
NOM_PRESS_GEN_6	62456
NOM_PRESS_GEN_6_SYS	62457
NOM_PRESS_GEN_6_DIA	62458
NOM_PRESS_GEN_6_MEAN	62459
NOM_PRESS_GEN_7	62460
NOM_PRESS_GEN_7_SYS	62461
NOM_PRESS_GEN_7_DIA	62462
NOM_PRESS_GEN_7_MEAN	62463
NOM_PRESS_GEN_8	62464
NOM_PRESS_GEN_8_SYS	62465
NOM_PRESS_GEN_8_DIA	62466
NOM_PRESS_GEN_8_MEAN	62467
NOM_ECG_AMPL_ST_BASELINE_I	62481
NOM_ECG_AMPL_ST_BASELINE_II	62482
NOM_ECG_AMPL_ST_BASELINE_V1	62483
NOM_ECG_AMPL_ST_BASELINE_V2	62484
NOM_ECG_AMPL_ST_BASELINE_V3	62485
NOM_ECG_AMPL_ST_BASELINE_V4	62486
NOM_ECG_AMPL_ST_BASELINE_V5	62487
NOM_ECG_AMPL_ST_BASELINE_V6	62488
NOM_ECG_AMPL_ST_BASELINE_III	62541
NOM_ECG_AMPL_ST_BASELINE_AVR	62542
NOM_ECG_AMPL_ST_BASELINE_AVL	62543
NOM_ECG_AMPL_ST_BASELINE_AVF	62544
NOM_AGE	63504
NOM_AGE_GEST	63505
NOM_AWAY_CORR_COEF	63508
NOM_AWAY_RESP_RATE_SPONT	63509
NOM_AWAY_TC	63510
NOM_BIRTH_LENGTH	63512
NOM_BREATH_RAPID_SHALLOW_INDEX	63513
NOM_C20_PER_C_INDEX	63514
NOM_CARD_CONTRACT_HEATHER_INDEX	63516
NOM_CONC_ALP	63517
NOM_CONC_CA_GEN_NORM	63522
NOM_CONC_CA_SER	63524
NOM_CONC_CO2_TOT	63525
NOM_CONC_CO2_TOT_CALC	63526
NOM_CONC_CREA_SER	63527
NOM_RESP_RATE_SPONT	63528
NOM_CONC_GLO_SER	63529
NOM_CONC_GLU_SER	63530
NOM_CONC_HB_CORP_MEAN	63532
NOM_CONC_K_SER	63535
NOM_CONC_NA_EXCR	63536
NOM_CONC_PCO2_ART_ADJ	63538
NOM_CONC_PCO2_CAP_ADJ	63539
NOM_CONC_PH_CAP_ADJ	63543
NOM_CONC_PH_GEN_ADJ	63544
NOM_CONC_PO2_ART_ADJ	63547
NOM_CONC_PO2_CAP_ADJ	63548
NOM_CREA_OSM	63551

NOM_EEG_BURST_SUPPRN_INDEX	63552
NOM_EEG_ELEC_POTL_CRTX_GAIN_LEFT	63553
NOM_EEG_ELEC_POTL_CRTX_GAIN_RIGHT	63554
NOM_EEG_FREQ_PWR_SPEC_CRTX_MEDIAN_LEFT	63563
NOM_EEG_FREQ_PWR_SPEC_CRTX_MEDIAN_RIGHT	63564
NOM_EEG_PWR_SPEC_ALPHA_ABS_LEFT	63573
NOM_EEG_PWR_SPEC_ALPHA_ABS_RIGHT	63574
NOM_EEG_PWR_SPEC_BETA_ABS_LEFT	63579
NOM_EEG_PWR_SPEC_BETA_ABS_RIGHT	63580
NOM_EEG_PWR_SPEC_DELTA_ABS_LEFT	63587
NOM_EEG_PWR_SPEC_DELTA_ABS_RIGHT	63588
NOM_EEG_PWR_SPEC_THETA_ABS_LEFT	63593
NOM_EEG_PWR_SPEC_THETA_ABS_RIGHT	63594
NOM_ELEC_EVOK_POTL_CRTX_ACOUSTIC_AAI	63603
NOM_EXTRACT_O2_INDEX	63605
NOM_FLOW_AWAY_AIR	63607
NOM_FLOW_AWAY_EXP_ET	63610
NOM_FLOW_AWAY_MAX_SPONT	63613
NOM_FLOW_AWAY_TOT	63617
NOM_FLOW_CO2_PROD_RESP_TIDAL	63618
NOM_FLOW_URINE_PREV_24HR	63619
NOM_FREE_WATER_CLR	63620
NOM_HB_CORP_MEAN	63621
NOM_HEATING_PWR_INCUBATOR	63622
NOM_OUTPUT_CARD_INDEX_ACCEL	63625
NOM_PTC_CNT	63627
NOM_PULS_OXIM_PLETH_GAIN	63629
NOM_RATIO_AWAY_RATE_VOL_AWAY	63630
NOM_RATIO_BUN_CREA	63631
NOM_RATIO_CONC_BLD_UREA_NITROGEN_CREA_CALC	63632
NOM_RATIO_CONC_URINE_CREA_CALC	63633
NOM_RATIO_CONC_URINE_CREA_SER	63634
NOM_RATIO_CONC_URINE_NA_K	63635
NOM_RATIO_PaO2_FIO2	63636
NOM_RATIO_TIME_PD_PT	63637
NOM_RATIO_TIME_PD_PTT	63638
NOM_RATIO_TRAIN_OF_FOUR	63639
NOM_RATIO_URINE_SER_OSM	63640
NOM_RES_AWAY_DYN	63641
NOM_RESP_BREATH_ASSIST_CNT	63642
NOM_RIGHT_HEART_FRACT_EJECT	63643
NOM_TIME_PD_EVOK_REMAIN	63648
NOM_TIME_PD_EXP	63649
NOM_TIME_PD_FROM_LAST_MSMT	63650
NOM_TIME_PD_INSP	63651
NOM_TIME_PD_KAOLIN_CEPHALINE	63652
NOM_TIME_PD_PTT	63653
NOM_TRAIN_OF_FOUR_1	63655
NOM_TRAIN_OF_FOUR_2	63656
NOM_TRAIN_OF_FOUR_3	63657
NOM_TRAIN_OF_FOUR_4	63658
NOM_TRAIN_OF_FOUR_CNT	63659
NOM_TWITCH_AMPL	63660

NOM_UREA_SER	63661
NOM_VENT_ACTIVE	63664
NOM_VENT_AMPL_HFV	63665
NOM_VENT_CONC_AWAY_AGENT_DELTA	63666
NOM_VENT_CONC_AWAY_DESFL_DELTA	63667
NOM_VENT_CONC_AWAY_ENFL_DELTA	63668
NOM_VENT_CONC_AWAY_HALOTH_DELTA	63669
NOM_VENT_CONC_AWAY_ISOFL_DELTA	63670
NOM_VENT_CONC_AWAY_N2O_DELTA	63671
NOM_VENT_CONC_AWAY_O2_CIRCUIT	63672
NOM_VENT_CONC_AWAY_SEVOFL_DELTA	63673
NOM_VENT_PRESS_AWAY_END_EXP_POS_LIMIT_LO	63674
NOM_VENT_PRESS_AWAY_PV	63676
NOM_VENT_TIME_PD_RAMP	63677
NOM_VENT_VOL_AWAY_INSP_TIDAL_HFV	63678
NOM_VENT_VOL_TIDAL_HFV	63679
NOM_VOL_AWAY_EXP_TIDAL_SPONT	63682
NOM_VOL_AWAY_TIDAL_PSV	63683
NOM_VOL_CORP_MEAN	63684
NOM_VOL_FLUID_THORAC	63685
NOM_VOL_FLUID_THORAC_INDEX	63686
NOM_VOL_LVL_LIQUID_BOTTLE_AGENT	63687
NOM_VOL_LVL_LIQUID_BOTTLE_DESFL	63688
NOM_VOL_LVL_LIQUID_BOTTLE_ENFL	63689
NOM_VOL_LVL_LIQUID_BOTTLE_HALOTH	63690
NOM_VOL_LVL_LIQUID_BOTTLE_ISOFL	63691
NOM_VOL_LVL_LIQUID_BOTTLE_SEVOFL	63692
NOM_VOL_MINUTE_AWAY_INSP_HFV	63693
NOM_VOL_URINE_BAL_PD_INSTANT	63694
NOM_VOL_URINE_SHIFT	63695
NOM_VOL_VENT_L_END_SYS_INDEX	63697
NOM_WEIGHT_URINE_COL	63699
NOM_SAT_O2_TISSUE	63840
NOM_CEREB_STATE_INDEX	63841
NOM_SAT_O2_GEN_1	63842
NOM_SAT_O2_GEN_2	63843
NOM_SAT_O2_GEN_3	63844
NOM_SAT_O2_GEN_4	63845
NOM_TEMP_CORE_GEN_1	63846
NOM_TEMP_CORE_GEN_2	63847
NOM_PRESS_BLD_DIFF	63848
NOM_PRESS_BLD_DIFF_GEN_1	63852
NOM_PRESS_BLD_DIFF_GEN_2	63856
NOM_FLOW_PUMP_HEART_LUNG_MAIN	63860
NOM_FLOW_PUMP_HEART_LUNG_SLAVE	63861
NOM_FLOW_PUMP_HEART_LUNG_SUCTION	63862
NOM_FLOW_PUMP_HEART_LUNG_AUX	63863
NOM_FLOW_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN	63864
NOM_FLOW_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE	63865
NOM_TIME_PD_PUMP_HEART_LUNG_AUX_SINCE_START	63866
NOM_TIME_PD_PUMP_HEART_LUNG_AUX_SINCE_STOP	63867

NOM_VOL_DELIV_PUMP_HEART_LUNG_AUX	63868
NOM_VOL_DELIV_TOTAL_PUMP_HEART_LUNG_AUX	63869
NOM_TIME_PD_PLEGIA_PUMP_HEART_LUNG_AUX	63870
NOM_TIME_PD_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN_SINCE_START	63871
NOM_TIME_PD_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN_SINCE_STOP	63872
NOM_VOL_DELIV_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN	63873
NOM_VOL_DELIV_TOTAL_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN	63874
NOM_TIME_PD_PLEGIA_PUMP_HEART_LUNG_CARDIOPLEGIA_MAIN	63875
NOM_TIME_PD_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE_SINCE_START	63876
NOM_TIME_PD_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE_SINCE_STOP	63877
NOM_VOL_DELIV_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE	63878
NOM_VOL_DELIV_TOTAL_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE	63879
NOM_TIME_PD_PLEGIA_PUMP_HEART_LUNG_CARDIOPLEGIA_SLAVE	63880
NOM_RATIO_INSP_TOTAL_BREATH_SPONT	63888
NOM_VENT_PRESS_AWAY_END_EXP_POS_TOTAL	63889
NOM_COMPL_LUNG_PAV	63890
NOM_RES_AWAY_PAV	63891
NOM_RES_AWAY_EXP_TOTAL	63892
NOM_ELAS_LUNG_PAV	63893
NOM_BREATH_RAPID_SHALLOW_INDEX_NORM	63894

The following code from the object oriented partition are used for the alert source:

NOM_MOC_VMO	1
NOM_MOC_VMO_METRIC_NU	6
NOM_MOC_VMO_METRIC_SA_RT	9
NOM_MOC_VMS_MDS	33
NOM_MOC_VMS_MDS_COMPOS_SINGLE_BED	35
NOM_MOC_VMS_MDS_SIMP	37
NOM_MOC_BATT	41
NOM_MOC_PT_DEMOG	42
NOM_MOC_VMO_AL_MON	54
NOM_ATTR_GRP_AL_MON	2049
NOM_ATTR_GRP_METRIC_VAL_OBS	2051
NOM_ATTR_GRP_PT_DEMOG	2055
NOM_ATTR_GRP_SYS_APPL	2058
NOM_ATTR_GRP_SYS_ID	2059
NOM_ATTR_GRP_SYS_PROD	2060
NOM_ATTR_GRP_VMO_DYN	2064
NOM_ATTR_GRP_VMO_STATIC	2065
NOM_ATTR_AL_MON_P_AL_LIST	2306
NOM_ATTR_AL_MON_T_AL_LIST	2308
NOM_ATTR_ALTITUDE	2316
NOM_ATTR_AREA_APPL	2317
NOM_ATTR_COLOR	2321

NOM_ATTR_DEV_AL_COND	2326
NOM_ATTR_DISP_RES	2327
NOM_ATTR_GRID_VIS_I16	2330
NOM_ATTR_ID_ASSOC_NO	2333
NOM_ATTR_ID_BED_LABEL	2334
NOM_ATTR_ID_HANDLE	2337
NOM_ATTR_ID_LABEL	2340
NOM_ATTR_ID_LABEL_STRING	2343
NOM_ATTR_ID_MODEL	2344
NOM_ATTR_ID_PROD_SPECN	2349
NOM_ATTR_ID_TYPE	2351
NOM_ATTR_LINE_FREQ	2357
NOM_ATTR_LOCALIZN	2359
NOM_ATTR_METRIC_INFO_LABEL	2364
NOM_ATTR_METRIC_INFO_LABEL_STR	2365
NOM_ATTR_METRIC_SPECN	2367
NOM_ATTR_METRIC_STAT	2368
NOM_ATTR_MODE_MSMT	2373
NOM_ATTR_MODE_OP	2374
NOM_ATTR_NOM_VERS	2376
NOM_ATTR_NU_CMPD_VAL_OBS	2379
NOM_ATTR_NU_VAL_OBS	2384
NOM_ATTR_PT_BSA	2390
NOM_ATTR_PT_DEMOG_ST	2391
NOM_ATTR_PT_DOB	2392
NOM_ATTR_PT_ID	2394
NOM_ATTR_PT_NAME_FAMILY	2396
NOM_ATTR_PT_NAME_GIVEN	2397
NOM_ATTR_PT_SEX	2401
NOM_ATTR_PT_TYPE	2402
NOM_ATTR_SA_CALIB_I16	2404
NOM_ATTR_SA_CMPD_VAL_OBS	2407
NOM_ATTR_SA_RANGE_PHYS_I16	2410
NOM_ATTR_SA_SPECN	2413
NOM_ATTR_SA_VAL_OBS	2414
NOM_ATTR_SCALE_SPECN_I16	2415
NOM_ATTR_STD_SAFETY	2434
NOM_ATTR_SYS_ID	2436
NOM_ATTR_SYS_SPECN	2437
NOM_ATTR_SYS_TYPE	2438
NOM_ATTR_TIME_ABS	2439
NOM_ATTR_TIME_PD_SAMP	2445
NOM_ATTR_TIME_REL	2447
NOM_ATTR_TIME_STAMP_ABS	2448
NOM_ATTR_TIME_STAMP_REL	2449
NOM_ATTR_UNIT_CODE	2454
NOM_ATTR_VMS_MDS_STAT	2471
NOM_ATTR_PT_AGE	2520
NOM_ATTR_PT_HEIGHT	2524
NOM_ATTR_PT_WEIGHT	2527
NOM_ATTR_SA_FIXED_VAL_SPECN	2582
NOM_ATTR_SYS_ADT_ST	2586
NOM_ATTR_PT_PACED_MODE	2590
NOM_ACT_POLL_MDIB_DATA	3094
NOM_NOTI_MDS_CREAT	3334
NOM_NOTI_CONN_INDIC	3351

NOM_DEV_METER_CONC_SKIN_GAS	4264
NOM_DEV_METER_FLOW_BLD	4284
NOM_DEV_ANALY_CONC_GAS_MULTI_PARAM_MDS	4113
NOM_DEV_METER_CONC_SKIN_GAS_MDS	4265
NOM_DEV_MON_PHYSIO_MULTI_PARAM_MDS	4429
NOM_DEV_PUMP_INFUS_MDS	4449
NOM_DEV_SYS_PT_VENT_MDS	4465
NOM_DEV_SYS_MULTI_MODAL_MDS	4493
NOM_DEV_SYS_VS_CONFIG_MDS	5209
NOM_DEV_SYS_VS_UNCONFIG_MDS	5213
NOM_DEV_ANALY_SAT_O2_VMD	4106
NOM_DEV_ANALY_CONC_GAS_MULTI_PARAM_VMD	4114
NOM_DEV_ANALY_FLOW_AWAY_VMD	4130
NOM_DEV_ANALY_CARD_OUTPUT_VMD	4134
NOM_DEV_ANALY_PRESS_BLD_VMD	4174
NOM_DEV_ANALY_RESP_RATE_VMD	4186
NOM_DEV_CALC_VMD	4206
NOM_DEV_ECG_VMD	4262
NOM_DEV_METER_CONC_SKIN_GAS_VMD	4266
NOM_DEV_EEG_VMD	4274
NOM_DEV_METER_TEMP_BLD_VMD	4350
NOM_DEV_METER_TEMP_VMD	4366
NOM_DEV_MON_BLD_CHEM_MULTI_PARAM_VMD	4398
NOM_DEV_SYS_PT_VENT_VMD	4466
NOM_DEV_SYS_MULTI_MODAL_VMD	4494
NOM_DEV_SYS_ANESTH_VMD	4506
NOM_DEV_GENERAL_VMD	5122
NOM_DEV_ECG_RESP_VMD	5130
NOM_DEV_ARRHY_VMD	5134
NOM_DEV_PULS_VMD	5138
NOM_DEV_ST_VMD	5142
NOM_DEV_CO2_VMD	5146
NOM_DEV_PRESS_BLD_NONINV_VMD	5150
NOM_DEV_CEREB_PERF_VMD	5154
NOM_DEV_CO2_CTS_VMD	5158
NOM_DEV_CO2_TCUT_VMD	5162
NOM_DEV_O2_VMD	5166
NOM_DEV_O2_CTS_VMD	5170
NOM_DEV_O2_TCUT_VMD	5174
NOM_DEV_TEMP_DIFF_VMD	5178
NOM_DEV_CNTRL_VMD	5182
NOM_DEV_WEDGE_VMD	5190
NOM_DEV_O2_VEN_SAT_VMD	5194
NOM_DEV_CARD_RATE_VMD	5202
NOM_DEV_PLETH_VMD	5238
NOM_ATTR_PT_ID_INT	61441
NOM_SAT_O2_TONE_FREQ	61448
NOM_ATTR_CMPD_REF_LIST	61449
NOM_OBJ_HIF_KEY	61584
NOM_OBJ_DISP	61616
NOM_OBJ_SOUND_GEN	61648
NOM_OBJ_SETTING	61649
NOM_OBJ_PRINTER	61650
NOM_OBJ_EVENT	61683
NOM_OBJ_BATT_CHARGER	61690
NOM_OBJ_ECG_OUT	61691

NOM_OBJ_INPUT_DEV	61692
NOM_OBJ_NETWORK	61693
NOM_OBJ_QUICKLINK	61694
NOM_OBJ_SPEAKER	61695
NOM_ATTR_NET_ADDR_INFO	61696
NOM_ATTR_PCOL_SUPPORT	61697
NOM_OBJ_PUMP	61716
NOM_OBJ_IR	61717
NOM_ATTR_PT_NOTES1	61737
NOM_ATTR_PT_NOTES2	61738
NOM_ACT_POLL_MDIB_DATA_EXT	61755
NOM_ATTR_TIME_PD_POLL	61758
NOM_DEV_ANALY_PULS_CONT	61800
NOM_DEV_ANALY_BISPECTRAL_INDEX_VMD	61806
NOM_DEV_HIRES_TREND	61820
NOM_DEV_HIRES_TREND_MDS	61821
NOM_DEV_HIRES_TREND_VMD	61822
NOM_DEV_MON_PT_EVENT_VMD	61826
NOM_DEV_DERIVED_MSMT	61828
NOM_DEV_DERIVED_MSMT_MDS	61829
NOM_DEV_DERIVED_MSMT_VMD	61830
NOM_OBJ_SENSOR	61902
NOM_OBJ_XDUCR	61903
NOM_OBJ_CHAN_1	61916
NOM_OBJ_CHAN_2	61917
NOM_OBJ_AWAY_AGENT_1	61918
NOM_OBJ_AWAY_AGENT_2	61919
NOM_ATTR_PT_BSA_FORMULA	61932
NOM_ATTR_MDS_GEN_INFO	61946
NOM_OBJ_HIF_MOUSE	61983
NOM_OBJ_HIF_TOUCH	61984
NOM_OBJ_HIF_SPEEDPOINT	61985
NOM_OBJ_HIF_ALARMBOX	61986
NOM_OBJ_BUS_I2C	61987
NOM_OBJ_CPU_SEC	61988
NOM_OBJ_LED	61990
NOM_OBJ_RELAY	61991
NOM_ATTR_POLL_OBJ_PRIO_NUM	61992
NOM_OBJ_BATT_1	61996
NOM_OBJ_BATT_2	61997
NOM_OBJ_DISP_SEC	61998
NOM_OBJ_AGM	61999
NOM_ATTR_POLL_NU_PRIO_LIST	62009
NOM_ATTR_POLL_RTSA_PRIO_LIST	62010
NOM_OBJ_CABLE	62016

The following codes from the event partition are used for the alert code:

NOM_EVT_ABSENT	4
NOM_EVT_CONTAM	14
NOM_EVT_DISCONN	22
NOM_EVT_DISTURB	24
NOM_EVT_EMPTY	26
NOM_EVT_ERRATIC	32
NOM_EVT_EXH	36
NOM_EVT_FAIL	38
NOM_EVT_HI	40
NOM_EVT_IRREG	58

NOM_EVT_LO	62
NOM_EVT_MALF	70
NOM_EVT_NOISY	74
NOM_EVT_OBSTRUC	80
NOM_EVT_REVERSED	96
NOM_EVT_SUST	106
NOM_EVT_UNAVAIL	110
NOM_EVT_UNDEF	112
NOM_EVT_WARMING	124
NOM_EVT_WEAK	128
NOM_EVT_BREATH_ABSENT	136
NOM_EVT_CALIB_FAIL	138
NOM_EVT_CONFIG_ERR	142
NOM_EVT_RANGE_ERR	164
NOM_EVT_RANGE_OVER	166
NOM_EVT_SRC_ABSENT	174
NOM_EVT_SYNCH_ERR	182
NOM_EVT_BATT_LO	194
NOM_EVT_BATT_PROB	198
NOM_EVT_CUFF_NOT_DEFLATED	230
NOM_EVT_CUFF_INFLAT_OVER	232
NOM_EVT_EQUIP_MALF	242
NOM_EVT_TUBE_OCCL	250
NOM_EVT_GAS_AGENT_IDENT_MALF	258
NOM_EVT_LEAD_DISCONN	268
NOM_EVT_LEADS_OFF	274
NOM_EVT_O2_SUPPLY_LO	296
NOM_EVT_OPTIC_MODULE_ABSENT	298
NOM_EVT_OPTIC_MODULE_DEFECT	300
NOM_EVT_SENSOR_DISCONN	308
NOM_EVT_SENSOR_MALF	310
NOM_EVT_SENSOR_PROB	312
NOM_EVT_SW_VER_UNK	322
NOM_EVT_TUBE_DISCONN	326
NOM_EVT_TUBE_OBSTRUC	330
NOM_EVT_XDUCR_DISCONN	336
NOM_EVT_XDUCR_MALF	338
NOM_EVT_INTENS_LIGHT_ERR	350
NOM_EVT_MSMT_DISCONN	352
NOM_EVT_MSMT_ERR	354
NOM_EVT_MSMT_FAIL	356
NOM_EVT_MSMT_INOP	358
NOM_EVT_MSMT_INTERRUP	362
NOM_EVT_MSMT_RANGE_OVER	364
NOM_EVT_MSMT_RANGE_UNDER	366
NOM_EVT_SIG_LO	380
NOM_EVT_SIG_UNANALYZEABLE	384
NOM_EVT_TEMP_HI_GT_LIM	394
NOM_EVT_UNSUPPORTED	400
NOM_EVT_WAVE_ARTIF_ERR	432
NOM_EVT_WAVE_SIG_QUAL_ERR	434
NOM_EVT_MSMT_INTERF_ERR	436
NOM_EVT_WAVE_OSCIL_ABSENT	442
NOM_EVT_VOLTAGE_OUT_OF_RANGE	460
NOM_EVT_INCOMPAT	600
NOM_EVT_ADVIS_CHK	6658

NOM_EVT_ADVIS_CALIB_AND_ZERO_CHK	6664
NOM_EVT_ADVIS_CONFIG_CHK	6666
NOM_EVT_ADVIS_SETTINGS_CHK	6668
NOM_EVT_ADVIS_SETUP_CHK	6670
NOM_EVT_ADVIS_SRC_CHK	6672
NOM_EVT_BATT_COND	6676
NOM_EVT_BATT_REPLACE	6678
NOM_EVT_ADVIS_CABLE_CHK	6680
NOM_EVT_ADVIS_GAS_AGENT_CHK	6688
NOM_EVT_ADVIS_LEAD_CHK	6690
NOM_EVT_ADVIS_SENSOR_CHK	6696
NOM_EVT_ADVIS_GAIN_DECR	6704
NOM_EVT_ADVIS_GAIN_INCR	6706
NOM_EVT_ADVIS_UNIT_CHK	6710
NOM_EVT_APNEA	3072
NOM_EVT_ECG_ASYSTOLE	3076
NOM_EVT_ECG_BEAT_MISSED	3078
NOM_EVT_ECG_BIGEM	3082
NOM_EVT_ECG_BRADY_EXTREME	3086
NOM_EVT_ECG_PACING_NON_CAPT	3102
NOM_EVT_ECG_PAUSE	3108
NOM_EVT_ECG_TACHY_EXTREME	3122
NOM_EVT_ECG_CARD_BEAT_RATE_IRREG	3158
NOM_EVT_ECG_PACER_NOT_PACING	3182
NOM_EVT_ECG_SV_TACHY	3192
NOM_EVT_ECG_V_P_C_RonT	3206
NOM_EVT_ECG_V_P_C_MULTIFORM	3208
NOM_EVT_ECG_V_P_C_PAIR	3210
NOM_EVT_ECG_V_P_C_RUN	3212
NOM_EVT_ECG_V_RHY	3220
NOM_EVT_ECG_V_TACHY	3224
NOM_EVT_ECG_V_TACHY_NON_SUST	3226
NOM_EVT_ECG_V_TRIGEM	3236
NOM_EVT_DESAT	3246
NOM_EVT_ECG_V_P_C_RATE	3252
NOM_EVT_STAT_AL_OFF	6144
NOM_EVT_STAT_BATT_CHARGING	6150
NOM_EVT_STAT_CALIB_MODE	6152
NOM_EVT_STAT_CALIB_RUNNING	6154
NOM_EVT_STAT_CALIB_INVIVO_RUNNING	6156
NOM_EVT_STAT_CALIB_LIGHT_RUNNING	6158
NOM_EVT_STAT_CALIB_PREINS_RUNNING	6160
NOM_EVT_STAT_SELFTEST_RUNNING	6164
NOM_EVT_STAT_ZERO_RUNNING	6170
NOM_EVT_STAT_OPT_MOD_SENSOR_CONN	6172
NOM_EVT_STAT_OPT_MOD_SENSOR_WARMING	6174
NOM_EVT_STAT_SENSOR_WARMING	6176
NOM_EVT_STAT_WARMING	6178
NOM_EVT_STAT_ECG_AL_ALL_OFF	6182
NOM_EVT_STAT_ECG_AL_SOME_OFF	6184
NOM_EVT_STAT_LEARN	6224
NOM_EVT_STAT_OFF	6226
NOM_EVT_STAT_STANDBY	6228
NOM_EVT_STAT_DISCONN	6256
NOM_EVT_ADVIS_CALIB_REQD	6662
NOM_EVT_ECG_V_FIB_TACHY	61444

NOM_EVT_WAIT_CAL	61678
NOM_EVT_ADVIS_CHANGE_SITE	61682
NOM_EVT_ADVIS_CHECK_SITE_TIME	61684
NOM_EVT_STAT_FW_UPDATE_IN_PROGRESS	61688
NOM_EVT_EXT_DEV_AL_CODE_1	61690
NOM_EVT_EXT_DEV_AL_CODE_2	61692
NOM_EVT_EXT_DEV_AL_CODE_3	61694
NOM_EVT_EXT_DEV_AL_CODE_4	61696
NOM_EVT_EXT_DEV_AL_CODE_5	61698
NOM_EVT_EXT_DEV_AL_CODE_6	61700
NOM_EVT_EXT_DEV_AL_CODE_7	61702
NOM_EVT_EXT_DEV_AL_CODE_8	61704
NOM_EVT_EXT_DEV_AL_CODE_9	61706
NOM_EVT_EXT_DEV_AL_CODE_10	61708
NOM_EVT_EXT_DEV_AL_CODE_11	61710
NOM_EVT_EXT_DEV_AL_CODE_12	61712
NOM_EVT_EXT_DEV_AL_CODE_13	61714
NOM_EVT_EXT_DEV_AL_CODE_14	61716
NOM_EVT_EXT_DEV_AL_CODE_15	61718
NOM_EVT_EXT_DEV_AL_CODE_16	61720
NOM_EVT_EXT_DEV_AL_CODE_17	61722
NOM_EVT_EXT_DEV_AL_CODE_18	61724
NOM_EVT_EXT_DEV_AL_CODE_19	61726
NOM_EVT_EXT_DEV_AL_CODE_20	61728
NOM_EVT_EXT_DEV_AL_CODE_21	61730
NOM_EVT_EXT_DEV_AL_CODE_22	61732
NOM_EVT_EXT_DEV_AL_CODE_23	61734
NOM_EVT_EXT_DEV_AL_CODE_24	61736
NOM_EVT_EXT_DEV_AL_CODE_25	61738
NOM_EVT_EXT_DEV_AL_CODE_26	61740
NOM_EVT_EXT_DEV_AL_CODE_27	61742
NOM_EVT_EXT_DEV_AL_CODE_28	61744
NOM_EVT_EXT_DEV_AL_CODE_29	61746
NOM_EVT_EXT_DEV_AL_CODE_30	61748
NOM_EVT_EXT_DEV_AL_CODE_31	61750
NOM_EVT_EXT_DEV_AL_CODE_32	61752
NOM_EVT_EXT_DEV_AL_CODE_33	61754
NOM_EVT_ST_MULTI	61756
NOM_EVT_ADVIS_BSA_REQD	61760
NOM_EVT_ADVIS_PRESUMED_CVP	61762
NOM_EVT_MSMT_UNSUPPORTED	61764
NOM_EVT_BRADY	61766
NOM_EVT_TACHY	61768
NOM_EVT_ADVIS_CHANGE_SCALE	61770
NOM_EVT_MSMT_RESTART	61772
NOM_EVT_TOO_MANY_AGENTS	61774
NOM_EVT_STAT_PULSE_SRC_RANGE_OVER	61778
NOM_EVT_STAT_PRESS_SRC_RANGE_OVER	61780
NOM_EVT_MUSCLE_NOISE	61782
NOM_EVT_LINE_NOISE	61784
NOM_EVT_IMPED_HI	61786
NOM_EVT_AGENT_MIX	61788
NOM_EVT_IMPEDS_HI	61790
NOM_EVT_ADVIS_PWR_HI	61792
NOM_EVT_ADVIS_PWR_OFF	61794
NOM_EVT_ADVIS_PWR_OVER	61796

NOM_EVT_ADVIS_DEACT	61798
NOM_EVT_CO_WARNING	61800
NOM_EVT_ADVIS_NURSE_CALL	61802
NOM_EVT_COMP_MALF	61804
NOM_EVT_AGENT_MEAS_MALF	61806
NOM_EVT_ADVIS_WATER_TRAP_CHK	61808
NOM_EVT_STAT_AGENT_CALC_RUNNING	61810
NOM_EVT_ADVIS_ADAPTER_CHK	61814
NOM_EVT_ADVIS_PUMP_OFF	61816
NOM_EVT_ZERO_FAIL	61818
NOM_EVT_ADVIS_ZERO_REQD	61820
NOM_EVT_EXTR_HI	61830
NOM_EVT_EXTR_LO	61832
NOM_EVT_LEAD_DISCONN_YELLOW	61833
NOM_EVT_LEAD_DISCONN_RED	61834
NOM_EVT_CUFF_INFLAT_OVER_YELLOW	61835
NOM_EVT_CUFF_INFLAT_OVER_RED	61836
NOM_EVT_CUFF_NOT_DEFLATED_YELLOW	61837
NOM_EVT_CUFF_NOT_DEFLATED_RED	61838
NOM_EVT_ADVIS_ACTION_REQD	61840
NOM_EVT_OUT_OF_AREA	61842
NOM_EVT_LEADS_DISCONN	61844
NOM_EVT_DEV_ASSOC_CHK	61846
NOM_EVT_SYNCH_UNSUPPORTED	61848
NOM_EVT_ECG_ADVIS_SRC_CHK	61850
NOM_EVT_ALARM_TECH	61852
NOM_EVT_ALARM_TECH_YELLOW	61854
NOM_EVT_ALARM_TECH_RED	61856
NOM_EVT_ALARM_MED_YELLOW_SHORT	61858
NOM_EVT_ALARM_MED_YELLOW	61860
NOM_EVT_ALARM_MED_RED	61862
NOM_EVT_TELE_EQUIP_MALF	61874
NOM_EVT_SYNCH_ERR_ECG	61876
NOM_EVT_SYNCH_ERR_SPO2T	61878
NOM_EVT_ADVIS_ACTION_REQD_YELLOW	61880
NOM_EVT_ADVIS_NBP_SEQ_COMPLETED	61882
NOM_EVT_PACER_OUTPUT_LO	61884
NOM_EVT_ALARM_MORE_TECH	61886
NOM_EVT_ALARM_MORE_TECH_YELLOW	61888
NOM_EVT_ALARM_MORE_TECH_RED	61890
NOM_EVT_ADVIS_PATIENT_CONFLICT	61892
NOM_EVT_SENSOR_REPLACE	61894
NOM_EVT_ECG_ATR_FIB	61896
NOM_EVT_LIMITED_CONNECTIVITY	61900
NOM_EVT_DISABLED	61924
NOM_EVT_ECG_ABSENT	61926
NOM_EVT_SRR_INTERF	61928
NOM_EVT_SRR_INVALID_CHAN	61930
NOM_EVT_EXT_DEV_DEMO	62032
NOM_EVT_EXT_DEV_MONITORING	62034

Private Unicode Characters

The IntelliVue monitor may use the following private codes for UNICODE characters:

```
#define SUBSCRIPT_CAPITAL_E_CHAR         0xE145
           /* SUBSCRIPT CAPITAL E                           */
#define SUBSCRIPT_CAPITAL_L_CHAR         0xE14C
           /* SUBSCRIPT CAPITAL L                           */
#define LITER_PER_CHAR                   0xE400
           /* LITER PER - used in 4 char unit "l/min"     */
#define HYDROGEN_CHAR                    0xE401
           /* HYDROGEN - Used in 4 char unit "cmH2O"      */
#define ALARM_STAR_CHAR                  0xE40D
           /* ALARM STAR                                    */
#define CAPITAL_V_WITH_DOT_ABOVE_CHAR    0xE425
           /* CAPITAL_V_WITH_DOT_ABOVE (V with dot)       */
#define ZERO_WIDTH_NO_BREAK_SPACE_CHAR   0xFEFF
           /* The character 0xFEFF is used as FILL character.
           For each wide asian character, a FILL character is
           appended for size calculations. */
```

List of Constants Used Within the Protocol Definition

RO Types

```
#define ROIV_APDU                1
#define RORS_APDU                2
#define ROER_APDU                3
#define ROLRS_APDU               5
```

ROLRS Identifier

```
#define RORLS_FIRST       1   /* set in the first message */
#define RORLS_NOT_FIRST_NOT_LAST 2
#define RORLS_LAST        3   /* last RORLSapdu, one RORSapdu
                                 to follow */
```

ROSE Commands

```
typedef u_16            CMDType;
#define CMD_EVENT_REPORT                 0
#define CMD_CONFIRMED_EVENT_REPORT       1
#define CMD_GET                          3
#define CMD_SET                          4
#define CMD_CONFIRMED_SET                5
#define CMD_CONFIRMED_ACTION             7
```

ROER Error Values

```
#define NO_SUCH_OBJECT_CLASS              0
#define NO_SUCH_OBJECT_INSTANCE           1
#define ACCESS_DENIED                     2
#define GET_LIST_ERROR                    7
#define SET_LIST_ERROR                    8
#define NO_SUCH_ACTION                    9
#define PROCESSING_FAILURE                10
#define INVALID_ARGUMENT_VALUE            15
#define INVALID_SCOPE                     16
#define INVALID_OBJECT_INSTANCE           17
```

Action and Event Types

The Action and Event Types are defined in the Object Oriented Elements partition of the nomenclature.

```
#define NOM_NOTI_MDS_CREAT                3334
   /* MDS Create Notification */
#define NOM_NOTI_CONN_INDIC               3351
    /* connect indication event type */
#define NOM_ACT_POLL_MDIB_DATA            3094
    /* poll data action */
#define NOM_ACT_POLL_MDIB_DATA_EXT        61755
    /* extended poll data action */
```

Protocol Identification

The IDs for the protocol identification are from the Infrastructure nomenclature partition.

```
#define NOM_POLL_PROFILE_SUPPORT          1
    /* id for polling profile */
#define NOM_MDIB_OBJ_SUPPORT              258
    /* supported objects for the active profile */
#define NOM_ATTR_POLL_PROFILE_EXT         61441
    /* id for poll profile extensions opt. package */
```

Association Control

```
#define MDDL_VERSION1                     0x80000000
    /* Data Export Protocol Version */
#define NOMEN_VERSION                     0x40000000
    /* Nomenclature Version */
#define SYST_CLIENT                       0x80000000
    /* System Type Client */
#define SYST_SERVER                       0x00800000
    /* System Type Server */
#define HOT_START                         0x80000000
    /* Startup Mode Hotstart */
#define WARM_START                        0x40000000
    /* Startup Mode Warmstart */
#define COLD_START                        0x20000000
    /* Startup Mode Coldstart */
#define POLL_PROFILE_REV_0                0x80000000
    /* Poll Profile Revision */
#define P_OPT_DYN_CREATE_OBJECTS          0x40000000
    /* option dynamic object creation */
#define P_OPT_DYN_DELETE_OBJECTS          0x20000000
    /* option dynamic object deletion */
#define POLL_EXT_PERIOD_NU_1SEC           0x80000000
    /* 1 sec Real-time Numerics */
#define POLL_EXT_PERIOD_NU_AVG_12SEC      0x40000000
    /* 12 sec averaged Numerics */
#define POLL_EXT_PERIOD_NU_AVG_60SEC      0x20000000
    /* 1 min. averaged Numerics */
#define POLL_EXT_PERIOD_NU_AVG_300SEC     0x10000000
    /* 5 min. averaged Numerics */
#define POLL_EXT_PERIOD_RTSA              0x08000000
    /* allow enumeration objects */
#define POLL_EXT_ENUM                     0x04000000
    /* allow numeric priority list to be set */
#define POLL_EXT_NU_PRIO_LIST             0x02000000
    /* send timestamps for numerics with dynamic modalities */
#define POLL_EXT_DYN_MODALITIES           0x01000000
```

Label Mapping Table

With IntelliVue release G the nomenclature of some numeric and wave labels have been changed. The labels that previously resided in the namespace NOM_EMFC mainly used by VueLink devices have been moved into the NOM_SCADA namespace partition and the new defined NOM_SETTING namespace partition.

If your want to integrate support for the new nomenclature definitions in your existing client application you have to accept both label ids.

To guide you trough the transition of the nomenclature changes introduced in release G, you may find the following table useful. The revision F label is given first followed by the new label. For further descriptions of the old labels see the Revision F of the Data Export Programmers Guide.

Label Definition	Label Id
NLS_NOM_EMFC_sAVDel	(0x040180CC)
NLS_NOM_SETT_APNEA_ALARM_DELAY	(0x0402F8D9)
NLS_NOM_EMFC_C20_PER_C	(0x04010E78)
NLS_NOM_C20_PER_C_INDEX	(0x0002F81A)
NLS_NOM_EMFC_Rf_V5	(0x0401075C)
NLS_NOM_ECG_AMPL_ST_BASELINE_V5	(0x0002F417)
NLS_NOM_EMFC_Urine	(0x04010BD8)
NLS_NOM_FLOW_URINE_PREV_24HR	(0x0002F883)
NLS_NOM_EMFC_PT	(0x040105E4)
NLS_NOM_TIME_PD_PT	(0x0002F18B)
NLS_NOM_EMFC_SerCa	(0x0401059C)
NLS_NOM_CONC_CA_SER	(0x0002F824)
NLS_NOM_EMFC_sBPAl	(0x0401A024)
NLS_NOM_SETT_PRESS_AL_ONOFF	(0x0402F8F7)
NLS_NOM_EMFC_SetTmp	(0x04010AD8)
NLS_NOM_TEMP_BODY	(0x00024B5C)
NLS_NOM_EMFC_sCO2Wm	(0x0401815C)
NLS_NOM_SETT_VENT_CO2_WARMING_MONITOR_ONOFF	(0x0402F915)
NLS_NOM_EMFC_sAPkFl	(0x04018030)
NLS_NOM_SETT_FLOW_AWAY_INSP_APNEA	(0x0402F8ED)
NLS_NOM_EMFC_SerGlc	(0x04010590)
NLS_NOM_CONC_GLU_SER	(0x0002F82A)
NLS_NOM_EMFC_RT_PCT_BE	(0x04010810)
NLS_NOM_EEG_PWR_SPEC_BETA_REL_RIGHT	(0x0002F860)
NLS_NOM_EMFC_T4	(0x04010414)
NLS_NOM_TEMP_GEN_4	(0x0002F0CA)
NLS_NOM_EMFC_GOT	(0x0401060C)
NLS_NOM_CONC_GOT	(0x0002F188)
NLS_NOM_EMFC_highO2	(0x0401A020)
NLS_NOM_SETT_VENT_CONC_AWAY_O2_LIMIT_HI	(0x0402F919)
NLS_NOM_EMFC_MCV	(0x040105D4)
NLS_NOM_VOL_CORP_MEAN	(0x0002F8C4)

```
NLS_NOM_EMFC_sEnTrg                                         (0x040180B4)
NLS_NOM_SETT_TRIG_ONOFF                                     (0x0402F90C)

NLS_NOM_EMFC_Plts                                           (0x040105D0)
NLS_NOM_PLTS_CNT                                            (0x0002F167)

NLS_NOM_EMFC_sLInPr                                         (0x04018100)
NLS_NOM_SETT_PRESS_AWAY_MIN                                 (0x040250F2)

NLS_NOM_EMFC_GGT                                            (0x04010608)
NLS_NOM_CONC_GGT                                            (0x0002F189)

NLS_NOM_EMFC_sAGTWm                                         (0x0401816C)
NLS_NOM_SETT_VENT_AGENT_WARMING_MONITOR_ONOFF               (0x0402F90D)

NLS_NOM_EMFC_sAPVhP                                         (0x0401807C)
NLS_NOM_SETT_VENT_PRESS_AWAY_MAX_PV_APNEA                   (0x0402F931)

NLS_NOM_EMFC_sfgSEV                                         (0x040181AC)
NLS_NOM_SETT_CONC_AWAY_SEVOFL                               (0x040251E4)

NLS_NOM_EMFC_highMV                                         (0x0401A02C)
NLS_NOM_SETT_VENT_VOL_MINUTE_AWAY_LIMIT_HI                  (0x0402F94B)

NLS_NOM_EMFC_P6_MEAN                                        (0x04010407)
NLS_NOM_PRESS_GEN_6_MEAN                                    (0x0002F3FB)

NLS_NOM_EMFC_SpRR                                           (0x04010BF4)
NLS_NOM_RESP_RATE_SPONT                                     (0x0002F828)

NLS_NOM_EMFC_Sample                                         (0x04010AAC)
NLS_NOM_SETT_SAMPLE                                         (0x0402F956)

NLS_NOM_EMFC_CK_MM                                          (0x04010604)
NLS_NOM_CONC_CREA_KIN_MM                                    (0x0002F17F)

NLS_NOM_EMFC_sFlas                                          (0x040181F8)
NLS_NOM_SETT_VENT_FLOW_AWAY_ASSIST                          (0x0402F91C)

NLS_NOM_EMFC_RBC                                            (0x040105CC)
NLS_NOM_RB_CNT                                              (0x0002F169)

NLS_NOM_EMFC_TOF4                                           (0x04010DCC)
NLS_NOM_TRAIN_OF_FOUR_4                                     (0x0002F8AA)

NLS_NOM_EMFC_sSens                                          (0x04018188)
NLS_NOM_SETT_SENS_LEVEL                                     (0x0402F904)

NLS_NOM_EMFC_sSIMV                                          (0x04018118)
NLS_NOM_SETT_VENT_MODE_SYNC_MAND_INTERMIT                   (0x0402F924)

NLS_NOM_EMFC_UrCa                                           (0x04010624)
NLS_NOM_CONC_CA_URINE                                       (0x0002F19C)

NLS_NOM_EMFC_vECG                                           (0x0401119C)
NLS_NOM_ELEC_POTL_VECT                                      (0x0002F874)

NLS_NOM_EMFC_PCO2_ADJ                                       (0x04010A7C)
NLS_NOM_CONC_PCO2_GEN_ADJ                                   (0x0002F834)

NLS_NOM_EMFC_BLANK                                          (0x04010960)
NLS_NOM_METRIC_NOS                                          (0x0002EFFF)

NLS_NOM_EMFC_sPIP                                           (0x040180FC)
NLS_NOM_SETT_PRESS_AWAY_INSP_MAX                            (0x04025109)
```

NLS_NOM_EMFC_sALMRT	(0x040180F0)
NLS_NOM_SETT_VENT_TIME_PD_RAMP_AL	(0x0402F946)
NLS_NOM_EMFC_sfgO2	(0x040181B4)
NLS_NOM_SETT_FLOW_AWAY_O2	(0x0402F87F)
NLS_NOM_EMFC_UrNaEx	(0x040101B4)
NLS_NOM_CONC_NA_EXCR	(0x0002F830)
NLS_NOM_EMFC_P1_SYS	(0x04010031)
NLS_NOM_PRESS_GEN_1_SYS	(0x0002F0A5)
NLS_NOM_EMFC_LT_MPF	(0x040107F8)
NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_MEDIAN_LEFT	(0x0002F84B)
NLS_NOM_EMFC_extHR	(0x04010700)
NLS_NOM_CARD_BEAT_RATE_EXT	(0x0002F81B)
NLS_NOM_EMFC_TOF1	(0x04010DC0)
NLS_NOM_TRAIN_OF_FOUR_1	(0x0002F8A7)
NLS_NOM_EMFC_L_V4	(0x04010770)
NLS_NOM_ECG_ELEC_POTL_V4	(0x00020106)
NLS_NOM_EMFC_PPV	(0x040111E0)
NLS_NOM_PULS_PRESS_VAR	(0x0002F0E3)
NLS_NOM_EMFC_SO2_CALC	(0x04010A90)
NLS_NOM_SAT_O2_CALC	(0x0002F89C)
NLS_NOM_EMFC_TGL	(0x0401061C)
NLS_NOM_CONC_TGL	(0x0002F16F)
NLS_NOM_EMFC_P5	(0x04010400)
NLS_NOM_PRESS_GEN_5	(0x0002F3F4)
NLS_NOM_EMFC_PcCO2	(0x04010A78)
NLS_NOM_CONC_PCO2_CAP	(0x0002F159)
NLS_NOM_EMFC_Fe	(0x04010614)
NLS_NOM_CONC_FE_GEN	(0x0002F160)
NLS_NOM_EMFC_O2EI	(0x0401052C)
NLS_NOM_EXTRACT_O2_INDEX	(0x0002F875)
NLS_NOM_EMFC_sFIO2	(0x04018010)
NLS_NOM_SETT_VENT_CONC_AWAY_O2_INSP	(0x04027498)
NLS_NOM_EMFC_sAgent	(0x04018178)
NLS_NOM_SETT_CONC_AWAY_AGENT_TYPE	(0x0402F8E0)
NLS_NOM_EMFC_TFI	(0x040111A8)
NLS_NOM_VOL_FLUID_THORAC_INDEX	(0x0002F8C6)
NLS_NOM_EMFC_LT_AL	(0x040107E0)
NLS_NOM_EEG_PWR_SPEC_ALPHA_ABS_LEFT	(0x0002F855)
NLS_NOM_EMFC_Rf_aVF	(0x04010748)
NLS_NOM_ECG_AMPL_ST_BASELINE_AVF	(0x0002F450)
NLS_NOM_EMFC_RRmech	(0x04010850)
NLS_NOM_VENT_RESP_RATE	(0x00025022)
NLS_NOM_EMFC_ESR	(0x0401064C)

NLS_NOM_ES_RATE	(0x0002F17C)
NLS_NOM_EMFC_Rf_aVL	(0x04010744)
NLS_NOM_ECG_AMPL_ST_BASELINE_AVL	(0x0002F44F)
NLS_NOM_EMFC_BPAPPL	(0x040180BC)
NLS_NOM_SETT_VENT_PRESS_AWAY_BIPAP_LOW	(0x0402F92A)
NLS_NOM_EMFC_sO2Cal	(0x040180D8)
NLS_NOM_SETT_VENT_O2_CAL_MODE	(0x0402F926)
NLS_NOM_EMFC_aPTTWB	(0x04010E14)
NLS_NOM_TIME_PD_aPTT_WB	(0x0002F18D)
NLS_NOM_EMFC_HALLev	(0x0401087C)
NLS_NOM_VOL_LVL_LIQUID_BOTTLE_HALOTH	(0x0002F8CA)
NLS_NOM_EMFC_RT_PCT_DL	(0x04010814)
NLS_NOM_EEG_PWR_SPEC_DELTA_REL_RIGHT	(0x0002F868)
NLS_NOM_EMFC_Pat_T	(0x04010B54)
NLS_NOM_TEMP_BODY	(0x00024B5C)
NLS_NOM_EMFC_sEnSgh	(0x04018040)
NLS_NOM_SETT_VENT_MODE_SIGH	(0x0402F923)
NLS_NOM_EMFC_sPStat	(0x0401A028)
NLS_NOM_SETT_PUMP_STATUS	(0x0402F8FE)
NLS_NOM_EMFC_BSA_D	(0x04010440)
NLS_NOM_AREA_BODY_SURFACE_ACTUAL_DUBOIS	(0x0002F813)
NLS_NOM_EMFC_Field3	(0x04010AD0)
NLS_NOM_SETT_FIELD3	(0x0402F95B)
NLS_NOM_EMFC_VCO2ti	(0x040111C4)
NLS_NOM_FLOW_CO2_PROD_RESP_TIDAL	(0x0002F882)
NLS_NOM_EMFC_EDV	(0x04010534)
NLS_NOM_VOL_VENT_L_END_DIA	(0x00024C00)
NLS_NOM_EMFC_highTV	(0x0401A034)
NLS_NOM_SETT_VENT_VOL_TIDAL_LIMIT_HI	(0x0402F94D)
NLS_NOM_EMFC_PVcP	(0x0401046C)
NLS_NOM_VENT_PRESS_AWAY_PV	(0x0002F8BC)
NLS_NOM_EMFC_Tpat	(0x04010A38)
NLS_NOM_TEMP_BODY	(0x00024B5C)
NLS_NOM_EMFC_sRisTi	(0x04018284)
NLS_NOM_SETT_VENT_TIME_PD_RAMP	(0x0402F8BD)
NLS_NOM_EMFC_U_PER_SCr	(0x0401019C)
NLS_NOM_RATIO_CONC_URINE_CREA_SER	(0x0002F892)
NLS_NOM_EMFC_BSI	(0x04011198)
NLS_NOM_EEG_BURST_SUPPRN_INDEX	(0x0002F840)
NLS_NOM_EMFC_P4_SYS	(0x0401003D)
NLS_NOM_PRESS_GEN_4_SYS	(0x0002F0B1)
NLS_NOM_EMFC_sPin	(0x04018128)
NLS_NOM_SETT_PRESS_AWAY_INSP	(0x04025108)

NLS_NOM_EMFC_BE_B_CALC	(0x04010AC0)
NLS_NOM_BASE_EXCESS_BLD_ART_CALC	(0x0002F817)
NLS_NOM_EMFC_i_eAGT	(0x040106A0)
NLS_NOM_VENT_CONC_AWAY_AGENT_DELTA	(0x0002F8B2)
NLS_NOM_EMFC_UrDens	(0x04010BC0)
NLS_NOM_FLUID_DENS_URINE	(0x0002F19D)
NLS_NOM_EMFC_U_PER_Cre_CALC	(0x04010AE4)
NLS_NOM_RATIO_CONC_URINE_CREA_CALC	(0x0002F891)
NLS_NOM_EMFC_TVex	(0x040106B4)
NLS_NOM_VOL_AWAY_EXP_TIDAL	(0x0002F0E1)
NLS_NOM_EMFC_MCH	(0x040105D8)
NLS_NOM_HB_CORP_MEAN	(0x0002F885)
NLS_NOM_EMFC_Cartrg	(0x04010AB0)
NLS_NOM_SETT_CARTRG	(0x0402F957)
NLS_NOM_EMFC_SaO2	(0x04010548)
NLS_NOM_SAT_O2_ART	(0x00024B34)
NLS_NOM_EMFC_P8_DIA	(0x0401040E)
NLS_NOM_PRESS_GEN_8_DIA	(0x0002F402)
NLS_NOM_EMFC_SO2_r	(0x040111B8)
NLS_NOM_SAT_O2_RIGHT	(0x0002F89E)
NLS_NOM_EMFC_RT_MDF	(0x04010830)
NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_DOM_MEAN_RIGHT	(0x0002F84A)
NLS_NOM_EMFC_Lact	(0x04010AE8)
NLS_NOM_CONC_LACT	(0x0002F174)
NLS_NOM_EMFC_GasCar	(0x040181DC)
NLS_NOM_SETT_VENT_GAS_CARRIER	(0x0402F91F)
NLS_NOM_EMFC_sVolAl	(0x04018158)
NLS_NOM_SETT_VENT_VOL_AWAY_AL_ONOFF	(0x0402F947)
NLS_NOM_EMFC_dBili	(0x04010598)
NLS_NOM_CONC_BILI_DIRECT	(0x0002F17A)
NLS_NOM_EMFC_fgAGT	(0x04010520)
NLS_NOM_FLOW_AWAY_AGENT	(0x0002F876)
NLS_NOM_EMFC_sTrig	(0x04018014)
NLS_NOM_SETT_TRIG_LEVEL	(0x00000000)
NLS_NOM_EMFC_sVmax	(0x04018150)
NLS_NOM_SETT_VENT_VOL_LIMIT_AL_HI_ONOFF	(0x0402F949)
NLS_NOM_EMFC_P3	(0x04010038)
NLS_NOM_PRESS_GEN_3	(0x0002F0AC)
NLS_NOM_EMFC_BagVol	(0x04010CFC)
NLS_NOM_VOL_URINE_COL	(0x00026830)
NLS_NOM_EMFC_PvO2_ADJ	(0x04010A68)
NLS_NOM_CONC_PO2_VEN_ADJ	(0x0002F83E)
NLS_NOM_EMFC_ALT	(0x04010BF0)
NLS_NOM_CONC_GPT	(0x0002F187)

NLS_NOM_EMFC_BP_SYS	(0x04010889)
NLS_NOM_PRESS_BLD_SYS	(0x00024A01)
NLS_NOM_EMFC_P7_DIA	(0x0401040A)
NLS_NOM_PRESS_GEN_7_DIA	(0x0002F3FE)
NLS_NOM_EMFC_liPVAT	(0x0401A010)
NLS_NOM_SETT_APNEA_ALARM_DELAY_PV	(0x0402F8DA)
NLS_NOM_EMFC_T1	(0x04010064)
NLS_NOM_TEMP_GEN_1	(0x0002F0C7)
NLS_NOM_EMFC_CH2O	(0x04010118)
NLS_NOM_FREE_WATER_CLR	(0x0002F884)
NLS_NOM_EMFC_r	(0x04010E80)
NLS_NOM_AWAY_CORR_COEF	(0x0002F814)
NLS_NOM_EMFC_RC	(0x04010644)
NLS_NOM_RET_CNT	(0x0002F16A)
NLS_NOM_EMFC_SpAWRR	(0x04010510)
NLS_NOM_AWAY_RESP_RATE_SPONT	(0x0002F815)
NLS_NOM_EMFC_sMV	(0x040180B0)
NLS_NOM_SETT_VOL_MINUTE_AWAY	(0x04025148)
NLS_NOM_EMFC_sPincR	(0x0401814C)
NLS_NOM_SETT_VENT_AWAY_PRESS_RATE_INCREASE	(0x0402F912)
NLS_NOM_EMFC_MCHC	(0x040105DC)
NLS_NOM_CONC_HB_CORP_MEAN	(0x0002F82C)
NLS_NOM_EMFC_CHE	(0x040105F8)
NLS_NOM_CONC_CHE	(0x0002F182)
NLS_NOM_EMFC_P4	(0x0401003C)
NLS_NOM_PRESS_GEN_4	(0x0002F0B0)
NLS_NOM_EMFC_WBC	(0x040105C8)
NLS_NOM_WB_CNT	(0x0002F168)
NLS_NOM_EMFC_TOFcnt	(0x04010DAC)
NLS_NOM_TRAIN_OF_FOUR_CNT	(0x0002F8AB)
NLS_NOM_EMFC_HGB_CALC	(0x04010A34)
NLS_NOM_CONC_HB_ART_CALC	(0x0002F82B)
NLS_NOM_EMFC_CO_Hb	(0x04010628)
NLS_NOM_CONC_HB_CO_GEN	(0x00027180)
NLS_NOM_EMFC_GEF	(0x040111E4)
NLS_NOM_FRACT_EJECT	(0x0002F105)
NLS_NOM_EMFC_sExpTi	(0x040180E8)
NLS_NOM_SETT_VENT_TIME_PD_EXP	(0x0402F93F)
NLS_NOM_EMFC_sfgFl	(0x040181B8)
NLS_NOM_SETT_FLOW_AWAY_TOT	(0x0402F881)
NLS_NOM_EMFC_SerGlo	(0x040105BC)
NLS_NOM_CONC_GLO_SER	(0x0002F829)
NLS_NOM_EMFC_AnGap_CALC	(0x04010AA8)

NLS_NOM_CONC_AN_GAP_CALC	(0x0002F1A1)
NLS_NOM_EMFC_cktO2	(0x040106A8)
NLS_NOM_VENT_CONC_AWAY_O2_CIRCUIT	(0x0002F8B8)
NLS_NOM_EMFC_IUP_SYS	(0x04010055)
NLS_NOM_PRESS_INTRA_UTERAL_SYS	(0x0002F0D9)
NLS_NOM_EMFC_Field2	(0x04010ACC)
NLS_NOM_SETT_FIELD2	(0x0402F95A)
NLS_NOM_EMFC_AWV	(0x04010668)
NLS_NOM_VOL_AWAY	(0x0002F0DF)
NLS_NOM_EMFC_P3_MEAN	(0x0401003B)
NLS_NOM_PRESS_GEN_3_MEAN	(0x0002F0AF)
NLS_NOM_EMFC_BagWgt	(0x04010BB8)
NLS_NOM_WEIGHT_URINE_COL	(0x0002F8D3)
NLS_NOM_EMFC_O2_MANUAL	(0x04010AD4)
NLS_NOM_CONC_AWAY_O2	(0x00025164)
NLS_NOM_EMFC_i_eISO	(0x04010694)
NLS_NOM_VENT_CONC_AWAY_ISOFL_DELTA	(0x0002F8B6)
NLS_NOM_EMFC_P6_DIA	(0x04010406)
NLS_NOM_PRESS_GEN_6_DIA	(0x0002F3FA)
NLS_NOM_EMFC_iCa_N_CALC	(0x04011114)
NLS_NOM_CONC_CA_GEN_NORM_CALC	(0x0002F823)
NLS_NOM_EMFC_BEecf_CALC	(0x04010AA4)
NLS_NOM_CONC_BASE_EXCESS_ECF_CALC	(0x0002F821)
NLS_NOM_EMFC_sATV	(0x04018028)
NLS_NOM_SETT_VOL_AWAY_TIDAL_APNEA	(0x0402F951)
NLS_NOM_EMFC_pH_ADJ	(0x04010A48)
NLS_NOM_CONC_PH_GEN_ADJ	(0x0002F838)
NLS_NOM_EMFC_P2_DIA	(0x04010036)
NLS_NOM_PRESS_GEN_2_DIA	(0x0002F0AA)
NLS_NOM_EMFC_sSghNr	(0x04018024)
NLS_NOM_SETT_VENT_SIGH_MULT_RATE	(0x0402F93B)
NLS_NOM_EMFC_RT_TH	(0x04010828)
NLS_NOM_EEG_PWR_SPEC_THETA_ABS_RIGHT	(0x0002F86A)
NLS_NOM_EMFC_sfmax	(0x0401820C)
NLS_NOM_SETT_VENT_RESP_RATE_LIMIT_HI_PANT	(0x0402F937)
NLS_NOM_EMFC_UrGlc	(0x04010594)
NLS_NOM_CONC_GLU_URINE	(0x0002F19F)
NLS_NOM_EMFC_PTTrat	(0x04010E1C)
NLS_NOM_RATIO_TIME_PD_PTT	(0x0002F896)
NLS_NOM_EMFC_sfgHAL	(0x040181A4)
NLS_NOM_SETT_CONC_AWAY_HALOTH	(0x040251E0)
NLS_NOM_EMFC_sAPVI	(0x0401808C)
NLS_NOM_SETT_RATIO_IE_INSP_PV_APNEA	(0x0402F903)

NLS_NOM_EMFC_PO2_ADJ	(0x04010A60)
NLS_NOM_CONC_PO2_GEN_ADJ	(0x0002F83D)
NLS_NOM_EMFC_PcO2	(0x04010A5C)
NLS_NOM_CONC_PO2_CAP	(0x0002F15A)
NLS_NOM_EMFC_SerCl	(0x040105B0)
NLS_NOM_CONC_CHLOR_SER	(0x0002F15F)
NLS_NOM_EMFC_UrVol	(0x040101BC)
NLS_NOM_VOL_URINE_BAL_PD	(0x00026824)
NLS_NOM_EMFC_BP_DIA	(0x0401088A)
NLS_NOM_PRESS_BLD_DIA	(0x00024A02)
NLS_NOM_EMFC_L_II	(0x04010780)
NLS_NOM_ECG_ELEC_POTL_II	(0x00020102)
NLS_NOM_EMFC_DET	(0x04010B60)
NLS_NOM_SETT_TEMP	(0x04024B48)
NLS_NOM_EMFC_SerK	(0x040105AC)
NLS_NOM_CONC_K_SER	(0x0002F82F)
NLS_NOM_EMFC_FeNa	(0x0401012C)
NLS_NOM_FRACT_EXCR_NA	(0x0002F194)
NLS_NOM_EMFC_sPmax	(0x040180E0)
NLS_NOM_SETT_VENT_PRESS_AWAY_INSP_MAX	(0x0402F8BB)
NLS_NOM_EMFC_BPAPTL	(0x040180C4)
NLS_NOM_SETT_VENT_TIME_PD_BIPAP_LOW	(0x0402F93E)
NLS_NOM_EMFC_PT_WB	(0x04010E20)
NLS_NOM_TIME_PD_PT_WB	(0x0002F18F)
NLS_NOM_EMFC_sCircl	(0x040181C8)
NLS_NOM_SETT_VENT_CIRCUIT_TYPE	(0x0402F913)
NLS_NOM_EMFC_LSCALE	(0x04010808)
NLS_NOM_EEG_ELEC_POTL_CRTX_GAIN_LEFT	(0x0002F841)
NLS_NOM_EMFC_AccVol	(0x04010680)
NLS_NOM_VOL_INFUS_ACTUAL_TOTAL	(0x000268FC)
NLS_NOM_EMFC_sBkgFl	(0x04018190)
NLS_NOM_SETT_VENT_AWAY_FLOW_BACKGROUND	(0x0402F90F)
NLS_NOM_EMFC_RT_DL	(0x04010824)
NLS_NOM_EEG_PWR_SPEC_DELTA_ABS_RIGHT	(0x0002F864)
NLS_NOM_EMFC_fgDES	(0x04010854)
NLS_NOM_FLOW_AWAY_DESFL	(0x0002F878)
NLS_NOM_EMFC_SerMg	(0x040105A4)
NLS_NOM_CONC_MG_SER	(0x0002F15C)
NLS_NOM_EMFC_AWVex	(0x04010794)
NLS_NOM_VOL_AWAY_EXP	(0x0002F8C1)
NLS_NOM_EMFC_sPltTi	(0x04018018)
NLS_NOM_SETT_TIME_PD_RESP_PLAT	(0x0402F0FF)
NLS_NOM_EMFC_RT_BE	(0x04010820)
NLS_NOM_EEG_PWR_SPEC_BETA_ABS_RIGHT	(0x0002F85C)

NLS_NOM_EMFC_UrpH	(0x04010584)
NLS_NOM_CONC_PH_URINE	(0x00027064)
NLS_NOM_EMFC_T1_T2	(0x040100AC)
NLS_NOM_TEMP_DIFF	(0x0002E018)
NLS_NOM_EMFC_Patm	(0x040106AC)
NLS_NOM_PRESS_AIR_AMBIENT	(0x0002F06B)
NLS_NOM_EMFC_sPVcP	(0x04018064)
NLS_NOM_SETT_PRESS_AWAY_INSP	(0x04025108)
NLS_NOM_EMFC_sARR	(0x0401802C)
NLS_NOM_SETT_AWAY_RESP_RATE_APNEA	(0x0402F8DE)
NLS_NOM_EMFC_BUN_PER_cr	(0x04010110)
NLS_NOM_RATIO_BUN_CREA	(0x0002F88F)
NLS_NOM_EMFC_SerPro	(0x040105C0)
NLS_NOM_CONC_PROT_SER	(0x0002F178)
NLS_NOM_EMFC_HbF	(0x0401062C)
NLS_NOM_CONC_HB_FETAL	(0x0002F165)
NLS_NOM_EMFC_i_eDES	(0x0401069C)
NLS_NOM_VENT_CONC_AWAY_DESFL_DELTA	(0x0002F8B3)
NLS_NOM_EMFC_T2	(0x04010068)
NLS_NOM_TEMP_GEN_2	(0x0002F0C8)
NLS_NOM_EMFC_loPEEP	(0x0401A004)
NLS_NOM_VENT_PRESS_AWAY_END_EXP_POS_LIMIT_LO	(0x0002F8BA)
NLS_NOM_EMFC_TFC	(0x040111A4)
NLS_NOM_VOL_FLUID_THORAC	(0x0002F8C5)
NLS_NOM_EMFC_Length	(0x04010420)
NLS_NOM_BIRTH_LENGTH	(0x0002F818)
NLS_NOM_EMFC_sfgISO	(0x0401819C)
NLS_NOM_SETT_CONC_AWAY_ISOFL	(0x040251E8)
NLS_NOM_EMFC_i_eSEV	(0x04010698)
NLS_NOM_VENT_CONC_AWAY_SEVOFL_DELTA	(0x0002F8B9)
NLS_NOM_EMFC_RVrat	(0x04010E84)
NLS_NOM_RATIO_AWAY_RATE_VOL_AWAY	(0x0002F88E)
NLS_NOM_EMFC_FIO2_MANUAL	(0x04010ABC)
NLS_NOM_VENT_CONC_AWAY_O2_INSP	(0x00027498)
NLS_NOM_EMFC_tCO2	(0x04010588)
NLS_NOM_CONC_CO2_TOT	(0x0002F825)
NLS_NOM_EMFC_sVolas	(0x040181F4)
NLS_NOM_SETT_VENT_VOL_AWAY_ASSIST	(0x0402F948)
NLS_NOM_EMFC_REF	(0x04010530)
NLS_NOM_RIGHT_HEART_FRACT_EJECT	(0x0002F89B)
NLS_NOM_EMFC_RiseTi	(0x04010550)
NLS_NOM_VENT_TIME_PD_RAMP	(0x0002F8BD)
NLS_NOM_EMFC_sSghTV	(0x04018020)

```
NLS_NOM_SETT_VENT_VOL_TIDAL_SIGH                            (0x0402F8C0)

NLS_NOM_EMFC_RemTi                                          (0x04010DBC)
NLS_NOM_TIME_PD_EVOK_REMAIN                                 (0x0002F8A0)

NLS_NOM_EMFC_RT_EEG                                         (0x0401082C)
NLS_NOM_EEG_ELEC_POTL_CRTX_RIGHT                            (0x0002F846)

NLS_NOM_EMFC_TT                                             (0x040105E8)
NLS_NOM_TIME_PD_THROMBIN                                    (0x0002F191)

NLS_NOM_EMFC_inPkFl                                         (0x04010674)
NLS_NOM_FLOW_AWAY_INSP_MAX                                  (0x000250DD)

NLS_NOM_EMFC_PaCO2_ADJ                                      (0x04010A80)
NLS_NOM_CONC_PCO2_ART_ADJ                                   (0x0002F832)

NLS_NOM_EMFC_sMMV                                           (0x0401811C)
NLS_NOM_SETT_VENT_VOL_MINUTE_AWAY_MAND                      (0x040251CC)

NLS_NOM_EMFC_RT_PCT_TH                                      (0x04010818)
NLS_NOM_EEG_PWR_SPEC_THETA_REL_RIGHT                        (0x0002F86E)

NLS_NOM_EMFC_sPVE                                           (0x04018088)
NLS_NOM_SETT_RATIO_IE_EXP_PV                                (0x0402F900)

NLS_NOM_EMFC_LT_BE                                          (0x040107E4)
NLS_NOM_EEG_PWR_SPEC_BETA_ABS_LEFT                          (0x0002F85B)

NLS_NOM_EMFC_sAADel                                         (0x0401813C)
NLS_NOM_SETT_APNEA_ALARM_DELAY                              (0x0402F8D9)

NLS_NOM_EMFC_aPTTPE                                         (0x04010E18)
NLS_NOM_TIME_PD_aPTT_PE                                     (0x0002F18E)

NLS_NOM_EMFC_sIPPV                                          (0x040180A0)
NLS_NOM_SETT_VENT_RESP_RATE_MODE_PPV_INTERMIT_PAP           (0x0402F939)

NLS_NOM_EMFC_P2_MEAN                                        (0x04010037)
NLS_NOM_PRESS_GEN_2_MEAN                                    (0x0002F0AB)

NLS_NOM_EMFC_iCa_N                                          (0x04010E88)
NLS_NOM_CONC_CA_GEN_NORM                                    (0x0002F822)

NLS_NOM_EMFC_sO2Mon                                         (0x040180D4)
NLS_NOM_SETT_VENT_ANALY_CONC_GAS_O2_MODE                    (0x0402F90E)

NLS_NOM_EMFC_P6_SYS                                         (0x04010405)
NLS_NOM_PRESS_GEN_6_SYS                                     (0x0002F3F9)

NLS_NOM_EMFC_DESLev                                         (0x04010880)
NLS_NOM_VOL_LVL_LIQUID_BOTTLE_DESFL                         (0x0002F8C8)

NLS_NOM_EMFC_U_PER_POsm                                     (0x04010198)
NLS_NOM_RATIO_URINE_SER_OSM                                 (0x0002F898)

NLS_NOM_EMFC_RT_TP                                          (0x04010840)
NLS_NOM_EEG_PWR_SPEC_TOT_RIGHT                              (0x0002F872)

NLS_NOM_EMFC_NsLoss                                         (0x040101D4)
NLS_NOM_NSLOSS                                              (0x0002F16D)

NLS_NOM_EMFC_lowMV                                          (0x0401A018)
NLS_NOM_SETT_VENT_VOL_MINUTE_AWAY_LIMIT_LO                  (0x0402F94C)
```

NLS_NOM_EMFC_PTC	(0x04010DB8)
NLS_NOM_PTC_CNT	(0x0002F88B)
NLS_NOM_EMFC_sCMV	(0x04018114)
NLS_NOM_SETT_VENT_MODE_MAND_CTS_ONOFF	(0x0402F922)
NLS_NOM_EMFC_BP	(0x04010888)
NLS_NOM_PRESS_BLD	(0x00024A00)
NLS_NOM_EMFC_sChrge	(0x04018200)
NLS_NOM_SETT_EVOK_CHARGE	(0x0402F8E6)
NLS_NOM_EMFC_ESV	(0x04010538)
NLS_NOM_VOL_VENT_L_END_SYS	(0x00024C04)
NLS_NOM_EMFC_sNeblr	(0x04018044)
NLS_NOM_SETT_VENT_NEBULIZER_MODE	(0x0402F925)
NLS_NOM_EMFC_L_III	(0x04010784)
NLS_NOM_ECG_ELEC_POTL_III	(0x0002013D)
NLS_NOM_EMFC_i_eENF	(0x04010690)
NLS_NOM_VENT_CONC_AWAY_ENFL_DELTA	(0x0002F8B4)
NLS_NOM_EMFC_EDVI	(0x0401053C)
NLS_NOM_VOL_VENT_L_END_DIA_INDEX	(0x0002F8D0)
NLS_NOM_EMFC_RSBI	(0x04010EA0)
NLS_NOM_BREATH_RAPID_SHALLOW_INDEX	(0x0002F819)
NLS_NOM_EMFC_UrKEx	(0x040101A8)
NLS_NOM_CONC_K_URINE_EXCR	(0x0002F198)
NLS_NOM_EMFC_Twitch	(0x04010DB4)
NLS_NOM_TWITCH_AMPL	(0x0002F8AC)
NLS_NOM_EMFC_IUP_MEAN	(0x04010057)
NLS_NOM_PRESS_INTRA_UTERAL_MEAN	(0x0002F0DB)
NLS_NOM_EMFC_SerCK	(0x040105FC)
NLS_NOM_CONC_CREA_KIN_SER	(0x0002F180)
NLS_NOM_EMFC_alphaA	(0x040105F4)
NLS_NOM_CONC_ALPHA_AMYLASE	(0x0002F186)
NLS_NOM_EMFC_PT_PE	(0x04010E24)
NLS_NOM_TIME_PD_PT_PE	(0x0002F190)
NLS_NOM_EMFC_ExpTi	(0x0401066C)
NLS_NOM_TIME_PD_EXP	(0x0002F8A1)
NLS_NOM_EMFC_sPtCat	(0x04018164)
NLS_NOM_SETT_PAT_TYPE	(0x0402F8F6)
NLS_NOM_EMFC_fgENF	(0x04010860)
NLS_NOM_FLOW_AWAY_ENFL	(0x0002F879)
NLS_NOM_EMFC_tBili	(0x0401058C)
NLS_NOM_CONC_BILI_TOT	(0x0002F177)
NLS_NOM_EMFC_UrUrea	(0x04010580)
NLS_NOM_CONC_UREA_URINE	(0x0002F195)
NLS_NOM_EMFC_L_aVR	(0x04010788)
NLS_NOM_ECG_ELEC_POTL_AVR	(0x0002013E)

NLS_NOM_EMFC_P2	(0x04010034)
NLS_NOM_PRESS_GEN_2	(0x0002F0A8)
NLS_NOM_EMFC_LDH	(0x04010638)
NLS_NOM_CONC_LDH	(0x0002F17B)
NLS_NOM_EMFC_sTrVol	(0x04018138)
NLS_NOM_SETT_VENT_VOL_LUNG_TRAPD	(0x040251B8)
NLS_NOM_EMFC_tProt	(0x04010634)
NLS_NOM_CONC_PROT_TOT	(0x0002F179)
NLS_NOM_EMFC_sOxiAl	(0x04018168)
NLS_NOM_SETT_PULS_OXIM_SAT_O2_AL_ONOFF	(0x0402F8FD)
NLS_NOM_EMFC_B_PER_Cre_CALC	(0x04010AE0)
NLS_NOM_RATIO_CONC_BLD_UREA_NITROGEN_CREA_CALC	(0x0002F890)
NLS_NOM_EMFC_HFMVin	(0x040106D8)
NLS_NOM_VOL_MINUTE_AWAY_INSP_HFV	(0x0002F8CD)
NLS_NOM_EMFC_sTlow	(0x040181E4)
NLS_NOM_SETT_VENT_TIME_PD_EXP_APRV	(0x0402F940)
NLS_NOM_EMFC_TOF2	(0x04010DC4)
NLS_NOM_TRAIN_OF_FOUR_2	(0x0002F8A8)
NLS_NOM_EMFC_Rf_III	(0x0401073C)
NLS_NOM_ECG_AMPL_ST_BASELINE_III	(0x0002F44D)
NLS_NOM_EMFC_sGasPr	(0x040181C0)
NLS_NOM_SETT_VENT_GAS_PROBE_POSN	(0x0402F920)
NLS_NOM_EMFC_Met_Hb	(0x04010630)
NLS_NOM_CONC_HB_MET_GEN	(0x0002717C)
NLS_NOM_EMFC_P7_SYS	(0x04010409)
NLS_NOM_PRESS_GEN_7_SYS	(0x0002F3FD)
NLS_NOM_EMFC_L_V5	(0x04010774)
NLS_NOM_ECG_ELEC_POTL_V5	(0x00020107)
NLS_NOM_EMFC_T3	(0x04010410)
NLS_NOM_TEMP_GEN_3	(0x0002F0C9)
NLS_NOM_EMFC_AGTs	(0x04010CE4)
NLS_NOM_CONC_AWAY_AGENT_SEC	(0x0002F820)
NLS_NOM_EMFC_sPVinT	(0x04018068)
NLS_NOM_SETT_VENT_TIME_PD_INSP_PV	(0x0402F943)
NLS_NOM_EMFC_PatID	(0x04010B68)
NLS_NOM_PAT_ID	(0x0002F88A)
NLS_NOM_EMFC_Rf_V2	(0x04010750)
NLS_NOM_ECG_AMPL_ST_BASELINE_V2	(0x0002F414)
NLS_NOM_EMFC_Model	(0x04018110)
NLS_NOM_ID_MODEL	(0x0002F887)
NLS_NOM_EMFC_MinAwP	(0x0401050C)
NLS_NOM_PRESS_AWAY_MIN	(0x000250F2)
NLS_NOM_EMFC_LT_DL	(0x040107E8)

NLS_NOM_EEG_PWR_SPEC_DELTA_ABS_LEFT	(0x0002F863)
NLS_NOM_EMFC_tSerCa	(0x040105A0)
NLS_NOM_CONC_tCA_SER	(0x0002F15D)
NLS_NOM_EMFC_ScO2_CALC	(0x04010A9C)
NLS_NOM_SAT_O2_CAP_CALC	(0x0002F1A0)
NLS_NOM_EMFC_ECTOP	(0x04010090)
NLS_NOM_ECG_STAT_ECT	(0x0002D006)
NLS_NOM_EMFC_sFlCal	(0x04018154)
NLS_NOM_SETT_FLOW_CAL_MODE	(0x0402F8F1)
NLS_NOM_EMFC_L_V3	(0x0401076C)
NLS_NOM_ECG_ELEC_POTL_V3	(0x00020105)
NLS_NOM_EMFC_RHYTHM	(0x0401008C)
NLS_NOM_ECG_STAT_RHY	(0x0002D007)
NLS_NOM_EMFC_ACI	(0x040111AC)
NLS_NOM_OUTPUT_CARD_INDEX_ACCEL	(0x0002F889)
NLS_NOM_EMFC_P7_MEAN	(0x0401040B)
NLS_NOM_PRESS_GEN_7_MEAN	(0x0002F3FF)
NLS_NOM_EMFC_sIMV	(0x040180A4)
NLS_NOM_SETT_VENT_RESP_RATE_MODE_MAND_INTERMITT	(0x0402F938)
NLS_NOM_EMFC_SerAlb	(0x040105B4)
NLS_NOM_CONC_ALB_SER	(0x0002F163)
NLS_NOM_EMFC_Pmin	(0x0401067C)
NLS_NOM_PRESS_AWAY_MIN	(0x000250F2)
NLS_NOM_EMFC_pHa_ADJ	(0x04010A4C)
NLS_NOM_CONC_PH_ART_ADJ	(0x0002F836)
NLS_NOM_EMFC_sHFVRR	(0x04018108)
NLS_NOM_SETT_AWAY_RESP_RATE_HFV	(0x0402F8DF)
NLS_NOM_EMFC_sPWave	(0x0401803C)
NLS_NOM_SETT_AWAY_PRESS_PATTERN	(0x0402F8DC)
NLS_NOM_EMFC_sfgAGT	(0x04018198)
NLS_NOM_SETT_FLOW_AWAY_AGENT	(0x0402F876)
NLS_NOM_EMFC_BPAPPH	(0x040180C0)
NLS_NOM_SETT_VENT_PRESS_AWAY_BIPAP_HIGH	(0x0402F929)
NLS_NOM_EMFC_sAFIO2	(0x04018034)
NLS_NOM_SETT_VENT_CONC_AWAY_O2_INSP_APNEA	(0x0402F917)
NLS_NOM_EMFC_P6	(0x04010404)
NLS_NOM_PRESS_GEN_6	(0x0002F3F8)
NLS_NOM_EMFC_PTrat	(0x04010E28)
NLS_NOM_RATIO_TIME_PD_PT	(0x0002F895)
NLS_NOM_EMFC_IUP_DIA	(0x04010056)
NLS_NOM_PRESS_INTRA_UTERAL_DIA	(0x0002F0DA)
NLS_NOM_EMFC_TVin	(0x040106B0)
NLS_NOM_VOL_AWAY_INSP_TIDAL	(0x0002F0E0)

NLS_NOM_EMFC_PtVent	(0x04010BDC)
NLS_NOM_VENT_ACTIVE	(0x0002F8B0)
NLS_NOM_EMFC_LT_PCT_AL	(0x040107D0)
NLS_NOM_EEG_PWR_SPEC_ALPHA_REL_LEFT	(0x0002F859)
NLS_NOM_EMFC_Rdyn	(0x04010480)
NLS_NOM_RES_AWAY_DYN	(0x0002F899)
NLS_NOM_EMFC_sVMode	(0x04018000)
NLS_NOM_SETT_VENT_MODE	(0x0402F921)
NLS_NOM_EMFC_etAGTs	(0x04010CF0)
NLS_NOM_CONC_AWAY_AGENT_ET_SEC	(0x0002F81E)
NLS_NOM_EMFC_pHv_ADJ	(0x04010A50)
NLS_NOM_CONC_PH_VEN_ADJ	(0x0002F839)
NLS_NOM_EMFC_sHum	(0x04018288)
NLS_NOM_SETT_HUMID	(0x0402F103)
NLS_NOM_EMFC_highP	(0x0401A000)
NLS_NOM_SETT_VENT_PRESS_AWAY_LIMIT_HI	(0x0402F930)
NLS_NOM_EMFC_LT_TP	(0x04010804)
NLS_NOM_EEG_PWR_SPEC_TOT_LEFT	(0x0002F871)
NLS_NOM_EMFC_SCreat	(0x04010180)
NLS_NOM_CONC_CREA_SER	(0x0002F827)
NLS_NOM_EMFC_sExpFl	(0x04018134)
NLS_NOM_SETT_FLOW_AWAY_EXP	(0x0402F8EA)
NLS_NOM_EMFC_HFVTV	(0x040106E8)
NLS_NOM_VENT_VOL_TIDAL_HFV	(0x0002F8BF)
NLS_NOM_EMFC_UrCl	(0x040105B8)
NLS_NOM_CONC_CHLOR_URINE	(0x0002F19A)
NLS_NOM_EMFC_fgSEV	(0x04010858)
NLS_NOM_FLOW_AWAY_SEVOFL	(0x0002F880)
NLS_NOM_EMFC_sPlow	(0x040181EC)
NLS_NOM_SETT_VENT_PRESS_AWAY_EXP_APRV	(0x0402F92D)
NLS_NOM_EMFC_LT_PCT_DL	(0x040107D8)
NLS_NOM_EEG_PWR_SPEC_DELTA_REL_LEFT	(0x0002F867)
NLS_NOM_EMFC_Turine	(0x04010BC4)
NLS_NOM_TEMP_VESICAL	(0x0002F0C4)
NLS_NOM_EMFC_Rf_V1	(0x0401074C)
NLS_NOM_ECG_AMPL_ST_BASELINE_V1	(0x0002F413)
NLS_NOM_EMFC_ENFLev	(0x04010878)
NLS_NOM_VOL_LVL_LIQUID_BOTTLE_ENFL	(0x0002F8C9)
NLS_NOM_EMFC_liATi	(0x0401A00C)
NLS_NOM_SETT_APNEA_ALARM_DELAY	(0x0402F8D9)
NLS_NOM_EMFC_fgHAL	(0x0401085C)
NLS_NOM_FLOW_AWAY_HALOTH	(0x0002F87B)
NLS_NOM_EMFC_AP	(0x040105F0)
NLS_NOM_CONC_AP	(0x0002F185)

NLS_NOM_EMFC_sInsTi	(0x040180E4)
NLS_NOM_SETT_VENT_TIME_PD_INSP	(0x0402F941)
NLS_NOM_EMFC_sThigh	(0x040181E8)
NLS_NOM_SETT_VENT_TIME_PD_INSP_APRV	(0x0402F942)
NLS_NOM_EMFC_sCPAP	(0x040180F4)
NLS_NOM_SETT_PRESS_AWAY_CTS_POS	(0x040250F4)
NLS_NOM_EMFC_sO2Pr	(0x040181C4)
NLS_NOM_SETT_VENT_O2_PROBE_POSN	(0x0402F927)
NLS_NOM_EMFC_loPmax	(0x04018174)
NLS_NOM_SETT_PRESS_AWAY_INSP_MAX_LIMIT_LO	(0x0402F8FB)
NLS_NOM_EMFC_IUP	(0x04010054)
NLS_NOM_PRESS_INTRA_UTERAL	(0x0002F0D8)
NLS_NOM_EMFC_IMV	(0x04010138)
NLS_NOM_VENT_MODE_MAND_INTERMIT	(0x0002D02A)
NLS_NOM_EMFC_sTVap	(0x04018184)
NLS_NOM_SETT_VOL_AWAY_TIDAL_APPLIED	(0x0402F952)
NLS_NOM_EMFC_PVPI	(0x040111F0)
NLS_NOM_PERM_VASC_PULM_INDEX	(0x0002F106)
NLS_NOM_EMFC_OperID	(0x04010AB4)
NLS_NOM_SETT_OPERID	(0x0402F958)
NLS_NOM_EMFC_Ppeak	(0x040106CC)
NLS_NOM_PRESS_AWAY_INSP_MAX	(0x00025109)
NLS_NOM_EMFC_P5_DIA	(0x04010402)
NLS_NOM_PRESS_GEN_5_DIA	(0x0002F3F6)
NLS_NOM_EMFC_sADel	(0x0401817C)
NLS_NOM_SETT_APNEA_ALARM_DELAY	(0x0402F8D9)
NLS_NOM_EMFC_NIF	(0x04010E9C)
NLS_NOM_PRESS_AWAY_NEG_MAX	(0x000250F9)
NLS_NOM_EMFC_SpO2_APER	(0x040100E0)
NLS_NOM_PULS_OXIM_SAT_O2	(0x00024BB8)
NLS_NOM_EMFC_sTVin	(0x040181CC)
NLS_NOM_SETT_VOL_AWAY_INSP_TIDAL	(0x0402F0E0)
NLS_NOM_EMFC_RT_MPF	(0x04010834)
NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_MEDIAN_RIGHT	(0x0002F84C)
NLS_NOM_EMFC_RT_PPF	(0x04010838)
NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_PEAK_RIGHT	(0x0002F850)
NLS_NOM_EMFC_ALP	(0x04010640)
NLS_NOM_CONC_ALP	(0x0002F81D)
NLS_NOM_EMFC_CO2Cal	(0x040181E0)
NLS_NOM_SETT_VENT_CO2_CAL_MODE	(0x0402F914)
NLS_NOM_EMFC_sFlow	(0x040180F8)
NLS_NOM_SETT_VENT_FLOW	(0x0402F91B)
NLS_NOM_EMFC_sAWRR	(0x04018004)

NLS_NOM_SETT_AWAY_RESP_RATE	(0x04025012)
NLS_NOM_EMFC_sHInPr	(0x0401818C)
NLS_NOM_SETT_PRESS_AWAY_INSP_MAX	(0x04025109)
NLS_NOM_EMFC_set_T	(0x040181D0)
NLS_NOM_SETT_TEMP	(0x04024B48)
NLS_NOM_EMFC_BasePr	(0x04010554)
NLS_NOM_VENT_PRESS_AWAY_END_EXP_POS	(0x000251A8)
NLS_NOM_EMFC_SO2_l	(0x040111B4)
NLS_NOM_SAT_O2_LEFT	(0x0002F89D)
NLS_NOM_EMFC_Age	(0x04010BC8)
NLS_NOM_AGE	(0x0002F810)
NLS_NOM_EMFC_CT	(0x04010648)
NLS_NOM_TIME_PD_COAGULATION	(0x0002F192)
NLS_NOM_EMFC_L_V2	(0x04010768)
NLS_NOM_ECG_ELEC_POTL_V2	(0x00020104)
NLS_NOM_EMFC_sO2Suc	(0x04018048)
NLS_NOM_SETT_VENT_O2_SUCTION_MODE	(0x0402F928)
NLS_NOM_EMFC_sTPDel	(0x040180D0)
NLS_NOM_SETT_TACHY_APNEA_DELAY	(0x0402F906)
NLS_NOM_EMFC_Crea	(0x04010ADC)
NLS_NOM_CONC_CREA	(0x0002F173)
NLS_NOM_EMFC_NgInsP	(0x04010484)
NLS_NOM_PRESS_AWAY_NEG_MAX	(0x000250F9)
NLS_NOM_EMFC_P7	(0x04010408)
NLS_NOM_PRESS_GEN_7	(0x0002F3FC)
NLS_NOM_EMFC_MV	(0x040106B8)
NLS_NOM_VOL_MINUTE_AWAY	(0x00025148)
NLS_NOM_EMFC_SEVLev	(0x04010884)
NLS_NOM_VOL_LVL_LIQUID_BOTTLE_SEVOFL	(0x0002F8CC)
NLS_NOM_EMFC_Quick	(0x040105EC)
NLS_NOM_TIME_PD_THROMBOPLAS	(0x0002F193)
NLS_NOM_EMFC_PaFIO2	(0x04010BE0)
NLS_NOM_RATIO_PaO2_FIO2	(0x0002F894)
NLS_NOM_EMFC_pHc	(0x04010A44)
NLS_NOM_CONC_PH_CAP	(0x0002F158)
NLS_NOM_EMFC_ESVI	(0x04010540)
NLS_NOM_VOL_VENT_L_END_SYS_INDEX	(0x0002F8D1)
NLS_NOM_EMFC_Rinsp	(0x04010670)
NLS_NOM_RES_AWAY_INSP	(0x00025128)
NLS_NOM_EMFC_i_eN2O	(0x04010688)
NLS_NOM_VENT_CONC_AWAY_N2O_DELTA	(0x0002F8B7)
NLS_NOM_EMFC_Rf_aVR	(0x04010740)
NLS_NOM_ECG_AMPL_ST_BASELINE_AVR	(0x0002F44E)

NLS_NOM_EMFC_LT_TH	(0x040107EC)
NLS_NOM_EEG_PWR_SPEC_THETA_ABS_LEFT	(0x0002F869)
NLS_NOM_EMFC_RT_SEF	(0x0401083C)
NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_SPECTRAL_EDGE_RIGHT	(0x0002F854)
NLS_NOM_EMFC_RT_PCT_AL	(0x0401080C)
NLS_NOM_EEG_PWR_SPEC_ALPHA_REL_RIGHT	(0x0002F85A)
NLS_NOM_EMFC_Rexp	(0x04010664)
NLS_NOM_RES_AWAY_EXP	(0x00025124)
NLS_NOM_EMFC_P4_MEAN	(0x0401003F)
NLS_NOM_PRESS_GEN_4_MEAN	(0x0002F0B3)
NLS_NOM_EMFC_i_eO2	(0x040106A4)
NLS_NOM_VENT_CONC_AWAY_O2_DELTA	(0x00025168)
NLS_NOM_EMFC_Rf_V4	(0x04010758)
NLS_NOM_ECG_AMPL_ST_BASELINE_V4	(0x0002F416)
NLS_NOM_EMFC_P5_SYS	(0x04010401)
NLS_NOM_PRESS_GEN_5_SYS	(0x0002F3F5)
NLS_NOM_EMFC_PT_INR	(0x04010E2C)
NLS_NOM_PT_INTL_NORM_RATIO	(0x0002F18C)
NLS_NOM_EMFC_Elapse	(0x04010B34)
NLS_NOM_TIME_PD_FROM_LAST_MSMT	(0x0002F8A2)
NLS_NOM_EMFC_ACT	(0x04010E10)
NLS_NOM_TIME_PD_ACT	(0x0002F18A)
NLS_NOM_EMFC_sfgAir	(0x040181B0)
NLS_NOM_SETT_FLOW_AWAY_AIR	(0x0402F877)
NLS_NOM_EMFC_sSilnc	(0x04018080)
NLS_NOM_SETT_AL_SILENCE_ONOFF	(0x0402F8D8)
NLS_NOM_EMFC_TOFrat	(0x04010DB0)
NLS_NOM_RATIO_TRAIN_OF_FOUR	(0x0002F897)
NLS_NOM_EMFC_L_aVL	(0x0401078C)
NLS_NOM_ECG_ELEC_POTL_AVL	(0x0002013F)
NLS_NOM_EMFC_Field1	(0x04010AC8)
NLS_NOM_SETT_FIELD1	(0x0402F959)
NLS_NOM_EMFC_HFTVin	(0x040106E4)
NLS_NOM_VENT_VOL_AWAY_INSP_TIDAL_HFV	(0x0002F8BE)
NLS_NOM_EMFC_SvO2_CALC	(0x04010A98)
NLS_NOM_SAT_O2_VEN_CALC	(0x0002F166)
NLS_NOM_EMFC_AAI	(0x04011194)
NLS_NOM_ELEC_EVOK_POTL_CRTX_ACOUSTIC_AAI	(0x0002F873)
NLS_NOM_EMFC_TVPSV	(0x04010E98)
NLS_NOM_VOL_AWAY_TIDAL_PSV	(0x0002F8C3)
NLS_NOM_EMFC_VPB	(0x04010088)
NLS_NOM_ECG_V_P_C_CNT	(0x00024261)
NLS_NOM_EMFC_sMVDel	(0x04018144)
NLS_NOM_SETT_VOL_MINUTE_ALARM_DELAY	(0x0402F953)

NLS_NOM_EMFC_sCO2Al	(0x04018160)
NLS_NOM_SETT_AWAY_CO2_AL_ONOFF	(0x0402F8DB)
NLS_NOM_EMFC_HFVAmp	(0x0401055C)
NLS_NOM_VENT_AMPL_HFV	(0x0002F8B1)
NLS_NOM_EMFC_lowO2	(0x0401A01C)
NLS_NOM_SETT_VENT_CONC_AWAY_O2_LIMIT_LO	(0x0402F91A)
NLS_NOM_EMFC_BP_MEAN	(0x0401088B)
NLS_NOM_PRESS_BLD_MEAN	(0x00024A03)
NLS_NOM_EMFC_sSenFl	(0x0401805C)
NLS_NOM_SETT_VENT_AWAY_FLOW_SENSE	(0x0402F911)
NLS_NOM_EMFC_sDRate	(0x04018124)
NLS_NOM_SETT_FLOW_FLUID_PUMP	(0x04026858)
NLS_NOM_EMFC_fgISO	(0x04010864)
NLS_NOM_FLOW_AWAY_ISOFL	(0x0002F87C)
NLS_NOM_EMFC_fgAir	(0x040111BC)
NLS_NOM_FLOW_AWAY_AIR	(0x0002F877)
NLS_NOM_EMFC_SaO2_CALC	(0x04010A94)
NLS_NOM_SAT_O2_ART_CALC	(0x0002F164)
NLS_NOM_EMFC_sPVI	(0x04018084)
NLS_NOM_SETT_RATIO_IE_INSP_PV	(0x0402F902)
NLS_NOM_EMFC_Power	(0x04010B5C)
NLS_NOM_HEATING_PWR_INCUBATOR	(0x0002F886)
NLS_NOM_EMFC_sfgDES	(0x040181A8)
NLS_NOM_SETT_CONC_AWAY_DESFL	(0x040251D8)
NLS_NOM_EMFC_i_eHAL	(0x0401068C)
NLS_NOM_VENT_CONC_AWAY_HALOTH_DELTA	(0x0002F8B5)
NLS_NOM_EMFC_sTrgFl	(0x04018148)
NLS_NOM_SETT_VENT_FLOW_INSP_TRIG	(0x0402F91D)
NLS_NOM_EMFC_InsTi	(0x04010E74)
NLS_NOM_TIME_PD_INSP	(0x0002F8A3)
NLS_NOM_EMFC_CrCl	(0x04010124)
NLS_NOM_CONC_CREA_CLR	(0x0002F16C)
NLS_NOM_EMFC_UrNa_PER_K	(0x040101B0)
NLS_NOM_RATIO_CONC_URINE_NA_K	(0x0002F893)
NLS_NOM_EMFC_sCurnt	(0x040181FC)
NLS_NOM_SETT_EVOK_CURR	(0x0402F8E7)
NLS_NOM_EMFC_P3_SYS	(0x04010039)
NLS_NOM_PRESS_GEN_3_SYS	(0x0002F0AD)
NLS_NOM_EMFC_Rf_I	(0x04010734)
NLS_NOM_ECG_AMPL_ST_BASELINE_I	(0x0002F411)
NLS_NOM_EMFC_KCT	(0x04010654)
NLS_NOM_TIME_PD_KAOLIN_CEPHALINE	(0x0002F8A4)
NLS_NOM_EMFC_sPSVrp	(0x04018180)

NLS_NOM_SETT_VENT_TIME_PD_RAMP	(0x0402F8BD)
NLS_NOM_EMFC_P8	(0x0401040C)
NLS_NOM_PRESS_GEN_8	(0x0002F400)
NLS_NOM_EMFC_P2_SYS	(0x04010035)
NLS_NOM_PRESS_GEN_2_SYS	(0x0002F0A9)
NLS_NOM_EMFC_Air_T	(0x04010B58)
NLS_NOM_TEMP_AMBIENT	(0x0002F0C6)
NLS_NOM_EMFC_GPT	(0x04010610)
NLS_NOM_CONC_GPT	(0x0002F187)
NLS_NOM_EMFC_CK_MB	(0x04010600)
NLS_NOM_CONC_CREA_KIN_MB	(0x0002F181)
NLS_NOM_EMFC_P1_DIA	(0x04010032)
NLS_NOM_PRESS_GEN_1_DIA	(0x0002F0A6)
NLS_NOM_EMFC_fgFlow	(0x040111C0)
NLS_NOM_FLOW_AWAY_TOT	(0x0002F881)
NLS_NOM_EMFC_sBasFl	(0x04018058)
NLS_NOM_SETT_VENT_AWAY_FLOW_BASE	(0x0402F910)
NLS_NOM_EMFC_PTT	(0x040105E0)
NLS_NOM_TIME_PD_PTT	(0x0002F8A5)
NLS_NOM_EMFC_sAPVE	(0x04018090)
NLS_NOM_SETT_RATIO_IE_EXP_PV_APNEA	(0x0402F901)
NLS_NOM_EMFC_UrPro	(0x04010620)
NLS_NOM_CONC_PRO_URINE	(0x0002F19B)
NLS_NOM_EMFC_UCreat	(0x040101A0)
NLS_NOM_CONC_CREA_URINE	(0x0002F196)
NLS_NOM_EMFC_sfgENF	(0x040181A0)
NLS_NOM_SETT_CONC_AWAY_ENFL	(0x040251DC)
NLS_NOM_EMFC_SrUrea	(0x040105C4)
NLS_NOM_UREA_SER	(0x0002F8AD)
NLS_NOM_EMFC_PlGain	(0x04010514)
NLS_NOM_PULS_OXIM_PLETH_GAIN	(0x0002F88D)
NLS_NOM_EMFC_pHc_ADJ	(0x04010A54)
NLS_NOM_CONC_PH_CAP_ADJ	(0x0002F837)
NLS_NOM_EMFC_TOF3	(0x04010DC8)
NLS_NOM_TRAIN_OF_FOUR_3	(0x0002F8A9)
NLS_NOM_EMFC_exPkFl	(0x040111CC)
NLS_NOM_FLOW_AWAY_EXP_MAX	(0x000250D9)
NLS_NOM_EMFC_Rf_V3	(0x04010754)
NLS_NOM_ECG_AMPL_ST_BASELINE_V3	(0x0002F415)
NLS_NOM_EMFC_KPLUS	(0x0401065C)
NLS_NOM_CONC_K_GEN	(0x00027110)
NLS_NOM_EMFC_L_I	(0x0401077C)
NLS_NOM_ECG_ELEC_POTL_I	(0x00020101)

NLS_NOM_EMFC_sSghR	(0x0401801C)
NLS_NOM_SETT_VENT_SIGH_RATE	(0x0402F93C)
NLS_NOM_EMFC_BSA_B	(0x0401043C)
NLS_NOM_AREA_BODY_SURFACE_ACTUAL_BOYD	(0x0002F812)
NLS_NOM_EMFC_G_Age	(0x04010428)
NLS_NOM_AGE_GEST	(0x0002F811)
NLS_NOM_EMFC_PlOsm	(0x04010164)
NLS_NOM_PLASMA_OSM	(0x0002F16B)
NLS_NOM_EMFC_fgO2	(0x0401086C)
NLS_NOM_FLOW_AWAY_O2	(0x0002F87F)
NLS_NOM_EMFC_PcO2_ADJ	(0x04010A6C)
NLS_NOM_CONC_PO2_CAP_ADJ	(0x0002F83C)
NLS_NOM_EMFC_DABP	(0x0401054C)
NLS_NOM_VENT_TIME_PD_PPV	(0x00025360)
NLS_NOM_EMFC_sAPVcP	(0x0401806C)
NLS_NOM_SETT_VENT_PRESS_AWAY_PV_APNEA	(0x0402F933)
NLS_NOM_EMFC_sUrTi	(0x040181D4)
NLS_NOM_SETT_URINE_BAL_PD	(0x0402F8AF)
NLS_NOM_EMFC_sEnTP	(0x040180B8)
NLS_NOM_SETT_TACHAPNEA_AL_ONOFF	(0x0402F905)
NLS_NOM_EMFC_DPosP	(0x04010848)
NLS_NOM_VENT_TIME_PD_PPV	(0x00025360)
NLS_NOM_EMFC_sustP	(0x0401A014)
NLS_NOM_SETT_VENT_PRESS_AWAY_SUST_LIMIT_HI	(0x0402F935)
NLS_NOM_EMFC_RRsync	(0x0401084C)
NLS_NOM_RESP_BREATH_ASSIST_CNT	(0x0002F89A)
NLS_NOM_EMFC_sHFVFl	(0x04018104)
NLS_NOM_SETT_FLOW_AWAY_HFV	(0x0402F8EB)
NLS_NOM_EMFC_L_aVF	(0x04010790)
NLS_NOM_ECG_ELEC_POTL_AVF	(0x00020140)
NLS_NOM_EMFC_RT_AL	(0x0401081C)
NLS_NOM_EEG_PWR_SPEC_ALPHA_ABS_RIGHT	(0x0002F856)
NLS_NOM_EMFC_sMode	(0x04018098)
NLS_NOM_SETT_MODE_MSMT	(0x0402F8F5)
NLS_NOM_EMFC_sSPEEP	(0x040180AC)
NLS_NOM_SETT_VENT_PRESS_AWAY_END_EXP_POS_INTERMIT	(0x0402F92C)
NLS_NOM_EMFC_sPhigh	(0x040181F0)
NLS_NOM_SETT_VENT_PRESS_AWAY_INSP_APRV	(0x0402F92E)
NLS_NOM_EMFC_LT_PCT_TH	(0x040107DC)
NLS_NOM_EEG_PWR_SPEC_THETA_REL_LEFT	(0x0002F86D)
NLS_NOM_EMFC_sCycTi	(0x0401809C)
NLS_NOM_SETT_TIME_PD_MSMT	(0x0402F909)
NLS_NOM_EMFC_fgN2O	(0x04010868)
NLS_NOM_FLOW_AWAY_N2O	(0x0002F87E)

NLS_NOM_EMFC_AST	(0x0401063C)
NLS_NOM_CONC_AST	(0x0002F184)
NLS_NOM_EMFC_SpTVex	(0x040106E0)
NLS_NOM_VOL_AWAY_EXP_TIDAL_SPONT	(0x0002F8C2)
NLS_NOM_EMFC_sIE_1	(0x040180EC)
NLS_NOM_SETT_RATIO_IE	(0x04025118)
NLS_NOM_EMFC_P1_MEAN	(0x04010033)
NLS_NOM_PRESS_GEN_1_MEAN	(0x0002F0A7)
NLS_NOM_EMFC_PvCO2_ADJ	(0x04010A84)
NLS_NOM_CONC_PCO2_VEN_ADJ	(0x0002F835)
NLS_NOM_EMFC_TC	(0x04010E7C)
NLS_NOM_AWAY_TC	(0x0002F816)
NLS_NOM_EMFC_P4_DIA	(0x0401003E)
NLS_NOM_PRESS_GEN_4_DIA	(0x0002F0B2)
NLS_NOM_EMFC_P1	(0x04010030)
NLS_NOM_PRESS_GEN_1	(0x0002F0A4)
NLS_NOM_EMFC_hiSghP	(0x0401A008)
NLS_NOM_SETT_VENT_PRESS_AWAY_SIGH_LIMIT_HI	(0x0402F934)
NLS_NOM_EMFC_Rf_V6	(0x04010760)
NLS_NOM_ECG_AMPL_ST_BASELINE_V6	(0x0002F418)
NLS_NOM_EMFC_Diff_X	(0x04010224)
NLS_NOM_TEMP_DIFF	(0x0002E018)
NLS_NOM_EMFC_sMVAl	(0x040180DC)
NLS_NOM_SETT_VOL_MINUTE_AWAY_AL_ONOFF	(0x0402F955)
NLS_NOM_EMFC_P5_MEAN	(0x04010403)
NLS_NOM_PRESS_GEN_5_MEAN	(0x0002F3F7)
NLS_NOM_EMFC_sAPVO2	(0x04018078)
NLS_NOM_SETT_VENT_CONC_AWAY_O2_INSP_PV_APNEA	(0x0402F918)
NLS_NOM_EMFC_Wave	(0x04018170)
NLS_NOM_WAVE_LBL	(0x0002F8D2)
NLS_NOM_EMFC_UrK	(0x040101A4)
NLS_NOM_CONC_K_URINE	(0x0002F197)
NLS_NOM_EMFC_LT_MDF	(0x040107F4)
NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_DOM_MEAN_LEFT	(0x0002F849)
NLS_NOM_EMFC_RRaw	(0x040106C4)
NLS_NOM_VENT_RESP_RATE	(0x00025022)
NLS_NOM_EMFC_sAPVTi	(0x04018074)
NLS_NOM_SETT_VENT_TIME_PD_INSP_PV_APNEA	(0x0402F944)
NLS_NOM_EMFC_HI	(0x040111B0)
NLS_NOM_CARD_CONTRACT_HEATHER_INDEX	(0x0002F81C)
NLS_NOM_EMFC_sPEEP	(0x040180A8)
NLS_NOM_SETT_VENT_PRESS_AWAY_END_EXP_POS	(0x040251A8)
NLS_NOM_EMFC_L_V6	(0x04010778)

NLS_NOM_ECG_ELEC_POTL_V6	(0x00020108)
NLS_NOM_EMFC_COsm	(0x04010120)
NLS_NOM_CREA_OSM	(0x0002F83F)
NLS_NOM_EMFC_ISOLev	(0x04010874)
NLS_NOM_VOL_LVL_LIQUID_BOTTLE_ISOFL	(0x0002F8CB)
NLS_NOM_EMFC_Rf_II	(0x04010738)
NLS_NOM_ECG_AMPL_ST_BASELINE_II	(0x0002F412)
NLS_NOM_EMFC_tUrVol	(0x04010BBC)
NLS_NOM_VOL_URINE_BAL_PD_INSTANT	(0x0002F8CE)
NLS_NOM_EMFC_PcCO2_ADJ	(0x04010A88)
NLS_NOM_CONC_PCO2_CAP_ADJ	(0x0002F833)
NLS_NOM_EMFC_sfgN2O	(0x040181BC)
NLS_NOM_SETT_FLOW_AWAY_N2O	(0x0402F87E)
NLS_NOM_EMFC_UrFl	(0x04010890)
NLS_NOM_FLOW_URINE_INSTANT	(0x0002680C)
NLS_NOM_EMFC_sAPVRR	(0x04018070)
NLS_NOM_SETT_VENT_RESP_RATE_PV_APNEA	(0x0402F93A)
NLS_NOM_EMFC_LT_SEF	(0x04010800)
NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_SPECTRAL_EDGE_LEFT	(0x0002F853)
NLS_NOM_EMFC_Chol	(0x04010618)
NLS_NOM_CONC_CHOLESTEROL	(0x0002F16E)
NLS_NOM_EMFC_L_V1	(0x04010764)
NLS_NOM_ECG_ELEC_POTL_V1	(0x00020103)
NLS_NOM_EMFC_AwN2O	(0x04010518)
NLS_NOM_CONC_AWAY_N2O	(0x000251F0)
NLS_NOM_EMFC_P8_SYS	(0x0401040D)
NLS_NOM_PRESS_GEN_8_SYS	(0x0002F401)
NLS_NOM_EMFC_ICG	(0x040111A0)
NLS_NOM_IMPED_TTHOR_ECG	(0x0002F888)
NLS_NOM_EMFC_HCO3_CALC	(0x04010AA0)
NLS_NOM_CONC_HCO3_GEN_CALC	(0x0002F82E)
NLS_NOM_EMFC_sRRaw	(0x0401812C)
NLS_NOM_SETT_VENT_RESP_RATE	(0x04025022)
NLS_NOM_EMFC_sO2	(0x0401810C)
NLS_NOM_SETT_CONC_AWAY_O2	(0x04025164)
NLS_NOM_EMFC_sTV	(0x04018008)
NLS_NOM_SETT_VOL_AWAY_TIDAL	(0x0402513C)
NLS_NOM_EMFC_PCV	(0x04010650)
NLS_NOM_CONC_HCT_GEN	(0x00027184)
NLS_NOM_EMFC_Pmax	(0x04010678)
NLS_NOM_VENT_PRESS_AWAY_INSP_MAX	(0x0002F8BB)
NLS_NOM_EMFC_LT_PCT_BE	(0x040107D4)
NLS_NOM_EEG_PWR_SPEC_BETA_REL_LEFT	(0x0002F85F)

NLS_NOM_EMFC_sInsFl	(0x04018130)
NLS_NOM_SETT_FLOW_AWAY_INSP	(0x0402F8EC)
NLS_NOM_EMFC_UrVSht	(0x0401088C)
NLS_NOM_VOL_URINE_SHIFT	(0x0002F8CF)
NLS_NOM_EMFC_AGTLev	(0x04010870)
NLS_NOM_VOL_LVL_LIQUID_BOTTLE_AGENT	(0x0002F8C7)
NLS_NOM_EMFC_sPSV	(0x04018038)
NLS_NOM_SETT_VENT_PRESS_AWAY_PV	(0x0402F8BC)
NLS_NOM_EMFC_Urea	(0x04010AB8)
NLS_NOM_CONC_UREA_GEN	(0x0002F172)
NLS_NOM_EMFC_P8_MEAN	(0x0401040F)
NLS_NOM_PRESS_GEN_8_MEAN	(0x0002F403)
NLS_NOM_EMFC_RSCALE	(0x04010844)
NLS_NOM_EEG_ELEC_POTL_CRTX_GAIN_RIGHT	(0x0002F842)
NLS_NOM_EMFC_sRepTi	(0x04018208)
NLS_NOM_SETT_TIME_PD_TRAIN_OF_FOUR	(0x0402F8A6)
NLS_NOM_EMFC_LT_EEG	(0x040107F0)
NLS_NOM_EEG_ELEC_POTL_CRTX_LEFT	(0x0002F845)
NLS_NOM_EMFC_P3_DIA	(0x0401003A)
NLS_NOM_PRESS_GEN_3_DIA	(0x0002F0AE)
NLS_NOM_EMFC_SerPho	(0x040105A8)
NLS_NOM_CONC_P_SER	(0x0002F15E)
NLS_NOM_EMFC_eeFlow	(0x040111D0)
NLS_NOM_FLOW_AWAY_EXP_ET	(0x0002F87A)
NLS_NOM_EMFC_inAGTs	(0x04010CEC)
NLS_NOM_CONC_AWAY_AGENT_INSP_SEC	(0x0002F81F)
NLS_NOM_EMFC_iMg	(0x04010AC4)
NLS_NOM_CONC_MG_ION	(0x0002F15B)
NLS_NOM_EMFC_sFWave	(0x04018120)
NLS_NOM_SETT_VENT_FLOW_PATTERN	(0x0402F91E)
NLS_NOM_EMFC_UrOsm	(0x040101B8)
NLS_NOM_CONC_OSM_URINE	(0x0002F199)
NLS_NOM_EMFC_Paw	(0x040106BC)
NLS_NOM_PRESS_AWAY	(0x000250F0)
NLS_NOM_EMFC_DCO2	(0x040106DC)
NLS_NOM_COEF_GAS_TRAN	(0x000251D4)
NLS_NOM_EMFC_Pmean	(0x040106C0)
NLS_NOM_PRESS_AWAY_INSP_MEAN	(0x0002510B)
NLS_NOM_EMFC_LT_PPF	(0x040107FC)
NLS_NOM_EEG_FREQ_PWR_SPEC_CRTX_PEAK_LEFT	(0x0002F84F)
NLS_NOM_EMFC_lowTV	(0x0401A030)
NLS_NOM_SETT_VENT_VOL_TIDAL_LIMIT_LO	(0x0402F94E)
NLS_NOM_EMFC_PaO2_ADJ	(0x04010A64)
NLS_NOM_CONC_PO2_ART_ADJ	(0x0002F83B)

NLS_NOM_EMFC_sPkFl	(0x0401800C)
NLS_NOM_SETT_FLOW_AWAY_INSP_MAX	(0x040250DD)
NLS_NOM_EMFC_SpPkFl	(0x0401048C)
NLS_NOM_FLOW_AWAY_MAX_SPONT	(0x0002F87D)
NLS_NOM_EMFC_sPulsD	(0x04018204)
NLS_NOM_SETT_TIME_PD_EVOK	(0x0402F908)
NLS_NOM_EMFC_BPAPTH	(0x040180C8)
NLS_NOM_SETT_VENT_TIME_PD_BIPAP_HIGH	(0x0402F93D)
NLS_NOM_EMFC_iCa	(0x04010A2C)
NLS_NOM_CONC_CA_GEN	(0x00027118)
NLS_NOM_EMFC_tCO2_CALC	(0x04010A8C)
NLS_NOM_CONC_CO2_TOT_CALC	(0x0002F826)
NLS_NOM_EMFC_sHFVAm	(0x04018140)
NLS_NOM_SETT_HFV_AMPL	(0x0402F8F3)

Building a Computer Client

Interfacing the LAN interface with UDP/IP

When setting up a Computer Client, a network traffic analyzing tool can be useful to verify the success of each step. Widely used tools are:

- Microsoft® Network Monitor
- Tcpdump (available under the GNU Public License from ftp://ftp.ee.lbl.gov/)

Setting Up the BootP Server

Step 1: Connect the Computer Client to the IntelliVue monitor.

The Computer Client and the IntelliVue monitor should be connected with a crossover LAN cable. If you need a dedicated system to run the BootP server, use a hub/switch to connect the devices. It is strongly recommended that a dedicated network is used for the data export. Do not connect any additional devices.

Step 2: Start the BootP server.

Please refer to the documentation of your BootP/DHCP server for installation guidelines. If you use a DHCP server, make sure that the server supports BootP clients.

Step 3: Verify that the IntelliVue monitor receives a valid IP address.

Use a network monitor to verify that the IntelliVue monitor receives the correct IP address. If the IntelliVue monitor shows an **`Unsupported LAN`** INOP, it has not received a valid IP address.

If the IntelliVue monitor does not receive an IP address,

- make sure that there is no IP address conflict on the network
- try to reboot the IntelliVue monitor.

Parsing the Connect Indication Message

Step1: Verify that the Connect Indication message is sent.

Use a networked monitor and check that the IntelliVue monitor sends a subnet broadcast message to the Connect Indication port (24005). If the IntelliVue monitor does not send the message, verify that the IntelliVue monitor received a valid IP address from the BootP server (see “Setting Up the BootP Server” on page 279).

Step 2: Receive the Connect Indication message on the Computer Client.

Open a socket on the Computer Client that receives the subnet broadcast message. If the Computer Client does not receive the Connect Indication message, verify the correct network connection, use an ICMP echo (ping) to check connectivity of the IntelliVue monitor.

Step 3: Parse the Data Export Protocol Command.

The Computer Client must parse the Connect Indication message to determine the port for the Data Export Protocol. The message also contains the IP address of the IntelliVue monitor.

The Computer Client should check that all length and type fields in the message are set correctly, otherwise the message must be discarded.

Then the Computer Client should parse the appended *AttributeList* and extract the IP address and port information (refer to “Connect Indication Attributes” on page 107 for the specification of these attributes).

Interfacing the MIB/RS232 Interface with the Fixed Baudrate Protocol

Step 1: Connect the Computer Client to the IntelliVue monitor.

It may be useful to try out the Association Request/Response mechanism on the LAN interface before working with the MIB/RS232 interface. This might help to find out whether an error is related to a ill-formatted Data Export message or if it is related to a transport layer problem.

Step 2: Implement the framing algorithm.

The section “The Fixed Baudrate Protocol, RS232 Port Settings” on page 30 contains some examples which can be used to check if your framing algorithm works correctly. Remember to apply the framing algorithm to both the *Hdr* and *User Data* part of the message.

If you have tried out the Association Request message on the LAN interface, you can try to send the message within the Fixed Baudrate protocol. Just add the *Hdr* information and apply the framing algorithm.

You should keep the following points in mind when implementing the the Fixed Baudrate protocol:

- Verify that the checksum algorithm works correctly for received messages, i.e., make sure that received messages with a corrupt checksum are discarded.
- Make sure that you implement an exception handling in case the received message grows larger than your receive buffer (e.g., if an end of frame character is lost somewhere during communication).
- The Fixed Baudrate Protocol is not connection oriented. After starting your application, there may be an existing Data Export Association (either from running your own application previously or from another system which has been connected to the MIB/RS232 Interface before). This may have some unexpected consequences for your application.

Interfacing the MIB/RS232 Interface with the AutoSpeed Protocol

Step 1: Connect the Computer Client and the IntelliVue monitor.

If your operating system comes with an IrDA stack, please refer to the documentation of your operating system. The operating system will cover most of the steps below automatically.

It may be useful to try out the Association Request/Response mechanism on the LAN interface before working with the MIB/RS232 interface. This may help you to find out whether an error is related to a wrongly-formatted Data Export message or if it is related to a transport layer problem.

Step 2: Establish an IrDA connection

The IrDA protocol supports a device detection procedure. If the detection is successful, it will return information about the detected device. This information contains a device nickname and a service hints field which indicates that the device supports the IEEE 1073 standard.

After this the Computer Client can establish an IrLAP connection with the device. This involves the negotiation of the baudrate and packet size for the lower layers. Refer to the Serial Infrared Link Access Protocol (IrLAP) specification (see page 32) for more information on this topic.

Step 3: Query the IAS database

The IAS database contains the object “IEEE:1073:3:2:MDDL” with the attribute “IrDA:TinyTP:LsapSel”. This attribute contains the number of the TinyTP Service Access point for the Data Export protocol. The value type of the attribute is an *integer*. The value should be equal to 1 if the MIB/RS232 Interface is used for Data Export.

The database also contains an object named “IEEE:1073:3:2” with the attribute “NodeType”. This attribute is of type integer and specifies the type of driver which resides on the interface. A value of 1 indicates that it is a data source, i.e. it is used to export data from the monitor.

After finishing the IAS query, the Computer Client should close the IAS connection before connecting to the TinyTP Service Access Point.

Step 4: Connect to the IEEE:1073:3:2:MDDL TinyTP Service Access Point

After connecting to the TinyTP Service Access Point, the connection can be used to send Association Control and Data Export Protocol messages within TinyTP data packets.

You should check the following points for your IrDA protocol stack:

- The connection may be interrupted or reset due to communication problems (e.g., if the cable is disconnected, or the monitor is rebooted). The Computer Client should be able to recover from such problems and initiate a new connection. Note: when a disconnect occurs on the IrDA protocol layer, an Association on the Data Export protocol layer will be terminated automatically.
- The Data Export protocol is packet oriented, this means that data is exchanged as a sequence of packets. Your IrDA stack may or may not provide a packet oriented interface to the TinyTP layer. The Data Export software requires that a received IrDA packet contains only one Data Export Protocol message.

Establishing an Association

Step 1: Send an Association Request message to the IntelliVue monitor.

Format an Association Request message as described in the section “Association Request Message” on page 67. Make sure that all length fields are set correctly, the right byte order is used, and the compiler does not insert extra bytes for structure alignment.

Step 2: Parse the Association Response message sent by the IntelliVue monitor.

Verify that the IntelliVue monitor sends an Association Response message.

If the IntelliVue monitor does not send a Response message, this can have the following reasons:

- The Association Request message has been sent to the wrong port.
- The IntelliVue monitor is connected to a central station or has been connected to one (reboot the IntelliVue monitor).
- The Association Request message was not formatted correctly.

If the IntelliVue monitor sends a Refuse message, this can have the following reasons:

- The Association Request message was not formatted correctly or requested a protocol that is not supported by the IntelliVue monitor.
- The IntelliVue monitor already has an association with a different Computer Client on the same interface.
- The IntelliVue monitor already has an association with a different Computer Client on another interface and the active association uses a different source for the numeric data (only one source for numeric data may be active at a time). Please refer to “Association Request Message” on page 67 for more information on the different sources for numeric data.

If the Computer Client has an association with the IntelliVue monitor and sends a second Association Request from the same source port, the message is discarded.

Look for the byte sequence described in “Association Response Message” on page 73 to find the beginning of the User Data. Parse the User Data and make sure that the IntelliVue monitor sets the protocol versions and options as expected. Check that the requested optional packages are present.

Step 3: Parse the MDS Create Event message.

The IntelliVue monitor will send the MDS Create Event message shortly after the Association Response message. The Computer Client should parse the message and extract all necessary information. Refer to the section “Wave Objects” on page 82 for a description of the available attributes.

Step 4: Send an MDS Create Result message.

The Computer Client must send an MDS Create Result message to confirm the MDS Create Event message. Refer to “MDS CREATE EVENT RESULT” on page 55 to see how the message is formatted.

Make sure that the message uses the correct presentation context ID.

It is important that the result message has the same invoke ID as the MDS Create Event message.

If the IntelliVue monitor receives a correct MDS Create Result message, it stops re-sending MDS Create Event messages. Use a network monitor to verify this.

Step 5: Send a Release Request message.

Use the building blocks from the section “RELEASE REQUEST” on page 301 to build a Release Request message and send it to the IntelliVue monitor.

The IntelliVue monitor identifies a Computer Client based on its IP address and the source port of the messages. The Computer Client must use the same source port as in the Association Request for all communication during the association. If a message is sent from another source port, it will be treated as a message from a different Computer Client.

Step 6: Parse the Release Response message

The IntelliVue monitor sends the Release Response message to confirm that the association has been terminated. For the Computer Client it is sufficient to check the session header of the response and verify that it is indeed a Release Response message (see “Release Response” on page 73).

If the Computer Client does not receive the response message, it should try to resend the Release Request message.

To identify the IntelliVue Monitor software revision, poll the MDS objects system production attribute group and read the ProductSpecification attribute. (see “Attribute: Production Specification” on page 95)

Accessing Data

Step 1: Establish an association as described above.

Step 2: Send a Poll Data Request message to the IntelliVue monitor.

Message Frequencies

If the Computer Client sends Protocol Messages with a high frequency, the IntelliVue monitor is not able to process all the requests. Some of the messages will be discarded. The Computer Client can detect discarded Poll Data Request messages by checking the poll number in the response. The Computer Client must set the poll number so that it will be able to detect loss of messages.

Single and Extended Polling

If the Computer Client needs to access real-time numeric or wave data, it should use Poll Profile Extensions (see “EXTENDED POLL DATA REQUEST” on page 59). This avoids sending poll requests with a high frequency and reduces the communication overhead.

The Computer Client can use an Extended Poll Request only to access Numerics, Waves and Alarms. It must use Single Poll Data Requests to access data from Patient Demographics or from the Medical Device System object.

Receive the Poll Data Response message and parse it.

The IntelliVue monitor sends a Single or Extended Poll Data Result message if the Poll Request message was parsed correctly.

Availability of Data

Not all of the data is available right after a new association has been established. The time span until all data is collected depends on the internal update frequency of the data. Typical times are listed in the table below.

Object Type	Max. Time (Typical)
Numerics (real-time)	< 2 s
Numerics (12 second averaged)	< 18 s
Numerics (1 minute averaged)	< 70 s
Numerics (5 minute averaged)	< 310 s
Alarms	< 2 s
Patient Demographics	< 10 s
Medical Device System Object	< 1 s

During the startup phase, Poll Data Request messages on the object will result in Poll Data Response messages, which

- do not contain all the objects which are present in the IntelliVue monitor.
- do not contain all the available attributes of an object.

Numeric data is only available if a Measurement Server is connected to the IntelliVue monitor and if the system is not in stand-by mode. If a Measurement Server is connected to a running system, it may take several seconds until the data from the Measurement Server is available.

Parsing the Poll Result

The Poll Data Result message contains a checksum in the transport layer message. The Computer Client should verify that this checksum is correct. In the case of a corrupted checksum, the Computer Client must discard the message.

The Computer Client should check the poll number in the Poll Data Result message if it needs to detect lost messages. The Computer Client should check the *rel_time_stamp* which indicates the system time when the data was internally generated.

If the Computer Client needs to acquire a specific Numeric label (e.g., ABP), the preferred method is to use the *PhysioId* which is part of the Numeric Observed Value attribute (see “Numeric Objects” on page 75). The *physio_id* (physiological identifier) field contains a nomenclature code from the SCADA partition that identifies the represented value (typically a physiological measurement). It can be mapped to a label. However, for some numerics, the physio_id does not uniquely identify the measurement. E.g. all difference temperatures have the same physio_id, the numerics in the two channels of an EEG have the same physio_ids, the VueLink module may have numerics where the physio_id is not specified. However, if the label is derived by enumeration (e.g. the temperatures T1 and T2), the labels map to the same *PhysioId*. This ambiguity can be resolved if the user assigns other labels to the Numerics.

A Computer Client should not send Poll Requests for all attribute groups (polled_attr_grp = 0) when querying data with a high update frequency. Polling all attribute groups with a high frequency might lead to high system load and increased response latency. Future releases of the Data Export Protocol may support more attributes for each object.

If the IntelliVue monitor sends no response, check for the following causes:

- There is no association. Either the association was not established correctly or the IntelliVue monitor sent an Abort message (e.g., time-out) in the meantime.
- The Computer Client sent too many messages and messages were lost.
- The length of the transport layer message is corrupt.
- Length fields in the message are corrupt.

If the IntelliVue monitor sends a Remote Operation Error, this might have one of the following reasons:

- Wrong length field in the message.
- Wrong message type (*ro_type*, *command_type*, *action_type*).
- Wrong *managed_object* for the action (for Poll Requests, this must be the MDS object announced in the MDS Create Event).
- Wrong *polled_obj_type* (refer to “SINGLE POLL DATA REQUEST” on page 55 and “EXTENDED POLL DATA REQUEST” on page 59).
- Computer Client sent an Extended Poll Data Request, but the necessary optional package was not negotiated.
- Computer Client sent an Extended Poll Data Request with the wrong polled attribute group.
- Computer Client requested periodic Poll Data Result messages for too many objects. The Computer Client should at most send one request for Numerics (Metric Observed Value Attribute Group) and one for the AlertMonitor (Alert Monitor Attribute Group).

If the IntelliVue monitor sends a Poll Result message which does not contain all object/attributes check for the following problems:

- The Computer Client sent a Single Poll Data Request with the wrong polled attribute group. The Poll Result shows the objects with empty attribute lists (there are no attributes from the requested group).
- The association has been established and not all of the objects have been created. Wait until the objects are created.

Parsing AttributeLists

When parsing an *AttributeList*, the Computer Client should adhere to the following guidelines:

- Verify that the length fields in the *AttributeList* are consistent with other length fields in the message.
- Check both the count and length field of the *AttributeList* to detect the end of the list.
- Do not rely on the sequence of attributes in an *AttributeList*.
- Skip unknown attributes.
- Verify that the length field of each *AVAType* is consistent with its value.

If the Computer Client fails parsing the message, it is useful to compare the raw message (captured with a network monitor) with the Computer Client’s interpretation of the data. Common problems are:

- The Computer Client uses a different byte order. Wrong interpretation of length and count fields in particular can lead to problems.

- The Computer Client uses a different alignment for structures. The offset for members of a structure will be wrong, because the compiler for the Computer Client inserted bytes for alignment.
- Length fields denote the length of data appended, excluding the size of the length field.

Interpreting Data from Numerics

- Do not rely on the sequence of values within a Compound Numeric Observed Value attribute. The physiological identifiers must be interpreted.
- A triple valued pressure parameter can change to single valued (mean only), whenever the diastolic and systolic values are close together. This commonly happens when a pressure is being zeroed or when a transducer is left exposed to air. The parameter is still sent as a Compound Numeric Observed Value, even if only one value is available.
- The text in the label strings is localized. If you have a monitor with chinese localization, the strings will contain chinese UNICODE characters.

Interpreting Data from the Alert Monitor

- If the Computer Client wants to display Alarm messages, it should check the strings for UNICODE characters from the private use area (see “Definitions Shared by Protocols” on page 35).
- The text in the alarm strings is localized. If you have a monitor with chinese localization, the strings will contain chinese UNICODE characters.

Interpreting Wave Data

- The IntelliVue patient monitor supports the following wave types, which are defined by sample period, sample and array size (Sample Array Specification), and update period (Metric Specification) in the static context.

Wave Type	Sample Period	Sample Size	Array Size	Update Period	Bandwidth Requirement[1]
500 samples/s (ECG)	2 ms	16 bits	128 samples	256 ms	1064 bytes/s
250 samples/s (Compound ECG)	4 ms	16 bits	3*64 samples	256 ms	1640 bytes/s
125 samples/s	8 ms	16 bits	32 samples	256 ms	296 bytes/s
62.5 samples/s	16 ms	16 bits	16 samples	256 ms	168 bytes/s

[1] Observed values, not including context data.

- The Computer Client can poll the dynamic context to determine the available waves. Because of the high amount of data, the client should specify the required wave objects before requesting wave observed values in a periodic data poll.
- Up to three ECG waves (500 samples/s) can be polled simultaneously by selecting the appropriate lead labels in the Wave object priority list. The object handle is the same for all ECG waves. Waves can be identified by their physiological identifier.
- It is possible to select up to three individual ECG waves with 500 sps each or the single ECGcompound wave (containing three channels, 250 sps each). Additionally up to eight 125 sps or 62.5 sps waves may be chosen. Bandwidth restrictions need to be considered (see table above for bandwith usage of the individual wave types)

- In non-EASI mode, three ECG waves (250 samples/s, including the primary and secondary lead) can be polled by selecting the NLS_NOM_ECG_ELEC_POTL label in the Wave object priority list. The monitor sends poll results with a compound wave, containing three waves with common context. Waves can be identified by their physiological identifier.
- Up to eight non-ECG waves (125 or 62.5 samples/s) can be polled simultaneously by selecting the appropriate labels in the Wave object priority list.
- The Computer Client needs to keep track of the poll results time stamps to detect missing wave samples.
- Entries in the Wave object priority list are ignored if the label does not exist or the object is not available, or more than three ECG and/or more than eight non-ECG waves are specified.

The wave context can be polled separately or multiplexed with the wave observed values. If the *polled_attr_grp* is *0* in a periodic data poll request, the monitor reports one object's static and dynamic context per 1024 ms. Context attributes are included in the observation poll.

9

Troubleshooting

This chapter will help you identify and locate faults that may occur when using the Protocol. The procedure to locate faults uses a troubleshooting matrix.

When the fault has been identified, check the Possible Causes and corresponding Corrective Actions. Perform the corrective actions. Re-check the fault after each corrective action is performed until the fault has been cleared.It is assumed that you have a functioning Computer Client.

Fault	Possible Causes	Corrective Actions
Computer Client doesn't receive LAN messages	Cable connection is broken or wrong cable used.	• Verify that the IntelliVue monitor is correctly connected to the network. • Verify that the Computer Client is correctly connected to the network. Try to use an ICMP echo (ping) to check the monitor and Computer Client connections.
	IntelliVue monitor failure	Re-boot the IntelliVue monitor and try to make a new connection. Refer to the Troubleshooting section in the *Service Guide* of your device.
IntelliVue monitor shows an **Unsupported LAN** INOP	BootP server does not send a valid IP address.	Check the configuration of the BootP server. Check that the BootP server is correctly connected to the network.
	Cable connection is broken or wrong cable used.	Check the connection between the IntelliVue monitor and the BootP Server.
IntelliVue monitor shows a **No Central Monitoring** INOP	**Central Monitoring Mandatory** is configured to **On** in the monitor	Data Export must not be used with a central station. Configure **Central Monitoring** to **Optional**.
	Central Monitoring Mandatory is configured to **On** in the monitor and the connection to the central station is interrupted	Data Export must not be used with a central station. Reboot the IntelliVue monitor and make sure it is not connected to a central station.

Fault	Possible Causes	Corrective Actions
Computer Client doesn't receive messages with the AutoSpeed protocol	Cable connection is broken or wrong cable used.	Check the connection between the IntelliVue monitor and the Computer Client.
	Wrong configuration of MIB/ RS232 Interface	Check if the MIB/RS232 interface is configured for the desired protocol
	IntelliVue monitor failure	Re-boot the IntelliVue monitor and try to make a new connection. Disconnect the MIB/RS232 cable for more than 60s, this will most likely reset the IrDA stack of the client system too. Refer to the Troubleshooting section in the *Service Guide* of your device.
Computer Client does not establish an association.	Another Computer Client Application is already associated with the IntelliVue monitor.	Make sure no other Computer Client Application is trying to connect to the IntelliVue monitor. Reboot the IntelliVue monitor or wait until the association is timed out.
Computer Client does not report data.	Measurement Server is disconnected.	Connect the measurement server to the IntelliVue monitor,
	Parameter is switched off.	If the Computer Client requires a specific measurement, the parameter must be switched on in the IntelliVue monitor.
	Wave label is not included in the Wave object priority list.	Specify the wave objects to be polled in the Set Priority List Request
Wave Samples are missing in a perioidic data poll	Too many Wave objects polled.	Reduce the number of entries in the Wave object priority list.

NOTE The IntelliVue Data Export Interface cannot be accessed via the Local Area Network when the IntelliVue monitor is connected to the Philips LAN, e.g. to an Information Center (central station).

Communication via the MIB/RS232 Interface is always possible.

Further Troubleshooting

Further troubleshooting can be done using the Philips Data Export Test Tool (DETT).

DETT is used to test the communication interface protocol, which transfers data from the Philips IntelliVue Patient Monitor via the Local Area Network (LAN) Interface or Serial Interface (MIB/ RS232) to an external Computer.

Complete DETT functionality information is available in the DETT “Instructions for Use”.

Download DETT 453564212161_DETT.zip file from InCenter at: http://incenter.medical.philips.com

A personal InCenter login account is required to access DETT.

Customers or users without a personal InCenter login requiring the DETT, please contact your local Philips Response Center for further support.

The DETT “Instructions for Use” (453564254321.pdf) can be downloaded from InCenter. See the link listed above and its description.

DETT “Instruction for Use” Information is also available within the DETT program via “Help”.

10

Protocol Examples

Data Export Protocol Examples

CONNECT INDICATION EVENT

The Connect Indication message contains the *ConnectIndInfo* which is of variable length. The length fields in the message depend on the length of the *ConnectIndInfo*.This message is only available on the LAN interface.

```
Nomenclature        u_32                : 1.0
                        {0x00 0x00 0x01 0x00}
ROapdus             ro_type             : ROIV_APDU
                    length              : <xx>
                        {0x00 0x01 0xXX 0xXX}
ROIVapdu            invoke_id           : 0
                    command_type        : CMD_EVENT_REPORT
                    length              : <xx>
                        {0x00 0x00 0x00 0x00 0xXX 0xXX}
EventReportArg.
ManagedObjectId     m_obj_class       :NOM_MOC_VMS_MDS_COMPOS_SINGLE_BED
                    context_id          : 0
                    handle              : 0
RelativeTime        event_time          : 39424
OIDType             event_type          : NOM_NOTI_MDS_CONNECT_INDIC
u_16                length              : <xx>
                        {0x00 0x23 0x00 0x00 0x00 0x00 0x00 0x00
                         0x9A 0x00 0x0D 0x17 0xXX 0xXX}
ConnectIndInfo      [...]
```

MDS CREATE EVENT

The MDS Create Event message contains an *AttributeList* which is of variable length. The length fields in the message depend on the length of the *AttributeList*.

```
SPpdu              session_id          : 0xE100
                   p_context_id        : 2
                       {0xE1 0x00 0x00 0x02}
ROapdus            ro_type              : ROIV_APDU
                   length              : <xx>
                       {0x00 0x01 0xXX 0xXX}
ROIVapdu           invoke_id           : 1
                   command_type        : CMD_CONFIRMED_EVENT_REPORT
                   length              : <xx>
                       {0x00 0x01 0x00 0x01 0xXX 0xXX}
EventReportArg.
ManagedObjectId    m_obj_class         : NOM_MOC_VMS_MDS
                   context_id          : 0
                   handle              : 0
RelativeTime       event_time          : 126976
OIDType            event_type          : NOM_NOTI_MDS_CREAT
u_16               length              : <xx>
                       {0x00 0x21 0x00 0x00 0x00 0x00 0x00 0x01
                        0xf0 0x00 0x0d0x06 0xXX 0xXX}
MDSCreateInfo
ManagedObjectId    m_obj_class         : NOM_MOC_VMS_MDS
```

```
                  context_id          : 0
                  handle              : 0
                      {0x00 0x21 0x00 0x00 0x00 0x00}
AttributeList     [...]
```

MDS CREATE EVENT RESULT

```
SPpdu             session_id          : 0xE100
                  p_context_id        : 2
                      {0xE1 0x00 0x00 0x02}
ROapdus           ro_type             : RORS_APDU
                  length              : 20
                      {0x00 0x02 0x00 0x14}
RORSapdu          invoke_id           : 1
                  command_type        : CMD_CONFIRMED_EVENT_REPORT
                  length              : 14
                      {0x00 0x01 0x00 0x01 0x00 0x0e}
EventReportRes.
ManagedObjectId   m_obj_class         : NOM_MOC_VMS_MDS
                  context_id          : 0
                  handle              : 0
RelativeTime      event_time          : 4736768
OIDType           event_type          : NOM_NOTI_MDS_CREAT
u_16              length              : 0
                    {0x00 0x21 0x00 0x00 0x00 0x00 0x00 0x48
                     0x47 0x00 0x0d 0x06 0x00 0x00}
```

SINGLE POLL DATA REQUEST

```
SPpdu            session_id            : 0xE100
                 p_context_id          : 2
                    {0xE1 0x00 0x00 0x02}
ROapdus          ro_type               : ROIV_APDU
                 length                : 28
                    {0x00 0x01 0x00 0x1c}
ROIVapdu         invoke_id             : 0
                 command_type          : CMD_CONFIRMED_ACTION
                 length                : 22
                    {0x00 0x01 0x00 0x07 0x00 0x16}
ActionArgument
ManagedObjectId  m_obj_class           : NOM_MOC_VMS_MDS
                 context_id            : 0
                 handle                : 0
u_32             scope                 : 0
OIDType          action_type           : NOM_ACT_POLL_MDIB_DATA
u_16             length                : 8
                    {0x00 0x21 0x00 0x00 0x00 0x00 0x00 0x00
                     0x00 0x00 0x0c 0x16 0x00 0x08}
PollMdibDataReq
u_16             poll_number           : 1
TYPE             partition             : NOM_PART_OBJ
                 code                  : NOM_MOC_VMO_METRIC_NU
OIDType          polled_attr_grp       : all attribute groups
                    {0x00 0x01 0x00 0x01 0x00 0x06 0x00 0x00}
```

SINGLE POLL DATA RESULT

The Single Poll Data Result message contains a *PollInfoList* which is of variable length. The length fields in the message depend on the length of the *PollInfoList*.

```
SPpdu            session_id            : 0xE100
                 p_context_id          : 2
                    {0xE1 0x00 0x00 0x02}
ROapdus          ro_type               : RORS_APDU
                 length                : <xx>
                    {0x00 0x02 0xXX 0xXX}
RORSapdu         invoke_id             : 0
                 command_type          : CMD_CONFIRMED_ACTION
                 length                : <xx>
                    {0x00 0x00 0x00 0x07 0xXX 0xXX}
ActionResult
ManagedObjectId  m_obj_class           : NOM_MOC_VMS_MDS
                 context_id            : 0
                 handle                : 0
OIDType          action_type           : NOM_ACT_POLL_MDIB_DATA
u_16             length                : <xx>
```

```
                    {0x00 0x21 0x00 0x00 0x00 0x00 0x0c 0x16
                     0xXX 0xXX}
PollMdibDataReply
u_16             poll_number           : 1
RelativeTime     rel_time_stamp        : 4766464
AbsoluteTime     abs_time_stamp        : 0xffffffff 0xffffffff
TYPE             partition             : NOM_PART_OBJ
                 code                  : NOM_MOC_VMO_METRIC_NU
OIDType          polled_attr_grp       : all attribute groups
                    {0x00 0x01 0x00 0x48 0xbb 0x00 0xff 0xff
                     0xff 0xff 0xff 0xff 0xff 0xff 0x00 0x01
                     0x00 0x06 0x00 0x00}
PollInfoList     [...]
```

SINGLE POLL DATA RESULT (LINKED)

It is assumed that the IntelliVue monitor needs two messages to encode all the data from a Poll Request.

The first message would have a linked result header:

```
SPpdu            session_id            : 0xE100
                 p_context_id          : 2
                    {0xE1 0x00 0x00 0x02}
ROapdus          ro_type               : ROLRS_APDU
                 length                : <xx>
                    {0x00 0x05 0xXX 0xXX}
ROLRSapdu
RorlsId          state                 : RORLS_FIRST
                 count                 : 1
u_16             invoke_id             : 0
CMDType          command_type          : CMD_CONFIRMED_ACTION
u_16             length                : <xx>
                    {0x01 0x01 0x00 0x00 0x00 0x07 0xXX 0xXX}
ActionResult[...]
```

The second message would contain the rest of the data:

```
SPpdu            session_id            : 0xE100
                 p_context_id          : 2
                    {0xE1 0x00 0x00 0x02}
ROapdus          ro_type               : ROLRS_APDU
                 length                : <xx>
                    {0x00 0x05 0xXX 0xXX}
ROLRSapdu
RorlsId          state                 : RORLS_LAST
                 count                 : 2
u_16             invoke_id             : 0
CMDType          command_type          : CMD_CONFIRMED_ACTION
u_16             length                : <xx>
                    {0x03 0x02 0x00 0x00 0x00 0x07 0xXX 0xXX}
ActionResult[...]
```

Finally, the monitor sends a Remote Operation Result message:

```
SPpdu            session_id            : 0xE100
                 p_context_id          : 2
                    {0xE1 0x00 0x00 0x02}
ROapdus          ro_type               : RORS_APDU
                 length                :<xx>
                    {0x00 0x02 0xXX 0xXX}
RORSapdu         invoke_id             : 0
                 command_type          : CMD_CONFIRMED_ACTION
                 length                : <xx>
                    {0x00 0x00 0x00 0x07 0xXX 0xXX}
ActionResult     [...]
```

Note that all messages contain a fully encoded *ActionResult* data structure. The last Remote Operation Result message, however, would contain a *PollInfoList* structure with the *count* and *length* field set to 0. A client system should not depend on the terminating Remote Operation Result to have an empty *PollInfoList*. The message should be parsed as any other message.

EXTENDED POLL DATA REQUEST

The next example shows a message which could be used to access averaged data. The message will only be accepted if the optional package for Poll Profile Extensions has been negotiated during the association phase.

```
SPpdu            session_id           : 0xE100
                 p_context_id         : 2
                    {0xE1 0x00 0x00 0x02}
ROapdus          ro_type              : ROIV_APDU
                 length               : 32
                    {0x00 0x01 0x00 0x20}
ROIVapdu         invoke_id            : 0
                 command_type         : CMD_CONFIRMED_ACTION
                 length               : 26
                    {0x00 0x01 0x00 0x07 0x00 0x1a}
ActionArgument
ManagedObjectId  m_obj_class          : NOM_MOC_VMS_MDS
                 context_id           : 0
                 handle               : 0
u_32             scope                : 0
OIDType          action_type          : NOM_ACT_POLL_MDIB_DATA_EXT
u_16             length               : 12
                    {0x00 0x21 0x00 0x00 0x00 0x00 0x00 0x00
                     0x00 0x00 0xf1 0x3b 0x00 0x0c}
PollMdibDataReqExt
u_16             poll_number          : 1
TYPE             partition            : NOM_PART_OBJ
                 code                 : NOM_MOC_VMO_METRIC_NU
OIDType          polled_attr_grp      : all attribute groups
AttributeList
u_16             count                : 0
u_16             length               : 0
                    {0x00 0x01 0x00 0x01 0x00 0x06 0x00 0x00
                     0x00 0x00 0x00 0x00}
```

EXTENDED POLL DATA RESULT

The Extended Poll Data Result message contains an additional *sequence_no*, which is used if the client requests periodic replies.

```
SPpdu            session_id           : 0xE100
                 p_context_id         : 2
                    {0xE1 0x00 0x00 0x02}
ROapdus          ro_type              : RORS_APDU
                 length               : <xx>
                    {0x00 0x02 0xXX 0xXX}
RORSapdu         invoke_id            : 0
                 command_type         : CMD_CONFIRMED_ACTION
                 length               : <xx>
                    {0x00 0x00 0x00 0x07 0xXX 0xXX}
ActionResult
ManagedObjectId  m_obj_class          : NOM_MOC_VMS_MDS
                 context_id           : 0
                 handle               : 0
OIDType          action_type          : NOM_ACT_POLL_MDIB_DATA_EXT
u_16             length               : <xx>
                    {0x00 0x21 0x00 0x00 0x00 0x00 0xf1 0x3b
                     0xXX 0xXX}
PollMdibDataReplyExt
u_16             poll_number          : 1
u_16             sequence_no          : 0
RelativeTime     rel_time_stamp       : 4766464
AbsoluteTime     abs_time_stamp       : 0xffffffff 0xffffffff
TYPE             partition            : NOM_PART_OBJ
                 code                 : NOM_MOC_VMO_METRIC_NU
OIDType          polled_attr_grp      : all attribute groups
                    {0x00 0x01 0x00 0x00 0x00 0x48 0xbb 0x00
                     0xff 0xff 0xff 0xff 0xff 0xff 0xff 0xff
                     0x00 0x01 0x00 0x06 0x00 0x00}
PollInfoList     [...]
```

GET PRIORITY LIST REQUEST

```
SPpdu                    session_id              : 0xE100
```

```
                      p_context_id           : 2
               {0xE1 0x00 0x00 0x02}
ROapdus               ro_type                : ROIV_APDU
                      length                 : 22
               {0x00 0x01 0x00 0x16}
ROIVapdu              invoke_id              : 0
                      command_type           : CMD_GET
                      length                 : 16
               {0x00 0x00 0x00 0x03 0x00 0x10}
GetArgument
ManagedObjectId       m_obj_class            : NOM_MOC_VMS_MDS
                      context_id             : 0
                      handle                 : 0
u_32                  scope                  : 0
AttributeIdList       count                  : 1
                      length                 : 2
                      OIDType                : NOM_ATTR_POLL_RTSA_PRIO_LIST
               {0x00 0x21 0x00 0x00 0x00 0x00 0x00 0x00
                0x00 0x00 0x00 0x01 0x00 0x02 0xF2 0x3A}
```

GET PRIORITY LIST RESULT

```
SPpdu                 session_id             : 0xE100
                      p_context_id           : 2
               {0xE1 0x00 0x00 0x02}
ROapdus               ro_type                : RORS_APDU
                      length                 : <xx>
               {0x00 0x02 0xXX 0xXX}
RORSapdu              invoke_id              : 0
                      command_type           : CMD_GET
                      length                 : <xx>
               {0x00 0x00 0x00 0x03 0xXX 0xXX}
GetResult
ManagedObjectId       m_obj_class            : NOM_MOC_VMS_MDS
                      context_id             : 0
                      handle                 : 0
               {0x00 0x21 0x00 0x00 0x00 0x00}
AttributeList         count                  : 1
                      length                 : <xx>
AvaType               attribute_id           : NOM_ATTR_POLL_RTSA_PRIO_LIST
                      length                 : <xx>
               {0x00 0x01 0xXX 0xXX 0xF2 0x3A 0xXX 0xXX}
TextIdList            [...]
```

SET PRIORITY LIST REQUEST

```
SPpdu                 session_id             : 0xE100
                      p_context_id           : 2
               {0xE1 0x00 0x00 0x02}
ROapdus               ro_type                : ROIV_APDU
                      length                 : <xx>
               {0x00 0x01 0xXX 0xXX}
ROIVapdu              invoke_id              : 0
                      command_type           : CMD_CONFIRMED_SET
                      length                 : <xx>
               {0x00 0x00 0x00 0x05 0xXX 0xXX}
SetArgument
ManagedObjectId       m_obj_class            : NOM_MOC_VMS_MDS
                      context_id             : 0
                      handle                 : 0
u_32                  scope                  : 0
               {0x00 0x21 0x00 0x00 0x00 0x00 0x00 0x00
                0x00 0x00}
ModificationList      count                  : 1
```

```
                      length                : <xx>
AttributeModEntry     modifyOperator        : REPLACE
AvaType               attribute_id          : NOM_ATTR_POLL_RTSA_PRIO_LIST
                      length                : <xx>
               {0x00 0x01 0xXX 0xXX 0x00 0x00 0xF2 0x3A
                0xXX 0xXX}
TextIdList            [...]
```

SET PRIORITY LIST RESULT

```
SPpdu                 session_id            : 0xE100
                      p_context_id          : 2
               {0xE1 0x00 0x00 0x02}
ROapdus               ro_type               : RORS_APDU
                      length                : <xx>
               {0x00 0x02 0xXX 0xXX}
RORSapdu              invoke_id             : 0
                      command_type          : CMD_CONFIRMED_SET
                      length                : <xx>
               {0x00 0x00 0x00 0x05 0xXX 0xXX}
SetResult
ManagedObjectId       m_obj_class           : NOM_MOC_VMS_MDS
                      context_id            : 0
                      handle                : 0
               {0x00 0x21 0x00 0x00 0x00 0x00}
AttributeList         count                 : 1
                      length                : <xx>
AvaType               attribute_id          : NOM_ATTR_POLL_RTSA_PRIO_LIST
                      length                : <xx>
               {0x00 0x01 0xXX 0xXX 0xF2 0x3A 0xXX 0xXX}
TextIdList            [...]
```

AttributeList

This example shows an AttributeList which contains attributes from the Alert Monitor.

```
AttributeList     count                  : 5
                  length                 : 248
                     {0x00 0x05 0x00 0xf8}
AVAType           attribute_id           : NOM_ATTR_ID_HANDLE
                  length                 : 2
                  attribute_val          : 0x835d
                     {0x09 0x21 0x00 0x02 0x83 0x5d}
AVAType           attribute_id           : NOM_ATTR_ID_TYPE
                  length                 : 4
                  attribute_val          : 0x0001 0x0036
                     {0x09 0x2f 0x00 0x04 0x00 0x01 0x00 0x36}
AVAType           attribute_id           : NOM_ATTR_DEV_AL_COND
                  length                 : 10
                  attribute_val          : 0x1000 0x091a 0x0000 0x0002
                                           0x0000
                     {0x09 0x16 0x00 0x0a 0x10 0x00 0x09 0x1a
                      0x00 0x00 0x00 0x02 0x00 0x00}
AVAType           attribute_id           : NOM_ATTR_AL_MON_P_AL_LIST
                  length                 : 4
                  attribute_val          : 0x0000 0x0000
                     {0x09 0x02 0x00 0x04 0x00 0x00 0x00 0x00}
AVAType           attribute_id           : NOM_ATTR_AL_MON_T_AL_LIST
                  length                 : 208
attribute_val     : [...]
                     {0x09 0x04 0x00 0xd0 0x00 0x03 0x00 0xcc
                      0x4b 0xb8 0x01 0xba 0x00 0x02 0x10 0x00
                      0x00 0x02 0x00 0x00 0x83 0x3a 0x02 0x04
                      0x00 0x32 0x00 0x01 0x80 0x15 0x04 0x02
                      0x00 0x07 0x78 0x00 0x00 0x26 0x00 0x53
                      0x00 0x70 0x00 0x4f 0x20 0x82 0x00 0x20
                      0x00 0x4e 0x00 0x4f 0x00 0x4e 0x00 0x2d
                      0x00 0x50 0x00 0x55 0x00 0x4c 0x00 0x53
                      0x00 0x41 0x00 0x54 0x00 0x49 0x00 0x4c
                      0x00 0x45 0x00 0x00 0x50 0x00 0x01 0x12
                      0x00 0x02 0x10 0x00 0x00 0x09 0x00 0x00
                      0x02 0x91 0x02 0x04 0x00 0x32 0x00 0x01
                      0x00 0x03 0x01 0x0c 0x00 0x00 0x78 0x00
                      0x00 0x26 0x00 0x52 0x00 0x65 0x00 0x73
                      0x00 0x70 0x00 0x20 0x00 0x20 0x00 0x20
                      0x00 0x4c 0x00 0x45 0x00 0x41 0x00 0x44
                      0x00 0x53 0x00 0x20 0x00 0x4f 0x00 0x46
                      0x00 0x46 0x00 0x20 0x00 0x20 0x00 0x00
                      0x4a 0x04 0x00 0xf2 0x00 0x02 0x10 0x00
                      0x00 0x02 0x00 0x00 0x82 0x63 0x02 0x04
                      0x00 0x32 0x00 0x01 0x00 0x03 0x00 0x46
                      0x00 0x02 0x78 0x00 0x00 0x26 0x00 0x4e
                      0x00 0x42 0x00 0x50 0x00 0x20 0x00 0x20
                      0x00 0x20 0x00 0x20 0x00 0x45 0x00 0x51
                      0x00 0x55 0x00 0x49 0x00 0x50 0x00 0x20
                      0x00 0x4d 0x00 0x41 0x00 0x4c 0x00 0x46
                      0x00 0x20 0x00 0x00}
```

Association Control Protocol Examples

ASSOCIATION REQUEST

The following building blocks can be used to format an Association Request message:

AssocReqSessionHeader

```
0x0D <LI>
```

AssocReqSessionData

```
0x05 0x08 0x13 0x01 0x00 0x16 0x01 0x02
0x80 0x00 0x14 0x02 0x00 0x02
```

AssocReqPresentationHeader

```
0xC1 <LI> 0x31 0x80 0xA0 0x80 0x80 0x01
0x01 0x00 0x00 0xA2 0x80 0xA0 0x03 0x00
0x00 0x01 0xA4 0x80 0x30 0x80 0x02 0x01
0x01 0x06 0x04 0x52 0x01 0x00 0x01 0x30
0x80 0x06 0x02 0x51 0x01 0x00 0x00 0x00
0x00 0x30 0x80 0x02 0x01 0x02 0x06 0x0C
0x2A 0x86 0x48 0xCE 0x14 0x02 0x01 0x00
0x00 0x00 0x01 0x01 0x30 0x80 0x06 0x0C
0x2A 0x86 0x48 0xCE 0x14 0x02 0x01 0x00
0x00 0x00 0x02 0x01 0x00 0x00 0x00 0x00
0x00 0x00 0x61 0x80 0x30 0x80 0x02 0x01
0x01 0xA0 0x80 0x60 0x80 0xA1 0x80 0x06
0x0C 0x2A 0x86 0x48 0xCE 0x14 0x02 0x01
0x00 0x00 0x00 0x03 0x01 0x00 0x00 0xBE
0x80 0x28 0x80 0x06 0x0C 0x2A 0x86 0x48
0xCE 0x14 0x02 0x01 0x00 0x00 0x00 0x01
0x01 0x02 0x01 0x02 0x81
```

AssocReqUserData

The *AssocReqUserData* contains variable data, see see “Protocol Commands” on page 65.

AssocReqPresentationTrailer

```
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

ASSOCIATION RESPONSE

The following building blocks can be used to format an Association Response message:

AssocRespSessionHeader

```
0x0E <LI>
```

AssocRespSessionData

```
0x05 0x08 0x13 0x01 0x00 0x16 0x01 0x02
0x80 0x00 0x14 0x02 0x00 0x02
```

AssocRespPresentationHeader

```
0xC1 <LI> 0x31 0x80 0xA0 0x80 0x80 0x01
0x01 0x00 0x00 0xA2 0x80 0xA0 0x03 0x00
0x00 0x01 0xA5 0x80 0x30 0x80 0x80 0x01
0x00 0x81 0x02 0x51 0x01 0x00 0x00 0x30
0x80 0x80 0x01 0x00 0x81 0x0C 0x2A 0x86
0x48 0xCE 0x14 0x02 0x01 0x00 0x00 0x00
0x02 0x01 0x00 0x00 0x00 0x00 0x61 0x80
0x30 0x80 0x02 0x01 0x01 0xA0 0x80 0x61
0x80 0xA1 0x80 0x06 0x0C 0x2A 0x86 0x48
0xCE 0x14 0x02 0x01 0x00 0x00 0x00 0x03
0x01 0x00 0x00 0xA2 0x03 0x02 0x01 0x00
0xA3 0x05 0xA1 0x03 0x02 0x01 0x00 0xBE
0x80 0x28 0x80 0x02 0x01 0x02 0x81
```

AssocRespUserData

The *AssocRespUserData* contains variable data, see “Protocol Commands” on page 65.

AssocRespPresentationTrailer

```
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

REFUSE

The following building blocks can be used to format a Refuse message:

RefuseSessionHeader

```
0x0C 0x03
```

RefuseSessionData

```
0x32 0x01 0x00
```

RefusePresentationHeader

This block is empty in the Refuse message.

RefuseUserData

This block is empty in the Refuse message.

RefusePresentationTrailer

This block is empty in the Refuse message.

RELEASE REQUEST

The following building blocks can be used to format a Release Request message:

ReleaseReqSessionHeader

```
0x09 0x18
```

ReleaseReqSessionData

This block is empty in the Release Request message.

ReleaseReqPresentationHeader

```
0xC1 0x16 0x61 0x80 0x30 0x80 0x02 0x01
0x01 0xA0 0x80 0x62 0x80 0x80 0x01 0x00
0x00 0x00 0x00 0x00
```

ReleaseReqUserData

This block is empty in the Release Request message.

ReleaseReqPresentationTrailer

```
0x00 0x00 0x00 0x00
```

RELEASE RESPONSE

The following building blocks can be used to format a Release Response message:

ReleaseRespSessionHeader

```
0x0A 0x18
```

ReleaseRespSessionData

This block is empty in the Release Response message.

ReleaseRespPresentationHeader

```
0xC1 0x16 0x61 0x80 0x30 0x80 0x02 0x01
0x01 0xA0 0x80 0x63 0x80 0x80 0x01 0x00
0x00 0x00 0x00 0x00
```

ReleaseRespUserData

This block is empty in the Release Response message.

ReleaseRespPresentationTrailer

```
0x00 0x00 0x00 0x00
```

ASSOCIATION ABORT

The following building blocks can be used to format a Association Abort message:

AbortSessionHeader

```
0x19 0x2E
```

AbortSessionData

```
0x11 0x01 0x03
```

AbortPresentationHeader

```
0xC1 0x29 0xA0 0x80 0xA0 0x80 0x30 0x80
0x02 0x01 0x01 0x06 0x02 0x51 0x01 0x00
0x00 0x00 0x00 0x61 0x80 0x30 0x80 0x02
0x01 0x01 0xA0 0x80 0x64 0x80 0x80 0x01
0x01 0x00 0x00 0x00 0x00 0x00 0x00
```

AbortUserData

This block is empty in the Abort message.

AbortPresentationTrailer

```
0x00 0x00 0x00 0x00
```

User Data

The following section contains an example for the User Data which is contained in an Association Request message.

```
UserData
ASNLength        length                : 72
                    {0x48}
MDSEUserInfoStd
ProtocolVersion  protocol_version      : MDDL_VERSION1
NomenclatureVers.nomenclature_version  : NOMEN_VERSION
FunctionalUnits  functional_units      : 0
SystemType       system_type           : SYST_CLIENT
StartupMode       startup_mode         : COLD_START
                    {0x80 0x00 0x00 0x00 0x40 0x00 0x00 0x00
                     0x00 0x00 0x00 0x00 0x80 0x00 0x00 0x00
                     0x20 0x00 0x00 0x00}
Option List
AttributeList    count                 : 0
                 length                : 0
                    {0x00 0x00 0x00 0x00}
Supported Profiles
AttributeList    count                 : 1
                 length                : 44
                    {0x00 0x01 0x00 0x2c}
AVAType
OIDType          attribute_id          : NOM_POLL_PROFILE_SUPPORT
u_16             length                : 40
                    {0x00 0x01 0x00 0x28}
PollProfileSupport (attribute_val)
PollProfileRev.  poll_profile_revision : POLL_PROFILE_REV_0
RelativeTime     min_poll_period       : 800000
u_32              max_mtu_rx           : 1000
u_32              max_mtu_tx           : 1000
u_32             max_bw_tx             : 0xffff 0xffff
PollProfileOpt.  options               : 0x6000 0x0000
                    {0x80 0x00 0x00 0x00 0x00 0x00 0x09 0xc4
                     0x00 0x00 0x09 0xc4 0x00 0x00 0x03 0xe8
                     0xff 0xff 0xff 0xff 0x60 0x00 0x00 0x00}
Optional Packages
AttributeList    count                 : 1
                 length                : 12
                    {0x00 0x01 0x00 0x0c}
AVAType
OIDType          attribute_id          : NOM_ATTR_POLL_PROFILE_EXT
u_16             length                : 8
                     {0xf0 0x01 0x00 0x08}
PollProfileExt (attribute_val)
PollProfileExtOpt.options              : POLL_EXT_PERIOD_NU_AVG_60SEC
AttributeList    count                 : 0
                 length                : 0
                     {0x20 0x00 0x00 0x00 0x00 0x00 0x00 0x00}
```

With this User Data, the length field of the Presentation Header must be set to 220 (0xDC) and the length field of the Session Header must be set to 236 (0xEC).