MDS AERO SUPPORT CORPORATION

Proprietary Notice

This document is the property of MDS Aero Support Corporation, and is provided on condition that it be used exclusively for evaluation purposes. Any duplication or reproduction, in whole or in part, without prior written consent of an authorized MDS Aero Support Corporation representative is prohibited.

Revision History

Rev.	Date	Author	Sections Affected	Description
4	Feb 12, 2009	Kai Scheffler	2.32, 2.33	Added M1553B and THG subsystems.
4	Aug 21, 2009	Martin Hadaller	2.7.3.2 ff	Correct available chassis numbers to 1 - 3.
4	Aug 24, 2009	Martin Hadaller	1.5, 2.34	Added channel PLC Omron
5	Mar 01,2010	Martin Hadaller	2.31	Updated channel TSM
5	Mar 03,2010	Martin Hadaller	1.5 Figure 1 2.35	Added channel OPC
5	Apr 22, 2010	Peter Buchholz	1.5 Figure 1 2.36	Added channel VEXA
5	Jun 29, 2010	Martin Hadaller	2.36.3.5 2.36.3.7 2.36.3.11 2.36.3.12 2.36.3.13 2.36.3.14	Updated
5	Aug 23,2010	Martin Hadaller	1.5 Figure 1 2.37	Added channel DTS3250
5	Sep 08, 2010	Martin Hadaller	2.36.3.10	Added method GetVoltsSlopeAndOffset
5	Sep 23, 2010	Martin Hadaller	2.16.3.5	Added property IChannelPBS::InputType
5	Nov 02, 2010	Martin Hadaller	2.37.3.4 2.37.3.6 2.37.3.7	Updated DTS3250 Channel properties OutputEu, LowLimit and HighLimit
5	Nov 23,	Martin	2.36	Reviewed with MDS (FM, RS)

	2010	Hadaller	2.37	
5	Nov 24, 2010	Martin Hadaller	2.4.3.2	Added calculated channel persistency check

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

1.1.1 ProDAS (Professional Data Acquisition System) is a data acquisition system for gas turbine test cells. This specification defines the technical requirements for the interface offered by the Configuration Server-Channels.

1.1.2 The Configuration Server provides access to configuration data, which are stored in the configuration database as XML files.

1.1.3 This document defines the COM interface to be used by the clients.

1.1.4 The subsections *Events* Fired, *Usage Conditions and Restrictions*, *Persistent Data* and *Example* of the interface sections will be omitted if there is no relevant content.

1.1.5 For the sake of manageability, we have split the ICD for Configuration Server documentation in three parts. This document describes all sub-interfaces of the Configuration Server Channels interface.

1.2 Scope

1.2.1 This document is intended for programmers of the COM client components using the COM interface(s) specified herein as well as the programmers of the server program offering the COM interface(s).

1.3 Applicable Documents

Number	Title
ES78001.2620	Functional Requirements Document for ProDAS
ES78031.2660	Engineering Specification for Configuration Server
ICD78031.2661	Interface Control Document for the Configuration Server
ICD78031.2801	Interface Control Document for the Configuration Server - Subsystems

1.4 Codes and Standards

Number	Title
	ICD Template

1.5 Abbreviations and Definitions

Term	Definition
May	An option or permission
Shall	A mandatory requirement
Should	A recommendation
Will	A statement of intent
MDS	MDS Aero Support Corporation
MTU	MTU Aero Engines
ATH	Ambient Temperature / Humidity
ARINC	Aeronautical Radio Inc.
AVM	Aircraft Vibration Monitor
BCD	Binary Coded Decimal
BSTR	data type (32-bit character pointer)
C	C programming language
CEC	CEC C-CATS 8000 Vibration System
CG	Configuration GUI
CEMS	Continuous Emission Monitoring System
COM	Component Object Model
CONSORT	Pratt & Whitney interface computer to process ARINC data
CS	Configuration Server
DCOM	Distributed Component Object Model
DDS	Dynamic Data System
DDTC	Dynamic Data System Telemetry Control
DPS	DSA Pressure Scanner
DSA	Digital Sensor Array
DTS	Digital Thermocouple Sensor
EEC	Electronic Engine Control
EMS	Emission Monitoring System
ES	Engineering Specification
EU	Engineering Units
FCS	Facility Control System
FRD	Functional Requirements Document
GASS	General Analogue Sub-Systems from MDS
GASS AI	VXI GASS Analogue Input
GASS AO	VXI GASS Analogue Output
GASS DIO	VXI GASS Digital Input
GASS TC	VXI GASS Totaliser Counter
GASS VXI	VXI GASS Analogue Input, RTD Current Source and Frequency Measurement
GUI	Graphical User Interface
HIPPI	High Performance Parallel Interface
HP	Hewlett-Packard
HSS	High Speed Sentry
HSV	High Speed VXI
ICD	Interface Control Document
IDL	Interface Definition Language

LSB	Least Significant Bit
MG	Management GUI
MSB	Most Significant Bit
MSS	Mechanical Scanivalve
OLE	Object Linking and Embedding
OPC	OLE for Process Control
OTD	Open Transducer Detection
PBS	Pressure Brick Sub-System
PE	Piezo Electric
PLC	Programmable Logic Controller
PLCAB	Programmable Logic Controller of Allan Bradley
PLCGE	Programmable Logic Controller of GE Fanuc Facility Control
PLCOmron	Omron C200H Programmable Logic Controller
proDAS	Professional Data Acquisition System
PSI	Pressure Systems Incorporated
psi	Engineering Unit for pressure measurement
RMS/DC	Root Mean Square/ Direct Current
ROC	Rate of Change
RTE	Real Time Engine
RTD	Resistance-Temperature Device
RTP	Real Time Product
SCP	Signal Conditioning Plug-in
SCUTR	SCanning UTR
SDI	Source Destination Identifier
SS	Sub-System
SSM	Sign Status Matrix
TBDAU	Test Bed Data Acquisition Unit
TC	Thermo Couple
TDM	Time Division Multiplexing
TF	Tracking Filter
TFIA	Tracking Filter Input Assembly
ThrustES	Engine Specific Thrust Measurement System
ThrustNobel	Thrust Measurement System of Nobel
THG	Thrust Generic
TruTemp	Kaye TruTemp Temperature Measurement Subsystem
TSM	PSI Temperature Scanning Module
UTR	Uniform Temperature Reference
UTRH	UTR Half Box
Vib	Vibration Charge Amps
VME	Versa Module Europa
VBS	Visual Basic Script
VEXA	VTI EX10xxA
VTI	VTI Instruments Coporation
VXI	VME eXtension for Instrumentation
XML	Extensible Mark-up Language
XSL	Extensible Stylesheet Language

2. DESIGN

2.1 Introduction

2.1.1 Please refer to the Interface Control Document for the Configuration Server for a design overview. This section describes the design of the Channels interface and its specific sub-interfaces.

2.1.2 Figure 1 illustrates the object model, which the clients have to use. An arrow from A to B in the diagram means, that an object with interface B is created by an appropriate get property of interface A. The mechanism to access the Channels interface is described in the Interface Control Document for the Configuration Server.

2.1.3 The interface AlarmLimits may be obtained from each subsystem specific channel interface.

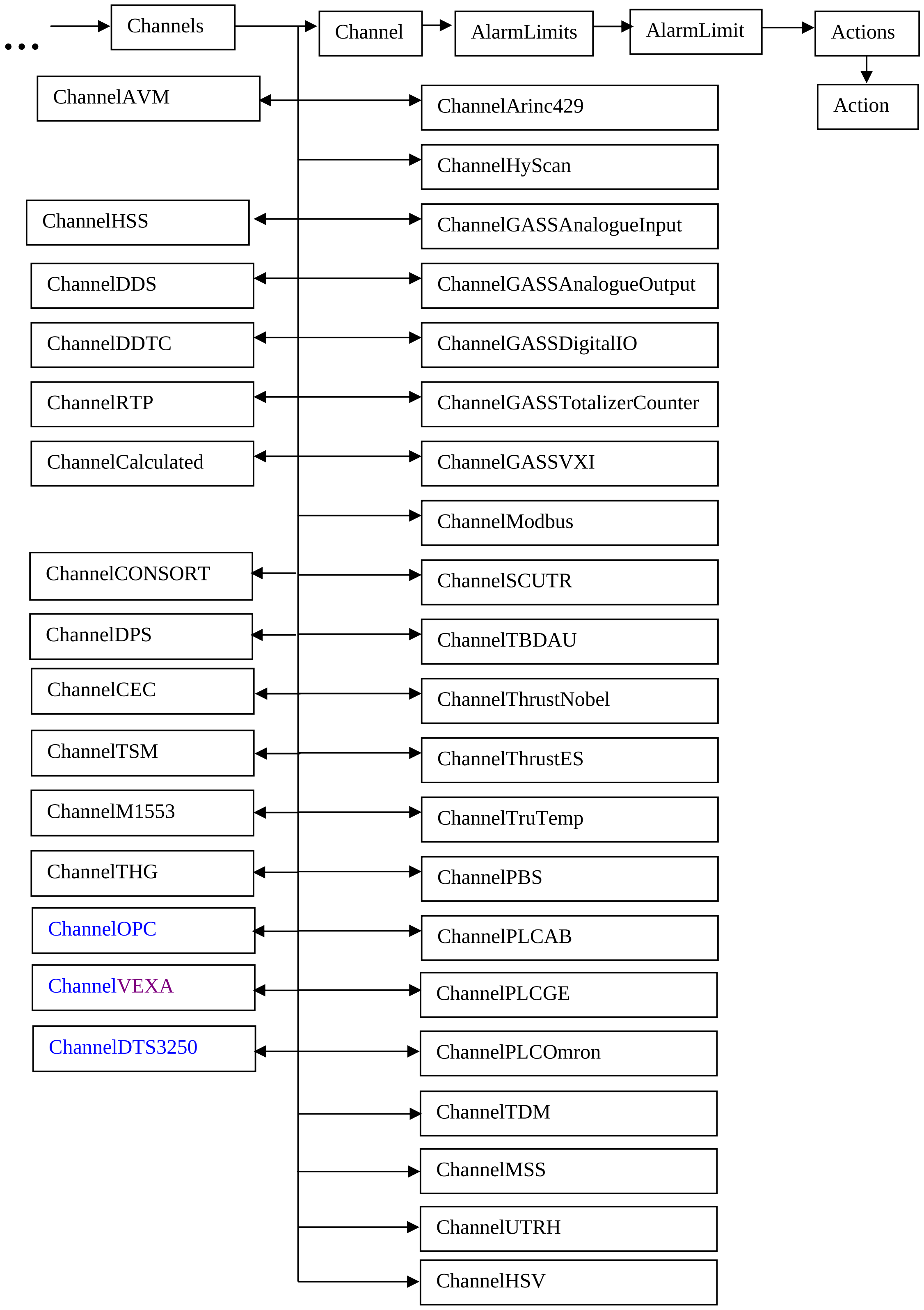

Figure 1: Object Model for Channels

2.2 Interface "ChannelArinc429"

2.2.1 Description

2.2.1.1 This interface represents a channel of an ARINC 429 subsystem.

2.2.2 Design

2.2.2.1 This interface shall be a dispatch interface.

2.2.2.2 This interface shall be an automation interface.

2.2.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.2.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.2.3 Methods and Properties

2.2.3.1 Property Pinout

```
// Pinout (Port)
[propget, id(1)]
HRESULT Pinout ([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT Pinout ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	defines a pinout in the list of pinouts described in the *ArincPinouts* interface. The default value shall be an empty string. If an empty string is set, an error shall be generated.

2.2.3.1.1 The Pinout must match a Pinout defined in the Subsystem *ArincPinouts* collection (cf. Interface Control Document for the Configuration Server - Subsystems, 2.3).

2.2.3.2 Property Label

```
[propget, id(2)]
HRESULT Label([out, retval] long *pVal);
[propput, id(2)]
HRESULT Label([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	a Label identifying an Arinc data word/channel in octal. Possible values are integers between 0 and 255. The default value shall be -1. If a value outside the allowed range is set or the *Label* is set to a *Label* of another channel with a matching SDI within the list of channels on the same port, an error shall be generated.

2.2.3.3 Property Bit

```
// Bit number
[propget, id(3)]
HRESULT Bit([out, retval] long *pVal);
[propput, id(3)]
HRESULT Bit([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the bit number for a Boolean channel. 11 represents the LSB and 29 the MSB. If the channel is a float, this parameter will be ignored i.e. the default shall be 0.

2.2.3.4 Property SDI

```
// SDI-Source Destination ID
[propget, id(4)]
HRESULT SDI([out, retval] long *pVal);
[propput, id(4)]
HRESULT SDI([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the Source/Destination Identifier. The default value shall be set to –1. The valid range is between 0 and 3. If any other value is set, an error shall be generated.

2.2.3.5 Property Coding

```
// Coding i.e. type of ARINC data
[propget, id(5)]
HRESULT Coding([out, retval] BSTR *pVal);
[propput, id(5)]
HRESULT Coding([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the encoding type of the Arinc data. Possible values are *BNR* and *BCD*. This property shall be ignored for Boolean channels. The default value shall be an empty string.

2.2.3.6 Property SSM

```
// SSM - Sign Status Matrix
[propget, id(6)]
HRESULT SSM([out, retval] long *pVal);
[propput, id(6)]
HRESULT SSM([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the SSM value corresponding to GOOD quality. This must be defined for all channels although it is ignored for BCD channels. The default value shall be -1. The valid range is between 0 and 3. If any other value is set, an error shall be generated.

2.2.3.7 Property SignificantBits

```
// Significant bits
[propget, id(7)]
HRESULT SignificantBits ([out, retval] long *pVal);
[propput, id(7)]
HRESULT SignificantBits ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the number of significant bits in the data word starting from the most significant bit. This is required for float channels only. The range is 1 to 19 for BNR encoded values, and can be 11, 15, 19, or 21 (for 3, 4, 5 and 6 digit BCD encoded values, respectively). The default value shall be 0.

2.2.3.8 Property Scale

```
// Scale i.e. the value of the most significant bit of the
// data word
[propget, id(8)]
HRESULT Scale([out, retval] long *pVal);
[propput, id(8)]
HRESULT Scale([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the value of the most significant bit of the data word. This must be a power of 2 from 2^0 to 2^18 and is required for float channels only. The default value shall be 0.

2.2.3.9 Property TimeOut

```
[propget, id(9)]
HRESULT TimeOut ([out, retval] long *pVal);
[propput, id(9)]
HRESULT TimeOut ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the interval, in milliseconds, allowed to receive the word twice i.e. this is only valid for input channels and the default is twice the channel *UpdateRate*.

2.2.3.10 Property OutputRate

```
// OutputRate (previously MSEC) Interval between transmissions
[propget, id(10)]
HRESULT OutputRate ([out, retval] long *pVal);
[propput, id(10)]
HRESULT OutputRate ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the interval between transmissions in milliseconds i.e. this is only valid for output channels. The default value shall be -1. . If a value less than 25 is specified, an error will be generated.

2.3 Interface "ChannelAVM"

2.3.1 Description

2.3.1.1 This interface represents a Tracking Filter, Tracking Filter Input Assembly, or Vibration Charge Amp channel of the Endevco AVM subsystem or a Vibration Charge Amp channel of the DDS_AVM subsystem.

2.3.2 Design

2.3.2.1 This interface shall be a dispatch interface.

2.3.2.2 This interface shall be an automation interface.

2.3.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.3.2.4 This interface shall include the properties: *SubsystemName, SubsystemType, ConfigLevel, IsChanged, LastModificationDate, Description, IsEnabled,* and *Name* of the **Channel** interface.

2.3.3 Methods and Properties

2.3.3.1 Property ChanNo

```
// Endevco Channel number
[propget, id(2)]
HRESULT ChanNo ([out, retval] long *pVal);
[propput, id(2)]
HRESULT ChanNo ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the AVM channel number. The default value shall be –1. The valid range is between 1 and 99. If any other value is set, an error shall be generated.

2.3.3.1.1 The Channel number shall be unique for the channels with the *InputType* set to *Tracking Filter* and shall be unique for all channels set to an *InputType* other than *Tracking Filter*.

2.3.3.2 Property InputType

```
// Channel Input type
[propget, id(3)]
HRESULT InputType([out, retval] BSTR *pVal);
[propput, id(3)]
HRESULT InputType([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the type of input transducer. The default value shall be an empty string. This property shall be either Single ended PE, Differential PE, Velocity, Isotron or Tracking Filter. If any other value is set, an error shall be generated.

2.3.3.3 Property Integration

```
// Integration
[propget, id(4)]
HRESULT Integration([out, retval] BSTR *pVal);
[propput, id(4)]
HRESULT Integration([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the type of output required. The default value shall be an empty string. The valid values shall be Acceleration, Velocity or Displacement output. The Acceleration output is not permitted for transducers with the *InputType* set to *Velocity* or *Tracking Filter*. If any other value is set, an error shall be generated.

2.3.3.4 Property AccelSens

```
// Acceleration Sensitivity
[propget, id(5)]
HRESULT AccelSens([out, retval] float *pVal);
[propput, id(5)]
HRESULT AccelSens([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	the acceleration sensitivity in pC/g. This is required if the *InputType* is an accelerometer (Single ended PE, Differential PE, or Isotron). The default value shall be 0.0. It must be 0.0 if *VeloSens* is set otherwise it will be a value between 1 and 110 with up to 3 decimal places. If any other value is set, an error shall be generated.

2.3.3.5 Property VeloSens

```
// Velocity Sensitivity
[propget, id(6)]
HRESULT VeloSens([out, retval] float *pVal);
[propput, id(6)]
HRESULT VeloSens([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	the transducer sensitivity in mV/ips. This is required if the *InputType* is set to *Velocity* transducer. The default value shall be 0.0. It must be 0.0 if *AccelSens* is set otherwise it will be a value between 40 and 500 with up to 2 decimal places. If any other value is set, an error shall be generated.

2.3.3.6 Property Range

```
[propget, id(7)]
HRESULT Range([out, retval] long *pVal);
[propput, id(7)]
HRESULT Range([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the EU full scale value for the output. This is required if the *InputType* is not set to *Tracking Filter*. This shall have the dimensions *g* (peak for acceleration), *inches/sec* (peak for velocity) and *milliseconds* (peak-to-peak for displacement). The default value shall be 0. The value shall be in the range 1 - 250. If any other value is set, an error shall be generated.

2.3.3.7 Property TimeConstant

```
[propget, id(8)]
HRESULT TimeConstant ([out, retval] long *pVal);
[propput, id(8)]
HRESULT TimeConstant ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the time constant for the RMS/DC converter in seconds for the DC output. This is required if the *InputType* is not set to *Tracking Filter*.The default value shall be 3. The value shall be in the range of 1 to 6 seconds. If any other value is set, an error shall be generated.

2.3.3.8 Property FilterType

```
[propget, id(9)]
HRESULT FilterType ([out, retval] BSTR *pVal);
[propput, id(9)]
HRESULT FilterType ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	defines which filtering shall be performed on the input. This could be either none (for acceleration and velocity), low, high or band. This is required if the *InputType* is not set to *Tracking Filter*. The default value shall be an empty string. The corresponding values are: Low, High and Both. If any other value is set, an error shall be generated.

2.3.3.9 Property FilterOrder

```
[propget, id(10)]
HRESULT FilterOrder ([out, retval] long *pVal);
[propput, id(10)]
HRESULT FilterOrder ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the order of the filter to be applied to the input. This is required if the *InputType* is not set to *Tracking Filter*. The default value shall be 0. Valid values are 4 for four-pole and 6 for six-pole. If any other value is set, an error shall be generated.

2.3.3.10 Property LowFreq

```
// Low frequency
[propget, id(11)]
HRESULT LowFreq ([out, retval] long *pVal);
[propput, id(11)]
HRESULT LowFreq ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the low cutoff frequency in Hz for high-pass and band-pass filters. This is required if the *InputType* is not set to *Tracking Filter*. The default value shall be 0. The valid range is between 15 and 579 Hz for a four-pole filter and between 15 and 528 Hz for a six-pole filter. If any other value is set, an error shall be generated.

2.3.3.11 Property HighFreq

```
// High frequency
[propget, id(12)]
HRESULT HighFreq ([out, retval] long *pVal);
[propput, id(12)]
HRESULT HighFreq ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the high cutoff frequency in Hz for low-pass and band-pass filters. This is required if the *InputType* is not set to *Tracking Filter*. The default value shall be 0. The valid range is between 28 and 6057 Hz for a four-pole filter and between 28 and 6645 Hz for a six-pole filter. If any other value is set, an error shall be generated.

2.3.3.12 Property VibChanNo

```
// Analogue Vibration Channel Number
[propget, id(14)]
HRESULT VibChanNo ([out, retval] long *pVal);
[propput, id(14)]
HRESULT VibChanNo ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the analogue vibration channel number for the tracking filter. This is required if the *InputType* is set to *Tracking Filter*. The default value shall be 0. It references the vibration channel relative to the existing chassis and can have a value of 1 to 8. If any other value is set, an error shall be generated.

2.3.3.13 Property TFBandWidth

```
// Tracking filter Bandwidth
[propget, id(15)]
HRESULT TFBandWidth ([out, retval] long *pVal);
[propput, id(15)]
HRESULT TFBandWidth ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the tracking filter bandwidth in Hz. This is required if the *InputType* is set to *Tracking Filter*. The default value shall be 0. The valid values are 1, 2, 5, 10 or 20 for Constant BW and 10 or 20 for Constant Q. If any other value is set, an error shall be generated.

2.3.3.14 Property TachChanNo

```
// Tachometer Channel Number
[propget, id(16)]
HRESULT TachChanNo ([out, retval] long *pVal);
[propput, id(16)]
HRESULT TachChanNo ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the tachometer input channel number. This is required if the *InputType* is set to *Tracking Filter*. The default value shall be 0. The valid range is between 1 and 4. If any other value is set, an error shall be generated.

2.3.3.15 Property TachRatio

```
// Tachometer ratios
[propget, id(17)]
HRESULT TachRatio ([out, retval] float *pVal);
[propput, id(7)]
HRESULT TachRatio ([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	the tachometer ratio. This is required if the *InputType* is set to *Tracking Filter*. The default value shall be 0.0. The valid range is between 0.0039 and 99.99. If any other value is set, an error shall be generated.

2.3.3.16 Property TachGain

```
// Tachometer gain
[propget, id(18)]
HRESULT TachGain ([out, retval] long *pVal);
[propput, id(18)]
HRESULT TachGain ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the tachometer gain. This is required if the *InputType* is set to *Tracking Filter*. The default value shall be -1. The valid values are 0, 1 or 2 where 0 represents a gain of 5, 1 represents a gain of 30 and 2 represents a gain of 0.5. If any other value is set, an error shall be generated.

2.3.3.17 Property ConstQ_BW

```
// Constant Q/Constant BW
[propget, id(19)]
HRESULT ConstQ_BW ([out, retval] BSTR *pVal);
[propput, id(19)]
HRESULT ConstQ_BW ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	a flag for either Constant Q or Constant Bandwidth. This is required if the *InputType* is set to *Tracking Filter*. The corresponding values shall be Q or BW. The default value shall be BW. If any other value is set, an error shall be generated.

2.4 Interface "ChannelCalculated"

2.4.1 Description

2.4.1.1 This interface represents a channel of a calculated subsystem.

2.4.2 Design

2.4.2.1 This interface shall be a dispatch interface.

2.4.2.2 This interface shall be an automation interface.

2.4.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.4.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.4.3 Methods and Properties

2.4.3.1 Property Equation

```
// Equation
[propget, id(2)]
HRESULT Equation ([out, retval] BSTR *pVal);
[propput, id(2)]
HRESULT Equation ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the expression used for calculating the value of the channel. The syntax of the expression is defined in the Functional requirements Document for proDAS, Annex A. The default value shall be “0.0”.

2.4.3.1.1 The syntax checking will be performed during the consistency check which occurs before each save operation or on request.

2.4.3.2 Property Persistency

```
// Persistency
[propget, id(2)]
HRESULT Persistency ([out, retval] long *pVal);
[propput, id(2)]
HRESULT Persistency ([in] long newVal);
```

Argument Name	**Description**

*pVal, newVal	Determines how the channel is saved when scanning is terminated: • 0 (default) = Volatile (not saved) • 1 = Test (saved on a per test basis) • 2 = Global (saved on a per test cell basis)

2.4.4 Usage Conditions and Restrictions

2.4.4.1 The maximum length of the equation is limited to 250 characters.

2.5 Interface "ChannelDDS"

2.5.1 Description

2.5.1.1 This interface will represent a channel for the DDS subsystem.

2.5.2 Design

2.5.2.1 This interface shall be a dispatch interface.

2.5.2.2 This interface shall be an automation interface.

2.5.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.5.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.5.3 Methods and Properties

2.5.3.1 Property Address

```
// Channel address
[propget, id(2)]
HRESULT Address ([out, retval] BSTR *pVal);
[propput, id(2)]
HRESULT Address ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	specifies what type of channel this is. The default value shall be an empty string. If an empty string is set an error shall be generated.

2.5.3.2 Property TrackedChannelName

```
// Tracked channel name
[propget, id(3)]
HRESULT TrackedChannelName([out, retval] BSTR *pVal);
[propput, id(3)]
HRESULT TrackedChannelName([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the name of the channel to be tracked. The default value shall be an empty string. If an empty string is set an error shall be generated.

2.5.3.3 Property TachoChannelName

```
// Tacho channel name
[propget, id(4)]
HRESULT TachoChannelName([out, retval] BSTR *pVal);
[propput, id(4)]
HRESULT TachoChannelName([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the name of the tacho channel to use to track the tracked channel. The default value shall be an empty string. If an empty string is set an error shall be generated.

2.5.3.4 Property EngineOrder

```
[propget, id(5)]
HRESULT EngineOrder([out, retval] long *pVal);
[propput, id(5)]
HRESULT EngineOrder([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the engine order to be tracked: 1 = first Engine Order, 2 = second Engine Order, etc. The default value shall be 0. If any other value is set an error shall be generated.

2.5.3.5 Property DeviceName

```
[propget, id(6)]
HRESULT DeviceName([out, retval] BSTR *pVal);
[propput, id(6)]
HRESULT DeviceName([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	The telemetry device name associated with the channel. The default value shall be an empty string. If an empty string is set or the name does not match a device name in the *DDSDevices* collection (cf. Interface Control Document for the Configuration Server - Subsystems, 2.9), an error shall be generated.

2.5.3.6 Property DeviceType

```
[propget, id(7)]
HRESULT DeviceType([out, retval] BSTR *pVal);
```

Argument Name	Description
*pVal	The device type associated to the telemetry device name defined in the DDS subsystem.

2.5.3.7 Property Port

```
[propget, id(8)]
HRESULT Port([out, retval] long *pVal);
```

Argument Name	Description
*pVal, newVal	The HIPPI Interface unit port number used to communicate with the telemetry device.

2.6 Interface "ChannelDDTC"

2.6.1 Description

2.6.1.1 This interface will represent a channel for the DDS DDTC subsystem.

2.6.2 Design

2.6.2.1 This interface shall be a dispatch interface.

2.6.2.2 This interface shall be an automation interface.

2.6.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.6.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.6.3 Methods and Properties

2.6.3.1 Property Address

```
// Channel address
[propget, id(2)]
HRESULT Address ([out, retval] BSTR *pVal);
[propput, id(2)]
HRESULT Address ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	specifies what type of channel this is. The default value shall be an empty string. If an empty string is set an error shall be generated.

2.6.3.2 Property StepValue

```
[propget, id(3)]
HRESULT StepValue ([out, retval] long *pVal);
[propput, id(3)]
HRESULT StepValue ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	a possible step value for the channel, which is used as an increment for the channel (raise/lower type of parameter). The default value shall be 0.

2.6.3.3 Property DeviceName

```
[propget, id(4)]
HRESULT DeviceName([out, retval] BSTR *pVal);
[propput, id(4)]
HRESULT DeviceName([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	The telemetry device name associated with the channel. The default value shall be an empty string. If an empty string is set or the name does not match a device name in the *DDSDevices* collection (cf. Interface Control Document for the Configuration Server - Subsystems, 2.9), an error shall be generated.

2.6.3.4 Property DeviceType

```
[propget, id(5)]
HRESULT DeviceType([out, retval] BSTR *pVal);
```

Argument Name	Description
*pVal	The device type associated to the telemetry device name defined in the DDS subsystem.

2.6.3.5 Property Port

```
[propget, id(6)]
HRESULT Port([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The HIPPI Interface unit port number used to communicate with the telemetry device.

2.7 Interface "ChannelGASSAnalogueInput"

2.7.1 Description

2.7.1.1 The GASS Analogue Input hardware comprises one or more E1413 VXI cards. Each GASSAI VXI card can have up to 8 Signal Conditioning Plug-ins attached. Each SCP contains 8 channels. The channels are numbered from 0 – 63 with channels 0 – 7 on the first SCP, 8 – 15 on the second etc.

2.7.1.2 There are currently two types of SCP in use referred to as 1502 and 1503. The 1503 has a superset of the 1502 features.

2.7.2 Design

2.7.2.1 This interface shall be a dispatch interface.

2.7.2.2 This interface shall be an automation interface.

2.7.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.7.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.7.3 Methods and Properties

2.7.3.1 Property Pinout

The calibration information is not retrieved by the Pinout name any more. Instead, the generic attribute CalSensor is used.

```
// GASSAI Pinout (Port)
[propget, id(5000)]
HRESULT Pinout ([out, retval] BSTR *pVal);
[propput, id(5000)]
HRESULT Pinout ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	a unique identifier to each connection point. Either this field or all of the following three properties (*Chassis*, *Slot* and *ChannelNumber*) must be defined. Either set can be used to uniquely identify the channel location. If all four are entered they must match. The default value shall be an empty string. If an empty string is set and *Chassis*, *Slot* and *ChannelNumber* are not set, an error shall be generated.

2.7.3.2 Property Chassis

```
[propget, id(5001)]
HRESULT Chassis ([out, retval] long *pVal);
[propput, id(5001)]
HRESULT Chassis ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the VXI chassis to which the channel belongs. The default value shall be 0. The valid range is from 1 to 3. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.7.3.3 Property Slot

```
[propget, id(5002)]
HRESULT Slot ([out, retval] long *pVal);
[propput, id(5002)]
HRESULT Slot ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the slot in the chassis to which the channel belongs. The default shall be 0. The valid range is from 1 to 12. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.7.3.4 Property ChannelNumber

```
// Card channel number
[propget, id(5003)]
HRESULT ChannelNumber ([out, retval] long *pVal);
[propput, id(5003)]
HRESULT ChannelNumber ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the channel number on the card to which the channel belongs. The default shall be –1. The valid range is from 0 to 63. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.7.3.5 Property Gain

```
[propget, id(5)]
HRESULT Gain ([out, retval] long *pVal);
[propput, id(5)]
HRESULT Gain ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the amplifier gain for the channel. A smaller input range requires a higher gain to optimise the resolution for the A/D converter. Valid gain settings for the 1503 SCP are 1, 8 and 64. The 1502 SCP only supports a gain setting of 1. The default shall be 1.

2.7.3.6 Property Range

```
[propget, id(6)]
HRESULT Range ([out, retval] float *pVal);
[propput, id(6)]
HRESULT Range ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	the voltage range of the A/D converter for the channel. The valid ranges in volts are 0.0625, 0.25, 1, 4 and 16. The default value shall be 16.

2.7.3.7 Property SensorName

This property is obsolete. The new properties are generic and named CalSensor, Cal Group and FixedSensor.

```
[propget, id(7)]
HRESULT SensorName ([out, retval] BSTR *pVal);
[propput, id(7)]
HRESULT SensorName ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the primary (sensor_name) and secondary (auxiliary_address) sensor identifier (dot separated). The default value shall be an empty string. An empty string is permitted. If a name is defined, it will be validated against the sensors defined in the database.

2.7.3.8 Property IsOTDOn

```
[propget, id(9)]
HRESULT IsOTDOn ([out, retval] BOOL *pVal);
[propput, id(9)]
HRESULT IsOTDOn ([in] BOOL newVal);
```

Argument Name	**Description**
*pVal, newVal	whether the ‘open transducer detection’ feature is turned on or off. This value must be the same for all channels on the same SCP. The default value shall be *false*.

2.7.3.9 Property InputFreq

```
// Input frequency
[propget, id(10)]
HRESULT InputFreq ([out, retval] long *pVal);
[propput, id(10)]
HRESULT InputFreq ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the input frequency in Hz. This is applicable to the 1503 SCP only. The value is ignored for the 1502 SCP. The default shall be -1. The valid values shall be 0, 2, 10 or 100. If any other value is set an error will be generated.

2.8 Interface "ChannelGASSAnalogueOutput"

2.8.1 Description

2.8.1.1 The GASS Analogue Output hardware comprises one or more E1418 VXI cards. This interface represents an analogue output channel of a VXI GASSAO subsystem.

2.8.2 Design

2.8.2.1 This interface shall be a dispatch interface.

2.8.2.2 This interface shall be an automation interface.

2.8.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.8.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.8.3 Methods and Properties

2.8.3.1 Property Pinout

```
// GASSAO Pinout (Port)
[propget, id(5000)]
HRESULT Pinout ([out, retval] BSTR *pVal);
[propput, id(5000)]
HRESULT Pinout ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	a unique identifier to each connection point. Either this field or all of the following three properties (*Chassis*, *Slot* and *ChannelNumber*) must be defined. Either set can be used to uniquely identify the channel location. If all four are entered they must match. The default value shall be an empty string. If an empty string is set and *Chassis*, *Slot* and *ChannelNumber* are not set, an error shall be generated.

2.8.3.2 Property Chassis

```
[propget, id(5001)]
HRESULT Chassis ([out, retval] long *pVal);
[propput, id(5001)]
HRESULT Chassis ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the VXI chassis to which the channel belongs. The default value shall be 0. The valid range is from 1 to 3. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.8.3.3 Property Slot

```
[propget, id(5002)]
HRESULT Slot ([out, retval] long *pVal);
[propput, id(5002)]
HRESULT Slot ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the slot in the chassis to which the channel belongs. The default shall be 0. The valid range is from 1 to 12. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.8.3.4 Property ChannelNumber

```
// Card channel number
[propget, id(5003)]
HRESULT ChannelNumber ([out, retval] long *pVal);
[propput, id(5003)]
HRESULT ChannelNumber ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the channel number on the card to which the channel belongs. The default shall be –1. The valid range is from 0 to 15. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.8.3.5 Property CurVolt

```
// Current / Voltage
[propget, id(5)]
HRESULT CurVolt ([out, retval] BSTR *pVal);
[propput, id(5)]
HRESULT CurVolt ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	whether the output is current or voltage. The default shall be an empty string. The corresponding values are: Current and Voltage. If any other value is set, an error shall be generated.

2.8.3.6 Property Slope

```
[propget, id(6)]
HRESULT Slope ([out, retval] float *pVal);
[propput, id(6)]
HRESULT Slope ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	the slope of the linear equation used to convert the engineering unit data into output voltage or current. The default value shall be 1.0.

2.8.3.7 Property Offset

```
[propget, id(7)]
HRESULT Offset ([out, retval] float *pVal);
[propput, id(7)]
HRESULT Offset ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	the offset of the linear equation used to convert the engineering unit data into output voltage or current. The default value shall be 0.0.

2.9 Interface "ChannelGASSDigitalIO"

2.9.1 Description

2.9.1.1 The GASS Digital I/O hardware comprises one E1458 VXI card. Each GASSDIO VXI card has 12 ports. Each port has 8 channels. The channels are numbered from 0 – 95 with channels 0 – 7 on one port, 8 – 15 on the next port etc.

2.9.2 Design

2.9.2.1 This interface shall be a dispatch interface.

2.9.2.2 This interface shall be an automation interface.

2.9.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.9.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.9.3 Methods and Properties

2.9.3.1 Property Pinout

```
// GASSDIO Pinout (Port)
[propget, id(5000)]
HRESULT Pinout ([out, retval] BSTR *pVal);
[propput, id(5000)]
HRESULT Pinout ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	a unique identifier to each connection point. Either this field or all of the following three properties (*Chassis*, *Slot* and *ChannelNumber*) must be defined. Either set can be used to uniquely identify the channel location. If all four are entered they must match. The default value shall be an empty string. If an empty string is set and *Chassis*, *Slot* and *ChannelNumber* are not set, an error shall be generated.

2.9.3.2 Property Chassis

```
[propget, id(5001)]
HRESULT Chassis ([out, retval] long *pVal);
[propput, id(5001)]
HRESULT Chassis ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the VXI chassis to which the channel belongs. The default value shall be 0. The valid range is from 1 to 3. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.9.3.3 Property Slot

```
[propget, id(5002)]
HRESULT Slot ([out, retval] long *pVal);
[propput, id(5002)]
HRESULT Slot ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the slot in the chassis to which the channel belongs. The default shall be 0. The valid range is from 1 to 12. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.9.3.4 Property ChannelNumber

```
// Card channel number
[propget, id(5003)]
HRESULT ChannelNumber ([out, retval] long *pVal);
[propput, id(5003)]
HRESULT ChannelNumber ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the channel number on the card to which the channel belongs. The default shall be –1. The valid range is from 0 to 95. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.9.3.4.1 Note groups of 8 consecutive channels constitute a port (0-7 = port 0, 8-15 = port 1 etc.).

2.9.3.5 Property Polarity

```
[propget, id(5)]
HRESULT Polarity ([out, retval] BSTR *pVal);
[propput, id(5)]
HRESULT Polarity ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the polarity of the port to which the channel belongs; Positive means that a channel value of *true* corresponds to a high voltage and *false* corresponds to a low voltage. Negative means vice versa. The default shall be Positive. The corresponding values are: Positive and Negative. If any other value is set, an error shall be generated.

2.9.3.5.1 Note: all channels on a port must have the same polarity and must all be input or output; if this is not the case the following definition is used: IN & OUT mixed on a port, port is set to IN. The Port’s first channel polarity drives the port polarity.

2.10 Interface "ChannelGASSTotalizerCounter"

2.10.1 Description

2.10.1.1 The GASS Totaliser Counter hardware comprises one or more E1415 VXI cards implementing the E1534 and E1538 SCP's. This interface represents an analogue input channel of a VXI GASSTC subsystem.

2.10.2 Design

2.10.2.1 This interface shall be a dispatch interface.

2.10.2.2 This interface shall be an automation interface.

2.10.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.10.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.10.3 Methods and Properties

2.10.3.1 Property Pinout

The calibration information is not retrieved by the Pinout name any more. Instead, the generic attribute CalSensor is used.

```
// Pinout (Port)
[propget, id(5000)]
HRESULT Pinout ([out, retval] BSTR *pVal);
[propput, id(5000)]
HRESULT Pinout ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	a unique identifier to each connection point. Either this field or all of the following three properties (*Chassis*, *Slot* and *ChannelNumber*) must be defined. Either set can be used to uniquely identify the channel location. If all four are entered they must match. The default value shall be an empty string. If an empty string is set and *Chassis*, *Slot* and *ChannelNumber* are not set, an error shall be generated.

2.10.3.2 Property Chassis

```
[propget, id(5001)]
HRESULT Chassis ([out, retval] long *pVal);
[propput, id(5001)]
HRESULT Chassis ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the VXI chassis to which the channel belongs. The default value shall be 0. The valid range is from 1 to 3. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.10.3.3 Property Slot

```
[propget, id(5002)]
HRESULT Slot ([out, retval] long *pVal);
[propput, id(5002)]
HRESULT Slot ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the slot in the chassis to which the channel belongs. The default shall be 0. The valid range is from 1 to 12. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.10.3.4 Property ChannelNumber

```
// Card channel number
[propget, id(5003)]
HRESULT ChannelNumber ([out, retval] long *pVal);
[propput, id(5003)]
HRESULT ChannelNumber ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the channel number on the card to which the channel belongs. The default shall be –1. The valid range is from 0 to 63. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.10.3.5 Property Type

```
[propget, id(5)]
HRESULT Type ([out, retval] BSTR *pVal);
[propput, id(5)]
HRESULT Type ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	defines what type of reading will be taken for the channel. Frequency or Period. Note the channel value is reported in Hz. The default shall be an empty string. The corresponding values are: Frequency and Period. If any other value is set, an error shall be generated.

2.10.3.6 Property Polarity

```
[propget, id(6)]
HRESULT Polarity ([out, retval] BSTR *pVal);
[propput, id(6)]
HRESULT Polarity ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the start edge polarity. Positive, rising edge trigger. Negative, falling edge trigger. The default shall be Positive. The corresponding values are: Positive and Negative. If any other value is set, an error shall be generated.

2.10.3.7 Property Threshold

```
[propget, id(7)]
HRESULT Threshold ([out, retval] float *pVal);
[propput, id(7)]
HRESULT Threshold ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	the threshold voltage at which an event is triggered. The default shall be –99V. Valid values for the trigger range are –46V to +46V in 375 mV increments. If any other value is set, an error shall be generated.

2.10.3.8 Property SensorName

This property is obsolete. The new properties are generic and named CalSensor, Cal Group and FixedSensor.

```
[propget, id(8)]
HRESULT SensorName ([out, retval] BSTR *pVal);
[propput, id(8)]
HRESULT SensorName ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the primary (sensor_name) and secondary (auxiliary_address) sensor identifier (dot separated). The default value shall be an empty string. An empty string is permitted. If a name is defined, it will be validated against the sensors defined in the database.

2.10.3.9 Property MinFrequency

```
[propget, id(10)]
HRESULT MinFrequency ([out, retval] long *pVal);
[propput, id(10)]
HRESULT MinFrequency ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the minimum expected frequency of the input in Hz. This is used to configure a timeout equal to 1/Frequency seconds. This is required in Frequency mode (*Type* = F) only. The default value shall be 1000. The valid range is 1 to 1000. If the value is set outside of this range, an error shall be generated.

2.10.3.10 Property Duration

```
[propget, id(11)]
HRESULT Duration ([out, retval] long *pVal);
[propput, id(11)]
HRESULT Duration ([in] long newVal);
```

Argument Name	**Description**

*pVal, newVal	the number of cycles to average. This parameter is required in the Period mode only (*Type* = P). The default value shall be 1. The valid range is 1 to 255. If the value is set outside of this range, an error shall be generated.

2.11 Interface "ChannelGASSVXI"

2.11.1 Description

2.11.1.1 The GASS VXI hardware comprises one or more E1413, E1415 and E1419 VXI cards. Each GASS VXI card can have up to 8 Signal Conditioning Plug-ins attached. Each SCP contains 8 channels. The channels are numbered from 0 to 63 with channels 0 to 7 on the first SCP, 8 to 15 on the second etc.

2.11.1.2 There are currently 10 types of SCP in use, referred to as 1501, 1502, 1503, 1505, 1508, 1509, 1512, 1534, 1536 and 1538. The 1501, 1502, 1503, 1505, 1508, 1509 and 1512 SCPs are all general analogue input SCPs. The 1503 is a fully programmable SCP, while the others are not programmable. The 1505 is an RTD current source SCP. The 1534 and 1538 SCPs are Frequency measurement programmable plug-ins used for Frequency, Period or Totaliser functions. The 1536 SCP is used for discrete input and output.

2.11.2 Design

2.11.2.1 This interface shall be a dispatch interface.

2.11.2.2 This interface shall be an automation interface.

2.11.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.11.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.11.3 Methods and Properties

2.11.3.1 Property Pinout

The calibration information is not retrieved by the Pinout name any more. Instead, the generic attribute CalSensor is used.

```
// Pinout (Port)
[propget, id(5000)]
HRESULT Pinout ([out, retval] BSTR *pVal);
[propput, id(5000)]
HRESULT Pinout ([in] BSTR newVal);
```

Argument Name	Description

*pVal, newVal	a unique identifier to each connection point. Either this field or all of the following three properties (*Chassis*, *Slot* and *ChannelNumber*) must be defined. Either set can be used to uniquely identify the channel location. If all four are entered they must match. The default value shall be an empty string. If an empty string is set and *Chassis*, *Slot* and *ChannelNumber* are not set, an error shall be generated.

2.11.3.2 Property Chassis

```
[propget, id(5001)]
HRESULT Chassis ([out, retval] long *pVal);
[propput, id(5001)]
HRESULT Chassis ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the VXI chassis to which the channel belongs. The default value shall be 0. The valid range is from 1 to 3. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.11.3.3 Property Slot

```
[propget, id(5002)]
HRESULT Slot ([out, retval] long *pVal);
[propput, id(5002)]
HRESULT Slot ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the slot in the chassis to which the channel belongs. The default shall be 0. The valid range is from 1 to 12. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.11.3.4 Property ChannelNumber

```
// Card channel number
[propget, id(5003)]
HRESULT ChannelNumber ([out, retval] long *pVal);
[propput, id(5003)]
HRESULT ChannelNumber ([in] long newVal);
```

Argument Name	**Description**

*pVal, newVal	the channel number on the card to which the channel belongs. The default shall be –1. The valid range is from 0 to 63. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.11.3.5 Property SensorName

This property is obsolete. The new properties are generic and named CalSensor, Cal Group and FixedSensor.

```
[propget, id(5)]
HRESULT SensorName ([out, retval] BSTR *pVal);
[propput, id(5)]
HRESULT SensorName ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the primary (sensor_name) and secondary (auxiliary_address) sensor identifier (dot separated). The default value shall be an empty string. An empty string is permitted. If a name is defined, it will be validated against the sensors defined in the database. For a thermocouple channel, this sensor will define the break point table used for the EU conversion of the thermocouple reading (millivolts to temperature) and the backward conversion of the RTD reference channel temperature reading (temperature to millivolts) which is used to compensate the thermocouple.

2.11.3.6 Property Type

```
[propget, id(7)]
HRESULT Type ([out, retval] BSTR *pVal);
[propput, id(7)]
HRESULT Type ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	specifies what type of reading shall be taken for the channel, Frequency, Period, Totalizer, Analogue or Discrete. The default value shall be an empty string. If any other value is set, an error shall be generated.

2.11.3.7 Property Gain

```
[propget, id(8)]
HRESULT Gain ([out, retval] long *pVal);
[propput, id(8)]
HRESULT Gain ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the amplifier gain for the channel. A smaller input range requires a higher gain to optimise the resolution for the A/D converter. Valid gain settings for the 1503 SCP are 1, 8 and 64. Any of the permissible values can be entered for other SCP types (1501, 1502, 1505, 1508, 1509, 1512) since these will be ignored. This is required in Analogue mode only (*Type* = Analogue). The default shall be 1.

2.11.3.8 Property Range

```
[propget, id(9)]
HRESULT Range ([out, retval] float *pVal);
[propput, id(9)]
HRESULT Range ([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	the voltage range of the A/D converter for the channel. The valid ranges in volts are 0.0625, 0.25, 1, 4 and 16. A combination of a Gain of 64 and a Range of 0.0625 is not allowed. This is required in Analogue mode only (*Type* = Analogue) but is ignored for the 1505 SCP type. The default value shall be 16.

2.11.3.9 Property IsOTDOn

```
[propget, id(10)]
HRESULT IsOTDOn ([out, retval] BOOL *pVal);
[propput, id(10)]
HRESULT IsOTDOn ([in] BOOL newVal);
```

Argument Name	**Description**
*pVal, newVal	whether the ‘open transducer detection’ feature is turned on or off. This value must be the same for all channels on the same SCP. This is required in Analogue mode only (*Type* = Analogue) but is ignored for the 1505 SCP. The default value shall be *false*.

2.11.3.10 Property InputFreq

```
// Input frequency
[propget, id(11)]
HRESULT InputFreq ([out, retval] long *pVal);
[propput, id(11)]
HRESULT InputFreq ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the input frequency in Hz. This is applicable to the 1503 SCP only and is ignored for all other types. The default shall be -1. The valid values shall be 0, 2, 10 or 100. This is required in Analogue mode only (*Type* = Analogue). If any other value is set, an error shall be generated.

2.11.3.11 Property Polarity

```
[propget, id(12)]
HRESULT Polarity ([out, retval] BSTR *pVal);
[propput, id(12)]
HRESULT Polarity ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the start edge polarity. This is required in the frequency (*Type* = Frequency), period (*Type* = Period), totaliser (*Type* = Totalizer) and discrete (*Type* = Discrete) modes. The default shall be Positive. The corresponding values are: Positive and Negative. If any other value is set, an error shall be generated. For Frequency, Period and Totalizer channels, Positive polarity indicates rising edge trigger while Negative polarity indicates falling edge trigger. For Discrete channels, Positive polarity indicates value 1 for high voltage reading while Negative polarity indicates value 0 for high voltage reading.

2.11.3.12 Property Threshold

```
[propget, id(13)]
HRESULT Threshold ([out, retval] float *pVal);
[propput, id(13)]
HRESULT Threshold ([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	the threshold voltage at which an event is triggered. The default shall be –99V. Valid values for the trigger range for totaliser, period and frequency channels are –46V to +46V in 375 mV increments and for discrete input channels 5, 12, 24 and 48 V. If any other value is set, an error shall be generated. This is required in the frequency (*Type* = Frequency), period (*Type* = Period), totaliser (*Type* = Totalizer) and discrete (*Type* = Discrete) modes.

2.11.3.13 Property MinFrequency

```
// Minimum frequency
[propget, id(14)]
HRESULT MinFrequency ([out, retval] long *pVal);
[propput, id(14)]
HRESULT MinFrequency ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the minimum expected frequency of the input in Hz. This is used to configure a timeout equal to 1/Frequency seconds. This is required in frequency (*Type* = Frequency) mode only. The default value shall be 1000. The valid range shall be 1 to 1000. If any other value is set, an error shall be generated.

2.11.3.14 Property Duration

```
[propget, id(15)]
HRESULT Duration ([out, retval] long *pVal);
[propput, id(15)]
HRESULT Duration ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the number of cycles to average. This parameter is required in the period mode only (*Type* = Period). The default value shall be 1. The valid range shall be 1 to 255. If any other value is set, an error shall be generated.

2.11.3.15 Property RTDReference

```
[propget, id(16)]
HRESULT RTDReference([out, retval] BSTR *pVal);
[propput, id(16)]
HRESULT RTDReference([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	Identifier for the RTD reference channel. If defined, this channel is processed as a thermocouple channel, otherwise normal analogue input processing is performed. This is used for analogue (*Type* = Analogue) mode only. If defined, the reference channel must be a valid channel.

2.11.3.16 Property DebounceTime

```
[propget, id(17)]
HRESULT DebounceTime([out, retval] float *pVal);
[propput, id(17)]
HRESULT DebounceTime([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	Debounce time of the switch for discrete input channels in milliseconds. Valid values are 0, 0.15, 0.3, 0.6, 1.2, 2.4, 4.8, 9.6, 19.2, 38.4, 76.6, 153.6, 307.2, 614.4, 1228.8 and 2457.6 ms. Channels 0-3 and 4-7 on a 1536 SCP will have the same debounce times i.e. on one SCP two times must be identified. This is required for discrete (*Type* = Discrete) mode only. The default value is 0.

2.12 Interface “ChannelHSV”

2.12.1 Description

2.12.1.1 The High Speed VXI hardware comprises one or more E1413 cards. Each card can have up to 8 Signal Conditioning Plug-ins attached. Each SCP contains 8 channels. The channels are numbered from 0 to 63 with channels 0 to 7 on the first SCP, 8 to 15 on the second etc.

2.12.1.2 There are currently 7 types of SCP in use, referred to as 1501, 1502, 1503, 1508, 1509, 1510 and 1512.

2.12.2 Design

2.12.2.1 This interface shall be a dispatch interface.

2.12.2.2 This interface shall be an automation interface.

2.12.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.12.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.12.3 Methods and Properties

2.12.3.1 Property Pinout

The calibration information is not retrieved by the Pinout name any more. Instead, the generic attribute CalSensor is used.

```
[propget, id(5000)]
HRESULT Pinout ([out, retval] BSTR *pVal);
[propput, id(5000)]
HRESULT Pinout ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	A unique identifier to each connection point. Either this field or all of the following three properties (*Chassis*, *Slot* and *ChannelNumber*) must be defined. Either set can be used to uniquely identify the channel location. If all four are entered they must match. The default value shall be an empty string. If an empty string is set and *Chassis*, *Slot* and *ChannelNumber* are not set, an error shall be generated.

2.12.3.2 Property Chassis

```
[propget, id(5001)]
HRESULT Chassis ([out, retval] long *pVal);
[propput, id(5001)]
HRESULT Chassis ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	The VXI chassis to which the channel belongs. The default value shall be 0. The valid range is from 1 to 3. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.12.3.3 Property Slot

```
[propget, id(5002)]
HRESULT Slot ([out, retval] long *pVal);
[propput, id(5002)]
HRESULT Slot ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	The slot in the chassis to which the channel belongs. The default shall be 0. The valid range is from 1 to 12. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.12.3.4 Property ChannelNumber

```
[propget, id(5003)]
HRESULT ChannelNumber ([out, retval] long *pVal);
[propput, id(5003)]
HRESULT ChannelNumber ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	The channel number on the card to which the channel belongs. The default shall be –1. The valid range is from 0 to 63. If a value outside this range is set and *Pinout* is an empty string, an error shall be generated.

2.12.3.5 Property SensorName

This property is obsolete. The new properties are generic and named CalSensor, Cal Group and FixedSensor.

```
[propget, id(5)]
HRESULT SensorName ([out, retval] BSTR *pVal);
[propput, id(5)]
HRESULT SensorName ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the primary (sensor_name) and secondary (auxiliary_address) sensor identifier (dot separated). The default value shall be an empty string. An empty string is permitted. If a name is defined, it will be validated against the sensors defined in the database. For a thermocouple channel, this sensor will define the break point table used for the EU conversion of the thermocouple reading (millivolts to temperature) and the backward conversion of the RTD reference channel temperature reading (temperature to millivolts) which is used to compensate the thermocouple.

2.12.3.6 Property Gain

```
[propget, id(6)]
HRESULT Gain ([out, retval] float *pVal);
[propput, id(6)]
HRESULT Gain ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	The amplifier gain for the channel. A smaller input range requires a higher gain to optimise the resolution for the A/D converter. Valid gain settings for the 1503 SCP are 1, 8 and 64. For the 1510 SCP, valid gain settings for the first four channels are 0.5, 8, 64 and 512 and for the last four channels 1. The default shall be 1.

2.12.3.7 Property Range

```
[propget, id(7)]
HRESULT Range ([out, retval] float *pVal);
[propput, id(7)]
HRESULT Range ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	The voltage range of the A/D converter for the channel. The valid ranges in volts are 0.0625, 0.25, 1, 4 and 16. A combination of a Gain of 64 and a Range of 0.0625 is not allowed for the 1503 SCP. The default value shall be 16.

2.12.3.7.1 The gain and voltage range combinations available to the E1510 SCP will be limited to the combinations listed in the table below, in order to comply with the hardware limitations. This will allow a maximum input voltage of between 0.00390625 V and 8 V to be processed depending on the settings chosen below.

E1510 Gain settings	Allowable E1413 Voltage Range Settings	Overall Input Channel Range (+/- Volts)
0.5	0.0625	0.125
0.5	0.25	0.5
0.5	1	2
0.5	4	8
8	0.0625	0.0078125
8	0.25	0.03125
8	1	0.125
8	4	0.5
64	0.25	0.00390625
64	1	0.015625
64	4	0.0625
512	4	0.0078125

2.12.3.8 Property IsOTDOn

```
[propget, id(8)]
HRESULT IsOTDOn ([out, retval] BOOL *pVal);
[propput, id(8)]
HRESULT IsOTDOn ([in] BOOL newVal);
```

Argument Name	Description
*pVal, newVal	Whether the ‘open transducer detection’ feature is turned on or off. This value must be the same for all channels on the same SCP. The default value shall be *false*.

2.12.3.9 Property InputFreq

```
[propget, id(9)]
HRESULT InputFreq ([out, retval] long *pVal);
[propput, id(9)]
HRESULT InputFreq ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	The input frequency in Hz. This is applicable to the 1503 SCP and 1510 SCP only and is ignored for all other SCP types. The default shall be -1. The valid values for the 1503 SCP will be 0, 2, 10 or 100. For the 1510 SCP, the valid values for the first four channels will be 15, 100, 250, 500, 1000 and for the last four channels 0. If any other value is set, an error shall be generated.

2.12.3.10 Property RTDReference

```
[propget, id(10)]
HRESULT RTDReference([out, retval] BSTR *pVal);
[propput, id(10)]
HRESULT RTDReference([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Identifier for the RTD reference channel. If defined, this channel is processed as a thermocouple channel, otherwise normal analogue input processing is performed. If defined, the reference channel must be a valid channel.

2.13 Interface "ChannelHSS"

2.13.1 Description

2.13.1.1 This interface represents a channel of a High Speed Sentry subsystem.

2.13.2 Design

2.13.2.1 This interface shall be a dispatch interface.

2.13.2.2 This interface shall be an automation interface.

2.13.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.13.2.4 This interface shall include the properties: *SubsystemName, IsEnabled, ConfigLevel, IsChanged, LastModificationDate, Description, Name* and *Engineering Unit* of the **Channel** interface.

2.13.3 Methods and Properties

2.13.3.1 Property Number

```
// Channel number
[propget, id(2)]
HRESULT Number([out, retval] long *pVal);
[propput, id(2)]
HRESULT Number([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the HSS channel number which shall be between 1 and 16. If any other value is set, an error shall be generated. The default value shall be –1.

2.13.3.2 Property SampleRate

```
[propget, id(4)]
HRESULT SampleRate([out, retval] long *pVal);
[propput, id(4)]
HRESULT SampleRate([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the sample rate of the channel. The default value shall be -1. Valid values are 1000, 2000, 3000, 4000 or 5000 Hz. If any other value is set, an error shall be generated.

2.13.3.3 Property Slope

```
[propget, id(5)]
HRESULT Slope([out, retval] float *pVal);
[propput, id(5)]
HRESULT Slope([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	the slope in the time domain used to convert the binary signal into engineering units. The default value shall be 1.0.

2.13.3.4 Property Offset

```
[propget, id(6)]
HRESULT Offset([out, retval] float *pVal);
[propput, id(6)]
HRESULT Offset([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	the offset in the time domain used to convert the binary signal into engineering units. The default value shall be 0.0.

2.14 Interface "ChannelHyScan"

2.14.1 Description

2.14.1.1 This interface represents a channel of a HyScan subsystem.

2.14.2 Design

2.14.2.1 This interface shall be a dispatch interface.

2.14.2.2 This interface shall be an automation interface.

2.14.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.14.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.14.3 Methods and Properties

2.14.3.1 Property ChannelType

```
[propget, id(1)]
HRESULT ChannelType ([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT ChannelType ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the type of channel being used, where DIFF represents a differential pressure reading and ABS represents an absolute pressure reading. The HyScan returns differential pressure readings from the ZOC modules. The subsystem calculates the absolute reading by adding the associated reference reading to the differential pressure reading. The default value shall be an empty string. If an invalid value is set, an error shall be generated.

2.14.3.2 Property HyScan

```
[propget, id(2)]
HRESULT HyScan ([out, retval] BSTR *pVal);
[propput, id(2)]
HRESULT HyScan ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the HyScan unit for the channel. Valid values shall be A, B or C depending on the number of PC's defined in the *SubsystemHyScan* hardware definition i.e. if only 2 PC's were defined, only A and B would be valid. The default value shall be an empty string. If an invalid value is set, an error shall be generated.

2.14.3.3 Property Module

```
[propget, id(3)]
HRESULT Module ([out, retval] long *pVal);
[propput, id(3)]
HRESULT Module ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the HyScan module number of the channel. Valid values shall be 1 to 64. The default value shall be 0. If an invalid value is set, an error shall be generated.

2.14.3.4 Property Port

```
[propget, id(4)]
HRESULT Port ([out, retval] long *pVal);
[propput, id(4)]
HRESULT Port ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the port number of the channel on the HyScan module. Valid values shall be 1 to 16. The default value shall be 0. If an invalid value is set, an error shall be generated.

2.14.3.5 Property HighLineRef

```
// Highline reference
[propget, id(5)]
HRESULT HighLineRef ([out, retval] BSTR *pVal);
[propput, id(5)]
HRESULT HighLineRef ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the High-line reference channel name used to convert a differential channel reading to an absolute channel reading. This is mandatory for ABS channel types and optional for DIFF channel types. The default value shall be an empty string. If an empty string is set for an ABS channel type, an error shall be generated.

2.14.3.6 Property UnitsFactor

```
[propget, id(6)]
HRESULT UnitsFactor ([out, retval] float *pVal);
[propput, id(6)]
HRESULT UnitsFactor ([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	an optional conversion factor to apply to the pressure reading. Pressures are nominally returned in *psi* by the HyScan system. The default value shall be 1.0.

2.15 Interface "ChannelModBus"

2.15.1 Description

2.15.1.1 This subsystem is used to communicate with the *Modbus* protocol via a *Serial* or an *Ethernet* interface. (Examples of such subsystems are Woodward EMS, the Richard-Oliver Continuous Emission Monitoring System (CEMS) and the Torquetronics Torquemeter system.)

2.15.2 Design

2.15.2.1 This interface shall be a dispatch interface.

2.15.2.2 This interface shall be an automation interface.

2.15.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.15.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.15.3 Methods and Properties

2.15.3.1 Property Slope

```
[propget, id(1)]
HRESULT Slope ([out, retval] float *pVal);
[propput, id(1)]
HRESULT Slope ([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	the slope of the linear equation used to convert the input data received from the slave device. This is ignored for Boolean type channels. The default value shall be 1.0.

2.15.3.2 Property Offset

```
[propget, id(2)]
HRESULT Offset ([out, retval] float *pVal);
[propput, id(2)]
HRESULT Offset ([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	the offset of the linear equation used to convert the input data received from the slave device. This is ignored for Boolean type channels. The default value shall be 0.0.

2.15.3.3 Property Location

```
[propget, id(3)]
HRESULT Location ([out, retval] long *pVal);
[propput, id(3)]
HRESULT Location ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the physical memory address of the data item in the slave device. This value starts at 1 and is limited to the maximum address defined in the *SubsystemModbus* hardware definition.

2.16 Interface "ChannelPBS"

2.16.1 Description

2.16.1.1 This interface represents a channel for a PSI pressure brick.

2.16.2 Design

2.16.2.1 This interface shall be a dispatch interface.

2.16.2.2 This interface shall be an automation interface.

2.16.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.16.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.16.3 Methods and Properties

2.16.3.1 Property SerialNumber

```
[propget, id(1)]
HRESULT SerialNumber([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT SerialNumber([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	a unique identifier for each pressure brick. This value will be validated against the list of serial numbers defined for the pressure bricks in the *SubsystemPBS* hardware definition.

2.16.3.2 Property Port

```
// Port number
[propget, id(2)]
HRESULT Port([out, retval] long *pVal);
[propput, id(2)]
HRESULT Port([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the port on the brick to which the channel belongs. The default shall be -1. The valid range is from 1 to 16. If a value outside this range is set an error shall be generated.

2.16.3.3 Property AvgSamples

```
// Number of samples to average together
[propget, id(3)]
HRESULT AvgSamples([out, retval] long *pVal);
[propput, id(3)]
HRESULT AvgSamples([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the number of samples to average together to attain one scan value. This value must be the same for all channels on the same brick. The default shall be 32.

2.16.3.4 Property AmbPressChannel

```
// Name of the channel which provides the ambient pressure
[propget, id(4)]
HRESULT AmbPressChannel([out, retval] BSTR *pVal);
[propput, id(4)]
HRESULT AmbPressChannel([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the name of the channel which provides the ambient pressure. The default value shall be an empty string.

2.16.3.5 Property InputType

```
// The type of the signal being measured
[propget, id(5)]
HRESULT InputType ([out, retval] BSTR *pVal);
[propput, id(5)]
HRESULT InputType ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	The type of the signal being measured, either pressure or temperature. Valid values are: PRESS (default) or TEMP.

2.17 Interface "ChannelPLCAB"

2.17.1 Description

2.17.1.1 This interface represents a channel in an Allan-Bradley PLC subsystem.

2.17.2 Design

2.17.2.1 This interface shall be a dispatch interface.

2.17.2.2 This interface shall be an automation interface.

2.17.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.17.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.17.3 Methods and Properties

2.17.3.1 Property Address

```
// Physical memory address
[propget, id(1)]
HRESULT Address([out, retval] long *pVal);
[propput, id(1)]
HRESULT Address([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the physical memory address on the PLC to which the channel belongs. The default shall be -1. If -1 is set, an error shall be generated. The valid address is from 0 to the maximum address defined in the *SubsystemPLC* hardware definition.

2.17.3.2 Property Bit

```
// Position in a 16 bit word. Boolean only.
[propget, id(2)]
HRESULT Bit([out, retval] long *pVal);
[propput, id(2)]
HRESULT Bit([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the position in a 16 bit word. The default shall be -1. If –1 is set and the channel data type is Boolean, an error shall be generated. The valid range is –1 for float channels or from 0 to 15 for Boolean channels.

2.18 Interface "ChannelPLCGE"

2.18.1 Description

2.18.1.1 This interface represents a channel in a GEFanuc PLC subsystem.

2.18.2 Design

2.18.2.1 This interface shall be a dispatch interface.

2.18.2.2 This interface shall be an automation interface.

2.18.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.18.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.18.3 Methods and Properties

2.18.3.1 Property Address

```
// Physical memory address
[propget, id(1)]
HRESULT Address([out, retval] long *pVal);
[propput, id(1)]
HRESULT Address([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the physical memory address on the PLC to which the channel belongs. The default shall be -1. If -1 is set, an error shall be generated. The valid address is from 1 to the maximum address defined in the *SubsystemPLC* hardware definition.

2.18.3.2 Property Slope

```
// The slope of a linear equation.
[propget, id(2)]
HRESULT Slope ([out, retval] float *pVal);
[propput, id(2)]
HRESULT Slope ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	the slope of the linear equation used to convert the input data received from the PLC. This is ignored for Boolean type channels and output channels. The default value shall be 1.0.

2.18.3.3 Property Offset

```
// The offset of a linear equation.
[propget, id(3)]
HRESULT Offset([out, retval] float *pVal);
[propput, id(3)]
HRESULT Offset([in] float newVal);
```

Argument Name	Description
*pVal, newVal	the offset of the linear equation used to convert the input data received from the PLC. This is ignored for Boolean type channels and output channels. The default value shall be 0.0.

2.19 Interface "ChannelRTP"

2.19.1 Description

2.19.1.1 This interface represents a channel of an RTP subsystem.

2.19.2 Design

2.19.2.1 This interface shall be a dispatch interface.

2.19.2.2 This interface shall be an automation interface.

2.19.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.19.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.19.3 Methods and Properties

2.19.3.1 Property HostName

```
[propget, id(1)]
HRESULT HostName ([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT HostName ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the host name of the RTP chassis that contains the channel. The default value shall be an empty string. If an empty string is set an error shall be generated. The hostname that is defined will be validated against the list of hostnames defined in the *SubsystemRTP* hardware definition.

2.19.3.2 Property TagName

```
[propget, id(2)]
HRESULT TagName ([out, retval] BSTR *pVal);
[propput, id(2)]
HRESULT TagName ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the tag name of the channel in the RTP system. The default value shall be an empty string. If this field is left as an empty string it will default to the channel name.

2.20 Interface "ChannelSCUTR"

2.20.1 Description

2.20.1.1 This interface will represent a single input channel for the Scanning UTR Subsystem.

2.20.2 Design

2.20.2.1 This interface shall be a dispatch interface.

2.20.2.2 This interface shall be an automation interface.

2.20.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.20.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.20.3 Methods and Properties

2.20.3.1 Property ChannelType

```
[propget, id(1)]
HRESULT ChannelType([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT ChannelType([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the channel type of the SCUTR channel. The default value shall be an empty string. It shall be one of *TC* (for thermocouple), *TC_RAW*, *RTD* (Resistance Temperature Device), *TRANSIENT* (for transient thermocouples) or *TRANS_RAW*. If any other value is set an error shall be generated.

2.20.3.2 Property SerialNumber

```
[propget, id(2)]
HRESULT SerialNumber([out, retval] BSTR *pVal);
[propput, id(2)]
HRESULT SerialNumber([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the serial number of the SCUTR box. The default value shall be an empty string. This value will be validated against the serial numbers defined in the *SubsystemSCUTR* hardware definition. If an empty string is set an error shall be generated.

2.20.3.3 Property ChannelNumber

```
[propget, id(3)]
HRESULT ChannelNumber([out, retval] long *pVal);
[propput, id(3)]
HRESULT ChannelNumber([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the channel number of the thermocouple within the SCUTR box or the RTD channel number. Its range shall be from 1 to 60 for Thermocouple type channels and 1 to 2 for Resistance Temperature Device type channels. The default value shall be 0. If a value out of range is set or the *ChannelNumber* is set to an already assigned channel number an error shall be generated.

2.20.3.4 Property SensorName

This property is obsolete. The new properties are generic and named CalSensor, Cal Group and FixedSensor. The FixedSensor should be used to store the thermocouple sensor.

```
// Sensor name (batch wire Id)
[propget, id(4)]
HRESULT SensorName([out, retval] BSTR *pVal);
[propput, id(4)]
HRESULT SensorName ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the name of the sensor for the thermocouple that is used to look up the break point table for engineering unit conversion. The default value shall be an empty string. If an empty string is set for a *TC* or *TRANSIENT* channel type, an error shall be generated.

2.21 Interface "ChannelTBDAU"

2.21.1 Description

2.21.1.1 This interface will represent a single input channel for the Test Bed Data Acquisition Unit Subsystem.

2.21.2 Design

2.21.2.1 This interface shall be a dispatch interface.

2.21.2.2 This interface shall be an automation interface.

2.21.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.21.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.21.3 Methods and Properties

2.21.3.1 Property Index

```
[propget, id(1)]
HRESULT Index([out, retval] long *pVal);
[propput, id(1)]
HRESULT Index([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the index of the channel in the master list, i.e. the list of all channels defined in a TBDAU subsystem. The default value shall be 0. Its range shall be from 1 to the maximum value defined in the *SubsystemTBDAU* hardware definition. If any other value is set an error shall be generated.

2.21.3.2 Property BitPosition

```
[propget, id(2)]
HRESULT BitPosition([out, retval] long *pVal);
[propput, id(2)]
HRESULT BitPosition([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the bit position in a 16-bit word for a Boolean channel. 1 represents the LSB and 16 the MSB. The default shall be 0. If the channel is not a Boolean channel, this parameter will be ignored. If any other value is set for a Boolean channel, an error shall be generated.

2.21.3.3 Property Slope

```
// Slope of the conversion function
[propget, id(3)]
HRESULT Slope([out, retval] float *pVal);
[propput, id(3)]
HRESULT Slope([in] float newVal);
```

Argument Name	Description
*pVal, newVal	the slope of the linear equation used to convert the input data received from the TBDAU. This is ignored for Boolean type channels. The default value shall be 1.0.

2.21.3.4 Property Offset

```
// Offset of the conversion function
[propget, id(4)]
HRESULT Offset([out, retval] float *pVal);
[propput, id(4)]
HRESULT Offset([in] float newVal);
```

Argument Name	Description
*pVal, newVal	the offset of the linear equation used to convert the input data received from the TBDAU. This is ignored for Boolean type channels. The default value shall be 0.0.

2.21.3.5 Property Min

```
// Minimum limit for the converted value
[propget, id(5)]
HRESULT Min([out, retval] float *pVal);
[propput, id(5)]
HRESULT Min([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	the valid minimum of the converted value. The default value shall be 0. This is ignored for Boolean type channels.

2.21.3.6 Property Max

```
// Maximum limit for the converted value
[propget, id(6)]
HRESULT Max([out, retval] float *pVal);
[propput, id(6)]
HRESULT Max([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	The valid maximum of the converted value. The default value shall be 0. This is ignored for Boolean type channels.

2.21.4 **Usage Conditions and Restrictions**

2.21.4.1 *Min* must be smaller than *Max*.

2.22 Interface "ChannelTDM"

2.22.1 Description

2.22.1.1 This interface will represent a single input channel for the TDM Subsystem.

2.22.2 Design

2.22.2.1 This interface shall be a dispatch interface.

2.22.2.2 This interface shall be an automation interface.

2.22.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.22.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.22.3 Methods and Properties

2.22.3.1 Property DeviceType

```
// Type of the TDM device
[propget, id(1)]
HRESULT DeviceType([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT DeviceType([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the type of the TDM device of the channel. The default value shall be an empty string. Its value shall be one of DIGBERT, PINCAHP or PINCAIP. If any other value is set, an error shall be generated.

2.22.3.2 Property DeviceName

```
// Name of the TDM device as found in the Calibration Database
[propget, id(2)]
HRESULT DeviceName([out, retval] BSTR *pVal);
[propput, id(2)]
HRESULT DeviceName ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the name of the TDM device as found in the calibration database. The default value shall be an empty string. If an empty string is set, an error shall be generated.

2.22.3.3 Property Number

```
// Channel Number
[propget, id(3)]
HRESULT Number([out, retval] long *pVal);
[propput, id(3)]
HRESULT Number([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the channel number of the TDM channel. The default value shall be 0. The range of the *Number* depends on the channel *Type*, i.e. from 1 to 104 for *TC*, from 1 to 128 for *TC_RAW*, from 1 to 8 for *RTD* and 1 for *PWRHI*, *PWRLO*, *PWROK* or *SYNC_LOCK* channel types. If any other value is set, an error shall be generated.

2.22.3.4 Property Type

```
// Channel Type
[propget, id(4)]
HRESULT Type([out, retval] BSTR *pVal);
[propput, id(4)]
HRESULT Type([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the channel type of the TDM channel. The default value shall be an empty string. The valid types are *TC*, *TC_RAW*, *RTD*, *PWRHI*, *PWRLO*, *PWROK* and *SYNC_LOCK*. If any other string is set, an error shall be generated.

2.22.3.5 Property SensorName

This property is obsolete. The new properties are generic and named CalSensor, Cal Group and FixedSensor. The FixedSensor should be used to store the thermocouple sensor.

```
[propget, id(5)]
HRESULT SensorName ([out, retval] BSTR *pVal);
[propput, id(5)]
HRESULT SensorName ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the calibration identifier of the thermocouple. The *SensorName* is only used for the *TC* channel type. The default value shall be an empty string. If an empty string is set and the channel type is *TC*, an error shall be generated.

2.23 Interface "ChannelThrustES"

2.23.1 Description

2.23.1.1 This interface will represent a single input channel for the Engine Specific Thrust Subsystem.

2.23.2 Design

2.23.2.1 This interface shall be a dispatch interface.

2.23.2.2 This interface shall be an automation interface.

2.23.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.23.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.23.3 Methods and Properties

2.23.3.1 Property SignalConditionerName

```
// Name of the signal conditioner
[propget, id(1)]
HRESULT SignalConditionerName([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT SignalConditionerName([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the name of the signal conditioner of the Thrust ES channel.

2.23.3.2 It shall be one of the strings:

- “WEI” (first working load cell signal conditioner using calibration and zero offset sensors)
- “WEI_RAW” (first working load cell signal conditioner without using calibration and zero offset sensors)
- “WEI2” (second working load cell signal conditioner using calibration and zero offset sensors)
- “WEI2_RAW” (second working load cell signal conditioner without using calibration and zero offset sensors)
- “TAD1” (first master load cell signal conditioner using only calibration sensor without zero offset sensor)
- “TAD2” (second master load cell signal conditioner using only calibration sensor without zero offset sensor)

- “TAD” (summation of the two master load cell signal conditioners TAD1 and TAD2 providing both have been defined).

2.23.3.2.1 The default value shall be an empty string. If an invalid string is set an error shall be generated. Note: only one channel of each type can be defined i.e. there is a maximum of 7 channels. The input will be validated against the definition for the *SubsystemThrustES* hardware.

2.23.3.3 Property CalibrationSensor

This property is obsolete. The new properties are generic and named CalSensor, Cal Group and FixedSensor. The CalSensor should be used to store the sensor for WEI and WEI2. The FixedSensor shall be used to store the sensor for TAD1 and TAD2.

```
// name of calibration sensor
[propget, id(2)]
HRESULT CalibrationSensor([out, retval] BSTR *pVal);
[propput, id(2)]
HRESULT CalibrationSensor([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the sensor name used to store the most recent calibration for the thrust channel (required for “TAD1”, TAD2”, “WEI” and “WEI2” signal conditioners). The default value shall be an empty string.

2.23.3.4 Property ZeroOffsetSensor

This property is obsolete. The new properties are generic and named CalSensor, Cal Group and FixedSensor. The FixedSensor shall be used to store the Zero Offset sensor.

```
// name of zero offset sensor
[propget, id(3)]
HRESULT ZeroOffsetSensor([out, retval] BSTR *pVal);
[propput, id(3)]
HRESULT ZeroOffsetSensor([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the sensor name used to store the zero offset for the thrust channel (required for “WEI” and “WEI2” signal conditioners). The default value shall be an empty string.

2.24 Interface "ChannelThrustNobel"

2.24.1 Description

2.24.1.1 This interface will represent a single input channel for the Nobel Thrust Subsystem.

2.24.2 Design

2.24.2.1 This interface shall be a dispatch interface.

2.24.2.2 This interface shall be an automation interface.

2.24.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.24.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.24.3 Methods and Properties

2.24.3.1 Property SignalConditionerName

```
// Name of the signal conditioner
[propget, id(1)]
HRESULT SignalConditionerName([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT SignalConditionerName([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the name of the signal conditioner of the Nobel Thrust channel.

2.24.3.1.1 It shall be one of the strings:

- “WEI” (first working load cell signal conditioner using calibration and zero offset sensors)
- “WEI_RAW” (first working load cell signal conditioner without using calibration and zero offset sensors)
- “WEI2” (second working load cell signal conditioner using calibration and zero offset sensors)
- “WEI2_RAW” (second working load cell signal conditioner without using calibration and zero offset sensors)
- “TAD1” (first master load cell signal conditioner using only calibration sensor without zero offset sensor)
- “TAD2” (second master load cell signal conditioner using only calibration sensor without zero offset sensor)

- “TAD” (summation of the two master load cell signal conditioners TAD1 and TAD2 providing both have been defined).

2.24.3.1.2 The default value shall be an empty string. If an invalid string is set an error shall be generated. Note: only one channel of each type can be defined i.e. there is a maximum of 7 channels. The input will be validated against the definition for the *SubsystemThrustNobel* hardware.

2.25 Interface "ChannelTruTemp"

2.25.1 Description

2.25.1.1 This interface will represent a single input channel for the Kaye TruTemp Temperature Measurement Subsystem.

2.25.2 Design

2.25.2.1 This interface shall be a dispatch interface.

2.25.2.2 This interface shall be an automation interface.

2.25.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.25.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.25.3 Methods and Properties

2.25.3.1 Property TemperatureType

```
[propget, id(1)]
HRESULT TemperatureType([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT TemperatureType([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	the channel type of the TruTemp channel.

2.25.3.1.1 It shall be one of the strings:

- “TC” (thermocouple temperature)
- “TC_RAW” (millivolts)
- “TC_COMP” (cold junction millivolts)
- “RTD” (temperature)
- “RTD_RAW” (resistance).

2.25.3.1.2 The default value shall be an empty string. If an invalid string is set an error shall be generated

2.25.3.2 Property RTDName

```
// name of external reference RTD
[propget, id(2)]
HRESULT RTDName([out, retval] BSTR *pVal);
[propput, id(2)]
HRESULT RTDName([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the name of the external reference RTD (resistance temperature device). The default value shall be an empty string, which means, no external RTD is used.

2.25.3.3 Property GatewayName

```
// name of gateway
[propget, id(3)]
HRESULT GatewayName([out, retval] BSTR *pVal);
[propput, id(3)]
HRESULT GatewayName([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the name of the gateway to which the scanner is assigned. The input will be validated against the list of Gateway definitions for the *SubsystemTruTemp* hardware. The default value shall be an empty string. If an invalid string is set an error shall be generated.

2.25.3.4 Property ScannerNumber

```
[propget, id(4)]
HRESULT ScannerNumber([out, retval] long *pVal);
[propput, id(4)]
HRESULT ScannerNumber ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the scanner (range from 1 to 239) connected with the specified gateway. The input will be validated against the list of Scanner definitions for the *SubsystemTruTemp* hardware. The default value shall be 0. If a value out of range is set or the *ScannerNumber* of an already assigned scanner is set an error shall be generated.

2.25.3.5 Property ChannelNumber

```
// channel number of thermocouple or RTD
[propget, id(5)]
HRESULT ChannelNumber([out, retval] long *pVal);
[propput, id(5)]
HRESULT ChannelNumber([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the thermocouple or RTD channel number for the scanner. The default value shall be 0, which means, the channel is not (yet) assigned to a thermocouple or an RTD. The *ChannelNumber* shall be in the range 1 – 20, if the *TemperatureType* is “TC”, “TC_RAW” or “TC_COMP” (because a maximum of 20 thermocouples can be connected within one scanner of a TruTemp subsystem). The *ChannelNumber* shall be in the range 1 – 4, if the *TemperatureType* is “RTD” or “RTD_RAW” (because a maximum of 5 thermocouples can be connected to one RTD in a scanner). If a value out of range is set or the *ChannelNumber* is set to an already assigned channel number an error shall be generated.

2.25.3.6 Property Range

```
// millivolt range of thermocouple
[propget, id(6)]
HRESULT Range([out, retval] long *pVal);
[propput, id(6)]
HRESULT Range([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	the millivolt range (value 30 or 60) of the thermocouple. The *Range* is only required, if *TemperatureType* is “TC”, “TC_RAW” or “TC_COMP”. The default value shall be 30. If a value other than 30 or 60 is set an error shall be generated.

2.25.3.7 Property SensorName

This property is obsolete. The new properties are generic and named CalSensor, Cal Group and FixedSensor. The FixedSensor shall be used to store thermocouple sensor.

```
// Sensor name to which the breakpoint table has been defined
[propget, id(7)]
HRESULT SensorName([out, retval] BSTR *pVal);
[propput, id(7)]
HRESULT SensorName ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	The sensor name used to convert millivolts to temperature. The *Sensor Name* is only required if *TemperatureType* is “TC” or “TC_COMP”. The default value shall be an empty string.

2.26 Interface “ChannelMSS”

2.26.1 Description

2.26.1.1 This interface will represent a single channel for the Mechanical Scanivalve subsystem.

2.26.2 Design

2.26.2.1 This interface shall be a dispatch interface.

2.26.2.2 This interface shall be an automation interface.

2.26.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.26.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.26.3 Methods and Properties

2.26.3.1 Property PortNumber

```
[proppget, id(1)]
HRESULT PortNumber([out, retval] long *pVal);
[propput, id(1)]
HRESULT PortNumber([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the port assigned to the channel being measured by the MSS. Values in the range 1 – 48 are valid. If a value outside the range is provided, an error will be generated. The default value is 0.

2.26.3.2 Property ModuleNumber

```
[propget, id(2)]
HRESULT ModuleNumber([out, retval] long *pVal);
[propput, id(2)]
HRESULT ModuleNumber([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the module via which the port is measured. The module number must be within the range 1 – 12 but it must also match a MSS subsystem module number that is defined in the SubsystemMSS hardware definition. If a value outside the range is given, an error will be generated. The default value is 0.

2.27 Interface "ChannelUTRH"

2.27.1 Description

2.27.1.1 This interface will represent a single channel for the UTR Half Box subsystem.

2.27.2 Design

2.27.2.1 This interface shall be a dispatch interface.

2.27.2.2 This interface shall be an automation interface.

2.27.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.27.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.27.3 Methods and Properties

2.27.3.1 Property HalfBoxName

```
[propget, id(1)]
HRESULT HalfBoxName([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT HalfBoxName([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	Specifies the name of the UTR Half Box. May be omitted if the property *IdentificationChannel* is defined. The default value is an empty string. If the box name is not empty and does not match a name in the SubsystemUTRH hardware definition, an error will be generated.

2.27.3.2 Property IdentificationChannel

```
[propget, id(2)]
HRESULT IdentificationChannel([out, retval] BSTR *pVal);
[propput, id(2)]
HRESULT IdentificationChannel([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the channel that provides the identification resistance measurement for this UTR Half Box. May be omitted if the property *HalfBoxName* is defined. The default value is an empty string. If the specified channel does not exist, an error will be generated.

2.27.3.3 Property BridgeInputVoltageChannel

```
[propget, id(3)]
HRESULT BridgeInputVoltageChannel([out, retval] BSTR *pVal);
[propput, id(3)]
HRESULT BridgeInputVoltageChannel([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the channel that measures the bridge input voltage. The default value is an empty string. If no channel is provided or the provided channel does not exist, an error will be generated.

2.27.3.4 Property BridgeOutputVoltageChannel

```
[propget, id(4)]
HRESULT BridgeOutputVoltageChannel([out, retval] BSTR *pVal);
[propput, id(4)]
HRESULT BridgeOutputVoltageChannel([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the channel that measures the bridge output voltage. The default value is an empty string. If no channel is provided or the provided channel does not exist, an error will be generated.

2.27.3.5 Property TemperatureUnit

```
[propget, id(3)]
HRESULT TemperatureUnit([out, retval] BSTR *pVal);
[propput, id(3)]
HRESULT TemperatureUnit([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the engineering unit for the RTD temperature measurement. Valid values are DegC for degrees Celsius, DegF for degrees Fahrenheit, DegK for degrees Kelvin or DegR for degrees Rankine.

2.27.4 **Usage Conditions and Restrictions**

2.27.4.1 Either *HalfBoxName* or *IdentificationChannel* must be defined. If neither one is defined an error will be generated.

2.28 Interface “ChannelDPS”

2.28.1 Description

2.28.1.1 This interface will represent a single channel for the DPS subsystem.

2.28.2 Design

2.28.2.1 This interface shall be a dispatch interface.

2.28.2.2 This interface shall be an automation interface.

2.28.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.28.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.28.3 Methods and Properties

2.28.3.1 Property ScannerName

```
[propget, id(1)]
HRESULT ScannerName([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT ScannerName([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the host name of the DSA Pressure Scanner. The default is an empty string. This value will be validated against the list of scanner names defined in the SubsystemDPS hardware definition. If the name is left empty an error will be generated.

2.28.3.2 Property Port

```
[propget, id(2)] HRESULT Port([out, retval] long *pVal);
[propput, id(2)] HRESULT Port([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the pressure port number on the scanner. The default is 0. The valid range is from 1 to 16. If a value outside this range is defined, an error will be generated.

2.28.3.3 Property AmbientPressureChannel

```
[propget, id(3)]
HRESULT AmbientPressureChannel ([out, retval] BSTR *pVal);
[propput, id(3)]
HRESULT AmbientPressureChannel ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	Optional parameter specifying the name of the ambient pressure channel in order to read pressures as absolute. The default value is an empty string.

2.29 Interface “ChannelCONSORT”

2.29.1 Description

2.29.1.1 This interface will represent a single channel for the CONSORT subsystem.

2.29.2 Design

2.29.2.1 This interface shall be a dispatch interface.

2.29.2.2 This interface shall be an automation interface.

2.29.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.29.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.29.3 Methods and Properties

2.29.3.1 Property Label

```
[propget, id(1)] HRESULT Label([out, retval] long *pVal);
[propput, id(1)] HRESULT Label([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the unique identifier for the data word in octal. The default is -1. The valid range is from 0o to 377o. If a value outside the range is defined, an error will be generated.

2.29.3.2 Property Bit

```
[propget, id(2)] HRESULT Bit([out, retval] long *pVal);
[propput, id(2)] HRESULT Bit([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Defines the bit number in the Status or Maintenance word. This is required for Boolean channels only. Valid values are from 14 to 29. The default is 0. If a value less than 14 or greater than 29 is specified, an error will be generated.

2.29.3.3 Property Lane

```
[propget, id(3)] HRESULT Lane([out, retval] BSTR *pVal);
[propput, id(3)] HRESULT Lane([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	Defines the Lane identifier. The default is an empty string. Valid values are *A* or *B*. If any other value is specified, an error will be generated.

2.29.3.4 Property Factor

```
[propget, id(4)] HRESULT Factor([out, retval] float *pVal);
[propput, id(4)] HRESULT Factor([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	Defines the factor for shifting the decimal point when processing SCALED INTEGER data or scale factor when processing RAW DATA. The default value is 1.0. This is only required if the channel is not a Boolean.

2.30 Interface “ChannelCEC”

2.30.1 Description

2.30.1.1 This interface will represent a single channel for the CEC subsystem.

2.30.2 Design

2.30.2.1 This interface shall be a dispatch interface.

2.30.2.2 This interface shall be an automation interface.

2.30.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.30.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.30.3 Methods and Properties

2.30.3.1 Property Pinout

```
[propget] HRESULT Pinout ([out, retval] BSTR* pVal);
[propput] HRESULT Pinout([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	Name of the pinout of a CEC subsystem to which the channel is assigned. The pinout must be unique for all the channels of the subsystem. The pinout must be defined in the *SubsystemCEC* hardware definition. The pinout will indicate whether the channel being configured belongs to an Amplifier card or to a Tachometer card. Only the last three properties (2.30.3.31 to 2.30.3.33) are applicable for a Tachometer channel. All the remaining properties are only applicable for an Amplifier channel (2.30.3.2 to 2.30.3.30). The default value will be an empty string.

2.30.3.2 Property `InputType`

```
[propget] HRESULT InputType ([out, retval] BSTR* pVal);
[propput] HRESULT InputType([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	The type of the input transducer. Valid values are *V* (Velocity Coil, self-generating), *S* (Acceleration, single-ended), *D* (Acceleration, differential), *VC* (Velocity, constant current), *VV* (Velocity, voltage excitation), *AC* (Acceleration, constant current), *AV* (Acceleration, voltage excitation), *VBP* (Velocity, backplane route), *ABP* (Acceleration , backplane route). The default value will be *D*.

2.30.3.3 Property `EngineeringUnitType`

```
[propget]
HRESULT EngineeringUnitType ([out, retval] BSTR* pVal);
[propput] HRESULT EngineeringUnitType([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	The engineering unit type for output. Valid values are *IMP* (imperial) and *MET* (metric). The default value will be *IMP*. Imperial units are as follows: • displacement mode – mils • velocity mode – ips • acceleration mode - g’s Metric units are as follows: • displacement mode – mm • velocity mode – cm/sec • acceleration mode – m/sec^2

2.30.3.4 Property `TransducerScaling`

```
[propget]
HRESULT TransducerScaling ([out, retval] BSTR* pVal);
[propput] HRESULT TransducerScaling([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	The scaling applied to the input transducer units. Valid values are *RMS*, *AVG* (average), *Pk* (Peak) and *PTP* (Peak-to-Peak). The default value will be *Pk*.

2.30.3.5 Property Sensitivity

```
[propget] HRESULT Sensitivity ([out, retval] float* pVal);
[propput] HRESULT Sensitivity([in] float newVal);
```

Argument Name	Description
*pVal, newVal	The sensitivity of the input transducer. Valid values are 6 to 1100 mV/ips for input type *V*, 1 to 110 pc/g for input type *S*, 1 to 110 pc/g for input type *D*, 6 to 1100 mV/ips for input type *VC*, 6 to 1100 mV/ips for input type VV, 1 to 110 mV/g for input type *AC* and 1 to 110 mV/g for input type *AV*. For other input types this property is ignored. The default value will be 0.

2.30.3.6 Property FilterMode

```
[propget] HRESULT FilterMode ([out, retval] BSTR* pVal);
[propput] HRESULT FilterMode([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	The type of the filter to be applied to the channel. Valid values are *N* (None), *F* (Fixed Bandpass), *V* (Variable Bandpass), *T* (Tracking Filter), *S* (Sweep Filter), *B* (Balance Mode). The default value will be *N*.

2.30.3.7 Property BackPlaneNumber

```
[propget] HRESULT BackPlaneNumber ([out, retval] long* pVal);
[propput] HRESULT BackPlaneNumber([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Route on backplane for signal. This field defines the backplane output for a physical input channel, or a backplane input for a backplane input channel. This must be unique for the driving channel. This is only required if communication using the backplane is required. Valid values are from 1 to 8. The default value value will be 0.

2.30.3.8 Property `IsTransducerFilterOn`

```
[propget]
HRESULT IsTransducerFilterOn ([out, retval] BOOL* pVal);
[propput] HRESULT IsTransducerFilterOn([in] BOOL newVal);
```

Argument Name	Description
*pVal, newVal	Flag indicating whether the 2 pole, 16 kHz transducer filter is to be turned on. It is only required for input types *S* and *B*. The default value will be *true*.

2.30.3.9 Property `OutputMode`

```
[propget] HRESULT OutputMode ([out, retval] BSTR* pVal);
[propput] HRESULT OutputMode([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	The output signal type, defining the integration. Valid values are *V[ips]* (Velocity, imperial), *A[g]* (Acceleration, imperial), *D[mils]* (Displacement, imperial), *V[cm/se]* (Velocity, metric), *A[m/s/s]* (Acceleration, metric) and *D[mm]* (Displacement, metric). The default value will be *V[ips]*. If the input type is a velocity, only output types *V* and *D* are permitted. If the input type is a displacement, only output type *D* is permitted.

2.30.3.10 Property `OutputRange`

```
[propget] HRESULT OutputRange ([out, retval] float* pVal);
[propput] HRESULT OutputRange([in] float newVal);
```

Argument Name	Description
*pVal, newVal	The maximum output value in engineering units. Valid values are from 0 to 200. The default value will be -1.

2.30.3.11 Property `OutputScaling`

```
[propget]
HRESULT OutputScaling ([out, retval] BSTR* pVal);
[propput] HRESULT OutputScaling([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	The scaling applied to the output units. Valid values are *RMS*, *AVG* (average), *Pk* (Peak) and *PTP* (Peak-to-Peak). The default value will be *Pk*.

2.30.3.12 Property `OutputRangeAC`

```
[propget] HRESULT OutputRangeAC ([out, retval] float* pVal);
[propput] HRESULT OutputRangeAC([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	AC voltage output maximum range in Volts. Valid values are from 0 to 10 V pk. The default value will be 1.

2.30.3.13 Property `OutputRangeDC`

```
[propget] HRESULT OutputRangeDC ([out, retval] float* pVal);
[propput] HRESULT OutputRangeDC([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	DC voltage output maximum range in Volts. Valid values are from 0 to 10. The default value will be 10.

2.30.3.14 Property `FixedFilterTable`

```
[propget]
HRESULT FixedFilterTable ([out, retval] BSTR* pVal);
[propput] HRESULT FixedFilterTable ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	Selects the filter table to use when selecting the low pass and high pass filter cut-off frequencies for a fixed filter. It is only required for filter mode *F*. Valid values are *A (5 Hz – 5 kHz)* or *B (1 kHz – 25 kHz)*. The default value will be an empty string.

2.30.3.15 Property `LowPassFreq`

```
[propget] HRESULT LowPassFreq ([out, retval] long* pVal);
[propput] HRESULT LowPassFreq([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	Low pass cut-off frequency for fixed filter in Hz. It is only required for filter mode *F*. Valid values are 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 225, 250, 275, 300, 325, 350, 375, 400, 425, 450, 475, 500, 600, 700, 800, 900, 1000, 1250, 1500, 1750, 2000, 2250, 2500, 2750, 3000, 3250, 3500, 3750, 4000, 4250, 4500, 4750, 5000 if fixed filter table *A* is selected, and 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000, 11000, 12000, 13000, 14000, 15000, 16000, 17000, 18000, 19000, 20000, 21000, 22000, 23000, 24000, 25000 if fixed filter table *B* is selected. The default value will be 0.

2.30.3.16 Property `HighPassFreq`

```
[propget] HRESULT HighPassFreq ([out, retval] long* pVal);
[propput] HRESULT HighPassFreq([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	High pass cut-off frequency for fixed filter in Hz. It is only required for filter mode *F*. Valid values are 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 250, 300, 350, 400, 450, 500, 600, 700, 800, 900, 1000 if fixed filter table *A* is selected, and 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000, 11000, 12000, 13000, 14000, 15000, 16000, 17000, 18000, 19000, 20000, 21000, 22000, 23000, 24000, 25000 if fixed filter table *B* is selected. The default value will be 0.

2.30.3.17 Property `FilterBandWidthType`

```
[propget]
HRESULT FilterBandWidthType ([out, retval] BSTR* pVal);
[propput] HRESULT FilterBandWidthType([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	Filter bandwidth type. It is only required for the filter modes *V*, *T*, *S* and *B*. Valid values are *F* (fixed) and *P* (percentage). The default value will be an empty string.

2.30.3.18 Property `FilterBandWidth`

```
[propget] HRESULT FilterBandWidth ([out, retval] long* pVal);
[propput] HRESULT FilterBandWidth([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Filter bandwidth value. It is only required for the filter modes *V*, *T*, *S* and *B*. Valid values are from 1 to 100 for filter bandwidth type *F* and 1 to 5 for filter bandwidth type *P*. The default value will be 0.

2.30.3.19 Property `FreqOutMin`

```
[propget] HRESULT FreqOutMin ([out, retval] float* pVal);
[propput] HRESULT FreqOutMin([in] float newVal);
```

Argument Name	Description
*pVal, newVal	Frequency at minimum DC voltage (0 V) on output. It is only required for the filter modes *V*, *T*, *S* and *B*. Valid values are positive. The default value will be 0.

2.30.3.20 Property `FreqOutMax`

```
[propget] HRESULT FreqOutMax ([out, retval] float* pVal);
[propput] HRESULT FreqOutMax([in] float newVal);
```

Argument Name	Description
*pVal, newVal	Frequency at maximum DC voltage (10 V) on output. It is only required for the filter modes *V*, *T*, *S* and *B*. Valid values are positive. The default value will be 0.

2.30.3.21 Property `VariableFilterFreq`

```
[propget]
HRESULT VariableFilterFreq ([out, retval] float* pVal);
[propput] HRESULT VariableFilterFreq([in] float newVal);
```

Argument Name	Description
*pVal, newVal	Centre frequency for variable filter in Hz. It is only required for filter mode *V*. Valid values are from 5 to 5000. The default value will be 0.

2.30.3.22 Property `FilterBandWidthMin`

```
[propget]
HRESULT FilterBandWidthMin ([out, retval] long* pVal);
[propput] HRESULT FilterBandWidthMin([in] long newVal);
```

Argument Name	Description
*pVal, newVal	The minimum bandwidth value in Hz when the bandwidth is calculated as a percentage of the center frequency. It is only required for filter modes *T*, *S* and *B* and for filter bandwidth type *P*. Valid values are from 1 to 100. The default value will be 0.

2.30.3.23 Property `FilterBandWidthMax`

```
[propget]
HRESULT FilterBandWidthMax ([out, retval] long* pVal);
[propput] HRESULT FilterBandWidthMax([in] long newVal);
```

Argument Name	Description
*pVal, newVal	The maximum bandwidth value in Hz when the bandwidth is calculated as a percentage of the center frequency. It is only required for filter modes *T*, *S* and *B* and for filter bandwidth type *P*. Valid values are from 1 to 100. The default value will be 0.

2.30.3.24 Property TachInputNumber

```
[propget] HRESULT TachInputNumber ([out, retval] long* pVal);
[propput] HRESULT TachInputNumber([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Tach input channel number assigned to channel. It is only required for filter modes *T* and *B*. Valid values are from 1 to 4. The default value will be 0.

2.30.3.25 Property TrackingFilterOrder

```
[propget]
HRESULT TrackingFilterOrder ([out, retval] long* pVal);
[propput] HRESULT TrackingFilterOrder([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Tracking filter order. This is the tracking filter multiple. The tracking filter frequency is set at the RPM rate times this number. It is only required for filter modes *T* and *B*. Valid values are from 1 to 4. The default value will be 0.

2.30.3.26 Property SweepFreqStep

```
[propget] HRESULT SweepFreqStep ([out, retval] float* pVal);
[propput] HRESULT SweepFreqStep([in] float newVal);
```

Argument Name	Description
*pVal, newVal	Frequency step in Hz. It is only required for filter mode *S*. Valid values are from 1 to 100. The default value will be 0.

2.30.3.27 Property `SweepFreqDelay`

```
[propget] HRESULT SweepFreqDelay ([out, retval] long* pVal);
[propput] HRESULT SweepFreqDelay([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	Frequency delay between frequency steps in ms. It is only required for filter mode *S*. Valid values are from 1 to 32767. The default value will be 0.

2.30.3.28 Property `SweepType`

```
[propget] HRESULT SweepType ([out, retval] BSTR* pVal);
[propput] HRESULT SweepType([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	Type of continuous sweep to be performed. It is only required for filter mode *S*. Valid values are *UP* (from *SweepFreqMin* to *SweepFreqMax*) and *DOWN* (from *SweepFreqMax* to *SweepFreqMin*). The default value will be an empty string.

2.30.3.29 Property `SweepFreqMin`

```
[propget] HRESULT SweepFreqMin ([out, retval] long* pVal);
[propput] HRESULT SweepFreqMin ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	Minimum sweep frequency value in Hz. It is only required for filter mode *S*. Valid values are from 5 to 5000. The default value will be 0.

2.30.3.30 Property `SweepFreqMax`

```
[propget] HRESULT SweepFreqMax ([out, retval] long* pVal);
[propput] HRESULT SweepFreqMax([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Maximum sweep frequency value in Hz. It is only required for filter mode *S*. Valid values are from 5 to 5000. The default value will be 0.

2.30.3.31 Property `TachChannelNumber`

```
[propget] HRESULT TachChannelNumber ([out, retval] long* pVal);
[propput] HRESULT TachChannelNumber([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Tachometer number indicating that the tachometer is enabled. Valid values are from 1 to 4. The default value will be 0.

2.30.3.32 Property `TachRatio`

```
[propget] HRESULT TachRatio ([out, retval] float* pVal);
[propput] HRESULT TachRatio([in] float newVal);
```

Argument Name	Description
*pVal, newVal	Tach ratio as the number of pulses per one revolution.Valid values are from 0.01 to 9999.99. The default value will be 0.

2.30.3.33 Property TachEdge

```
[propget] HRESULT TachEdge([out, retval] BSTR* pVal);
[propput] HRESULT TachEdge ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	Specifies which edge of the tach signal will be used as the trigger. Valid values are *LEAD* and *TRAIL*. The default value will be *LEAD*.

2.30.4 Usage Conditions and Restrictions

2.30.4.1 The OutputMode of channels in the same chassis must be either all imperial or all metric.

2.31 Interface "ChannelTSM"

2.31.1 Description

2.31.1.1 This interface will represent a single channel for the TSM subsystem.

2.31.2 Design

2.31.2.1 This interface shall be a dispatch interface.

2.31.2.2 This interface shall be an automation interface.

2.31.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.31.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.31.3 Methods and Properties

2.31.3.1 Property `SerialNumber`

```
[propget] HRESULT SerialNumber ([out, retval] long *pVal);
[propput] HRESULT SerialNumber([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	Serial number of the TSM scanner. Valid values are the serial numbers defined in the *SubsystemTSM* hardware definition. The default value will be 0.

2.31.3.2 Property `ChannelNumber`

```
[propget] HRESULT ChannelNumber ([out, retval] long *pVal);
[propput] HRESULT ChannelNumber([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	Channel number on the scanner. Valid values are from 1 to 16. The default value will be 0.

2.31.3.3 Property EngineeringUnitOutput

```
[propget]
HRESULT EngineeringUnitOutput ([out, retval] BSTR* pVal);
[propput] HRESULT EngineeringUnitOutput([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	Output engineering unit. Valid values and default values are defined in the table below (c.f. 2.31.4.10).

2.31.3.4 Property EngineeringUnitFilter

```
[propget]
HRESULT EngineeringUnitFilter ([out, retval] long* pVal);
[propput] HRESULT EngineeringUnitFilter([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	Weighting factor (n) for internal data filter, using the formula EU_delivered = EU_old + EU_new/n – EU_old/n. Values > 15 produce slow step response and should be used with caution. Valid values are from 1 to 100. The default value will be 1.

~~2.31.3.5 Property Sensor~~

This property is obsolete. The new properties are generic and named CalSensor, CalGroup and FixedSensor. The new generic FixedSensor property is used to store the thermocouple or RTD sensor.

```
[propget] HRESULT Sensor ([out, retval] BSTR* pVal);
[propput] HRESULT Sensor([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Name of sensor used to convert a channel data to account for batchwire, thermocouple, or RTD calibration. Limited to 40 character alphanumeric, (case sensitive). The default is an empty string. Must be specified if the channel Input Type is a T/C, User Defined or Smart Connector, and the EU_OUT is specified to be mV. Note: Can be of type sensor_name.aux_address or sensor_name. (c.f. 2.31.4.8)

2.31.3.6 Property Input Type

```
[propget] HRESULT InputType ([out, retval] BSTR* pVal);
[propput] HRESULT InputType([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Type of input signal on this channel. Valid values and default values are defined in the table below (c.f. 2.31.4.9).

2.31.4 Usage Conditions and Restrictions

2.31.4.1 The valid values and the default value for the *InputType* property depend on the *Type* property of the corresponding TSMModule defined in the *SubsystemTSM* hardware definition. The TSMModule is identified by the *ChannelNumber* property referring to the TSMScanner identified by the *SerialNumber* property.

2.31.4.2 Only one channel may be defined for a given channel number, except UTR and VoltB channels.

2.31.4.3 A UTR channel must have a thermocouple channel with the same TSMModule and update rate.

2.31.4.4 If a channel Input Type of VoltB is specified for a channel, then the corresponding channel with the same TSMModule and an Input Type of Volts, Volt1, Volt2, Volt3 or Volt4 must also be defined.

2.31.4.5 The same channel number for a scanner can be used to define a T/C, User Defined or Smart Connector channel Input Type and a UTR channel Input Type. In this case, both channels must be defined with the same scan rate.

2.31.4.6 The same channel number for a scanner can be used to define a Volts, Volt1, Volt2, Volt3 or Volt4 channel Input Type and a VoltB channel Input Type. In this case, both channels must be defined with the same scan rate.

2.31.4.7 If a channel Input Type of T/C, User Defined or Smart Connector is specified for a channel, and the EU Out is defined to be mV, then the corresponding UTR channel Input Type must also be defined.

2.31.4.8 The FixedSensor field can be entered as only a sensor name (without the auxiliary address) or as a combined sensor name and auxiliary address of the form sensor_name.aux_address.

2.31.4.9 Acceptable INPUT_TYPE values:

Type property of corresponding TSMModule	**InputType**
Thermocouple (TC)	Type K T/C (K) – Default Type N T/C (N) Type J T/C (J) Type T T/C (T) Type B T/C (B) Type E T/C (E) Type R T/C (R) Type S T/C (S) UTR (UTR)
Type K T/C (K)	Type K T/C (K) – Default UTR (UTR)
Type N T/C (N)	Type N T/C (N) – Default UTR (UTR)
Type J T/C (J)	Type J T/C (J) – Default UTR (UTR)
Type T T/C (T)	Type T T/C (T) – Default UTR (UTR)
Type B T/C (B)	Type B T/C (B) – Default UTR (UTR)
Type E T/C (E)	Type E T/C (E) – Default UTR (UTR)
Type R T/C (R)	Type R T/C (R) - Default UTR (UTR)
Type S T/C (S)	Type S T/C (S) - Default UTR (UTR)
Volts (V) (Volt1, Volt2, Volt3, Volt4 and VoltB are only applicable for 9146 scanners)	Volts (V) – Default Volt1 (V1) Volt2 (V2) Volt3 (V3) Volt4 (V4) VoltB (Vb) RTD385 (R385) RTD7990 (R7990) Resistance 220 Ohms (220) Resistance 440 Ohms (440) Resistance 990 Ohms (990) Resistance 19.8 kOhms (19800) 2.25k Thermistor (2kT) 5k Thermistor (5kT) 10k Thermistor (10kT)

Type property of corresponding TSMModule	**InputType**
User Defined (U) (only for 9146 scanners)	User Defined (UDF) – Default Smart Connector (SCON) UTR (UTR)

2.31.4.10 Acceptable EU_OUT values:

INPUT_TYPE	**Acceptable EU_OUT**
V, V1, V2, V3, V4, Vb	mV
R385, R7990, 2kT, 5kT, 10kT	C - Default F K R
K, N, J, T, B, E, R, S, UTR, UDF, SCON	C - Default F K R mV
220, 440, 990, 19800	NULL

2.32 Interface "ChannelM1553"

2.32.1 Description

2.32.1.1 This interface will represent a single channel for the M1553B subsystem.

2.32.2 Design

2.32.2.1 This interface shall be a dispatch interface.

2.32.2.2 This interface shall be an automation interface.

2.32.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.32.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.32.3 **Methods and Properties**

2.32.3.1 Property Message

```
[propget, id(1)]
HRESULT Message ([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT Message ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	The name of the message as defined in the M1553B subsystem.

2.32.3.2 Property StartWord

```
[propget, id(2)]
HRESULT StartWord ([out, retval] long *pVal);
[propput, id(2)]
HRESULT StartWord ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	The number of the start word within the message. Numbering starts with 1.

2.32.3.3 Property StartBit

```
[propget, id(3)]
HRESULT StartBit ([out, retval] long *pVal);
[propput, id(3)]
HRESULT StartBit ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	The number of the start bit within the start word. Numbering starts with 1.

2.32.3.4 Property SignificantBits

```
[propget, id(4)]
HRESULT SignificantBits ([out, retval] long *pVal);
```

```
[propput, id(4)]
HRESULT SignificantBits ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	The number of the bits for the channel. This should normally be 1 for a discrete channel and up to 32 for a float channel.

2.32.3.5 Property Encoding

```
[propget, id(5)]
HRESULT Encoding ([out, retval] BSTR *pVal);
[propput, id(5)]
HRESULT Encoding ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	One of the following: BNR, UINT, BCD, FLT, DIS and ASCII. BNR, UINT, BCD and FLT coding apply to float channels. DIS coding applies to discrete channels. ASCII coding can be associated to any input channel, but not to output channels.

2.32.3.6 Property Slope

```
[propget, id(6)]
HRESULT Slope ([out, retval] float *pVal);
[propput, id(6)]
HRESULT Slope ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	This field is required for a subset of float channels, and will be ignored for discrete channels. This field is required for BNR, UINT and BCD coding only.

2.32.3.7 Property Offset

```
[propget, id(7)]
HRESULT Offset ([out, retval] float *pVal);
[propput, id(7)]
HRESULT Offset ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	This field is required for a subset of float channels, and will be ignored for discrete channels. This field is required for BNR, UINT and BCD coding only.

2.33 Interface "ChannelThg"

2.33.1 Description

2.33.1.1 This interface will represent a single channel for the THG Subsystem.

2.33.2 Design

2.33.2.1 This interface shall be a dispatch interface.

2.33.2.2 This interface shall be an automation interface.

2.33.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.33.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.33.3 Methods and Properties

2.33.3.1 Property SignalConditionerName

```
// Name of the signal conditioner
[propget, id(1)]
HRESULT SignalConditionerName([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT SignalConditionerName([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the name of the signal conditioner of the THG channel.

2.33.3.2 It shall be one of the strings:

- „WRK1“ (first working load cell signal conditioner)
- „WRK2“ (second working load cell signal conditioner)
- „WRK3“ (third working load cell signal conditioner)
- „WRK4“ (fourth working load cell signal conditioner)

- „MSTF1“ (first forward master load cell signal conditioner)
- „MSTF2“ (second forward master load cell signal conditioner)
- „MSTR1“ (first reverse master load cell signal conditioner)
- „MSTR2“ (second reverse master load cell signal conditioner)
- „WRK_MEAS“ (necessary when there is a working load cell)
- „WRK_MEAS_TC“ (optional)
- „WRK_CORR“ (necessary when there is a working load cell)
- „WRK_CORR_AVG“ (optional)
- „MST_FORW“ (necessary when there is a forward master load cell)
- „MST_REV“ (necessary when there is a -reverse master load cell)

2.33.3.2.1 The default value for the *get* property shall be an empty string. If an invalid string is set an error shall be generated. Note: only one channel of each type can be defined i.e. there is a maximum of 14 channels. The input will be validated against the definition for the *SubsystemThg* hardware.

2.33.3.3 Property CalibrationSensor

This property is obsolete. The new properties are generic and named CalSensor, Cal Group and FixedSensor. The CalSensor shall be used to store the calibration sensor for the WRK_CORR signal conditioner, and the FixedSensor shall be used to store the calibration sensor for the WRK1, WRK2, WRK3, WRK4, MSTF1, MSTF2, MSTR1 and MSTR2 signal conditioner.

```
// name of calibration sensor
[propget, id(2)]
HRESULT CalibrationSensor([out, retval] BSTR *pVal);
[propput, id(2)]
HRESULT CalibrationSensor([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the sensor name used to store the most recent calibration for the THG channel. The default value shall be an empty string.

2.33.3.4 Property ZeroOffsetSensor

This property is obsolete. The new properties are generic and named CalSensor, CalGroup and FixedSensor. The FixedSensor shall be used to store the Zero Offset sensor for the WRK_CORR signal conditioner.

```
// name of zero offset sensor
[propget, id(3)]
HRESULT ZeroOffsetSensor([out, retval] BSTR *pVal);
[propput, id(3)]
```

```
HRESULT ZeroOffsetSensor([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the sensor name used to store the zero offset for the THG channel. The default value shall be an empty string.

2.34 Interface "ChannelPLCOmron"

2.34.1 Description

2.34.1.1 This interface represents a channel in an Omron PLC subsystem.

2.34.2 Design

2.34.2.1 This interface shall be a dispatch interface.

2.34.2.2 This interface shall be an automation interface.

2.34.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.34.2.4 This interface shall include the properties of the **Channel** interface plus those mentioned below.

2.34.3 Methods and Properties

2.34.3.1 Property Address

```
// Physical memory address
[propget, id(1)]
HRESULT Address([out, retval] long *pVal);
[propput, id(1)]
HRESULT Address([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the physical memory address on the PLC to which the channel belongs. The default value for the *get* property shall be -1. If -1 is set, an error shall be generated. The valid address is from 1 to the maximum address defined in the *SubsystemPLC* hardware definition.

2.35 Interface "ChannelOPC"

2.35.1 Description

2.35.1.1 This interface will represent a single channel for the OPC subsystem.

2.35.2 Design

2.35.2.1 This interface shall be a dispatch interface.

2.35.2.2 This interface shall be an automation interface.

2.35.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.35.2.4 This interface shall include the properties of the Channel interface plus those mentioned below.

2.35.3 Methods and Properties

2.35.3.1 Property Prefix

```
[propget, id(3010)]
HRESULT Prefix ([out, retval] BSTR *pVal);
[propput, id(3010)]
HRESULT Prefix ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	The name of the prefix as defined in the OPC subsystem.

2.35.3.2 Property Path

```
[propget, id(3011)]
HRESULT Path ([out, retval] BSTR *pVal);
[propput, id(3011)]
HRESULT Path ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	The name of the path as defined in the OPC subsystem.

2.35.3.3 Property Tag

```
[propget, id(3012)]
HRESULT Tag ([out, retval] BSTR *pVal);
[propput, id(3012)]
HRESULT Tag ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	The name of the tag as defined in the OPC subsystem.

2.36 Interface "ChannelVEXA"

2.36.1 Description

2.36.1.1 This interface will represent a single channel for the VEXA subsystem.

2.36.2 Design

2.36.2.1 This interface shall be a dispatch interface.

2.36.2.2 This interface shall be an automation interface.

2.36.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.36.2.4 The interface *Channel* shall be implemented as part of this interface by aggregation.

2.36.3 Methods and Properties

2.36.3.1 Property HostName

```
[propget, id(4001)]
HRESULT HostName ([out, retval] BSTR *pVal);
[propput, id(4001)]
HRESULT HostName ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	Network host name of the device. Must be defined for the subsystem.

2.36.3.2 Property Category

```
[propget, id(4002)]
HRESULT Category ([out, retval] BSTR *pVal);
[propput, id(4002)]
HRESULT Category ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Category/type of channel on the device: • Analog input = 0 (voltage or thermocouple input) • CJC = 1 • DIO = 2

2.36.3.3 Property ChannelNumber

```
[propget, id(4003)]
HRESULT ChannelNumber ([out, retval] BSTR *pVal);
[propput, id(4003)]
HRESULT ChannelNumber ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Channel number on the device. Depending on property *Category* the valid values are: • Analog input : 0 - 47 • CJC : 0 - 11 • DIO : 0 – 7 Note: The total number of available CJC channels varies depending on the device model type. (refer to 2.36.3.14)

2.36.3.4 Property OutputEu

```
[propget, id(4004)]
HRESULT OutputEu ([out, retval] BSTR *pVal);
[propput, id(4004)]
HRESULT OutputEu ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	Final engineering unit conversion applied to the value by the subsystem. This property is only applicable if property *Category* is “analog input” and property *AiIsThermocouple* is set

2.36.3.5 Property AiIsThermocouple

```
[propget, id(4011)]
HRESULT AiIsThermocouple ([out, retval] BOOL *pVal);
```

Argument Name	**Description**
*pVal, newVal	Flag whether channel is a thermocouple input. Only applicable if property *Category* is “analog input”

2.36.3.6 Property AiThermocoupleType

```
[propget, id(4012)]
HRESULT AiThermocoupleType ([out, retval] long *pVal);
[propput, id(4012)]
HRESULT AiThermocoupleType ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	Type of thermocouple input. Valid values are: • Type B = 66 • Type E = 69 • Type J = 74 • Type K = 75 • Type N = 78 • Type R = 82 • Type S = 83 • Type T = 84 (The ASCII code is used). This property is only applicable if property *Category* is “analog input” and property *AiIsThermocouple* is set

2.36.3.7 Property AiFilterFrequency

```
[propget, id(4013)]
HRESULT AiFilterFrequency ([out, retval] long *pVal);
[propput, id(4013)]
HRESULT AiFilterFrequency ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	Low pass filter cut-off frequency. Only applicable if property *Category* is “analog input”. The default value shall be 4.

2.36.3.8 Property AiVoltageRange

```
[propget, id(4014)]
HRESULT AiVoltageRange ([out, retval] float *pVal);
[propput, id(4014)]
HRESULT AiVoltageRange ([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	Input voltage range. Only applicable if property *Category* is “analog input”. Valid values and default values are defined in the table below (2.36.3.13).

2.36.3.9 Property AiOtdEnabled

```
[propget, id(4015)]
HRESULT AiOtdEnabled ([out, retval] BOOL *pVal);
[propput, id(4015)]
HRESULT AiOtdEnabled ([in] BOOL newVal);
```

Argument Name	**Description**
*pVal, newVal	Open thermocouple detection flag. Only applicable if property *Category* is “analog input” and property *AiIsThermocouple* is set

2.36.3.10 Method GetVoltsSlopeAndOffset AiSlope

```
[id(4030)]
HRESULT GetVoltsSlopeAndOffset([out] float* pValSlope,[out]
float* pValOffset)
```

Argument Name	**Description**
*pValSlope	EU conversion slope determined by the signal conditioner. Returns ‘1’ if no signal conditioner is defined. Only applicable if property *Category* is “analog input” and property *AiIsThermocouple* is not set.
*pValOffset	EU conversion offset determined by the signal conditioner. Returns ‘0’ if no signal conditioner is defined. Only applicable if property *Category* is “analog input” and property *AiIsThermocouple* is not set.

2.36.3.11 Property DioStaticLevel

```
[propget, id(4021)]
HRESULT DioStaticLevel ([out, retval] BOOL *pVal);
[propput, id(4021)]
HRESULT DioStaticLevel ([in] BOOL newVal);
```

Argument Name	**Description**
*pVal, newVal	Level of OFF state: • TRUE : OFF state is high • FALSE : OFF state is low (default) Only applicable if property *Category* is “DIO” and the channel is an output channel.

2.36.3.12 Property DioOutputLatch

```
[propget, id(4022)]
HRESULT DioOutputLatch ([out, retval] BOOL *pVal);
[propput, id(4022)]
HRESULT DioOutputLatch ([in] BOOL newVal);
```

Argument Name	**Description**
*pVal, newVal	● TRUE : constant output (default) ● FALSE : pulsed output Only applicable if property *Category* is "DIO" and the channel is an output channel.

2.36.3.13 Acceptable AiVoltageRange values:

Channel Type	**Input Range**
Physical thermocouple defined as Thermocouple	± 67 mV (default)
Physical thermocouple defined as Voltage	± 67 mV (default) ± 100 mV
Physical voltage input	± 10 mV ± 67 mV ± 100 mV ± 1 V ± 10 V

Note: refer to the subsystem configuration for the physical type and the defined channel type (of analog inputs)

2.36.3.14 Valid numbers of CJC channels depending on the device model type

Model	Applicable Channels	CJC Channel Number	
		(external)	(internal)
EX1000A	0 to 15 16 to 31 32 to 47	0 4 8	- - -
EX1016A	0 to 15 16 to 31	0 4	- -

	32 to 35	-	8
	36 to 39	-	9
	40 to 43	-	10
	44 to 47	-	11
EX1032A	0 to 15	0	-
	16 to 19	-	4
	20 to 23	-	5
	24 to 27	-	6
	28 to 31	-	7
	32 to 35	-	8
	36 to 39	-	9
	40 to 43	-	10
	44 to 47	-	11
EX1000A-TC	0 to 3	-	0
EX1048A	4 to 7	-	1
	8 to 11	-	2
	12 to 15	-	3
	16 to 19	-	4
	20 to 23	-	5
	24 to 27	-	6
	28 to 31	-	7
	32 to 35	-	8
	36 to 39	-	9
	40 to 43	-	10
	44 to 47	-	11

2.37 Interface "ChannelDTS3250"

2.37.1 Description

2.37.1.1 This interface will represent a single channel for the DTS3250 subsystem.

2.37.2 Design

2.37.2.1 This interface shall be a dispatch interface.

2.37.2.2 This interface shall be an automation interface.

2.37.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Channels*.

2.37.2.4 The interface *Channel* shall be implemented as part of this interface by aggregation.

2.37.3 Methods and Properties

2.37.3.1 Property HostName

```
[propget, id(4001)]
HRESULT HostName ([out, retval] BSTR *pVal);
[propput, id(4001)]
HRESULT HostName ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Network host name of the device. Must be defined for the subsystem.

2.37.3.2 Property ChannelCategory

```
[propget, id(4002)]
HRESULT ChannelCategory ([out, retval] long *pVal);
[propput, id(4002)]
HRESULT ChannelCategory ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Category/type of channel on the device: • Thermocouple = 0 • RTD = 1

2.37.3.3 Property ChannelNumber

```
[propget, id(4003)]
HRESULT ChannelNumber ([out, retval] long *pVal);
[propput, id(4003)]
HRESULT ChannelNumber ([in] long newVal);
```

Argument Name	Description

*pVal, newVal	Channel number on the device. Depending on property ChannelType the valid values are: • Thermocouple : 1 to NUM_CHANNELS as defined in the Subsystem Editor • RTD : 1 to NUM_CHANNELS/8 Note: The method *GetNumberOfChannels* of the *DTS3250Device* interface can be used to determine the maximum number of channels depending on the channel type (Thermocouple or RTD).

2.37.3.4 Property OutputEu

```
[propget, id(4004)]
HRESULT OutputEu ([out, retval] BSTR *pVal);
[propput, id(4004)]
HRESULT OutputEu ([in] BSTR newVal);
```

Argument Name	**Description**
*pVal, newVal	Final engineering unit conversion applied to the value by the subsystem. This property is only applicable if property *ChannelType* is “Thermocouple” The default value is the *DefaultEu* from the selected Device.

2.37.3.5 Property ThermocoupleType

```
[propget, id(4011)]
HRESULT ThermocoupleType ([out, retval] long *pVal);
[propput, id(4011)]
HRESULT ThermocoupleType ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Type of thermocouple input. Valid values are: • ‘B’ = 66 • ‘E’ = 69 • ‘J’ = 74 • ‘K’ = 75 • ‘N’ = 78 • ‘R’ = 82 • ‘S’ = 83 • ‘T’ = 84 (The ASCII code is used). This property is only applicable if property *ChannelType* is “Thermocouple”

2.37.3.6 Property LowLimit

```
[propget, id(4012)]
HRESULT LowLimit ([out, retval] float *pVal);
[propput, id(4012)]
HRESULT LowLimit ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	Low limit value for the channel. Valid values are any value lower than property *HighLimit* and within the overall range of the thermocouple type. Note: The method *GetLowLimit* of the *SubsystemDTS3250* interface can be used to determine the values for the default and overall limits. Note: The LowLimit value is returned in the selected OutputEu.

2.37.3.7 Property HighLimit

```
[propget, id(4013)]
HRESULT HighLimit ([out, retval] float *pVal);
[propput, id(4013)]
HRESULT HighLimit ([in] float newVal);
```

Argument Name	**Description**
*pVal, newVal	High limit value for the channel. Valid values are any value higher than property *LowLimit* and within the overall range of the thermocouple type. Note: The method *GetHighLimit* of the *SubsystemDTS3250* interface can be used to determine the values for the default and overall limits. Note: The HighLimit value is returned in the selected OutputEu.

2.37.3.8 Property ShieldSW

```
[propget, id(4014)]
HRESULT ShieldSW ([out, retval] long *pVal);
[propput, id(4014)]
HRESULT ShieldSW ([in] long newVal);
```

Argument Name	**Description**
*pVal, newVal	Connection method for thermocouple shield: • 0 : OPEN • 1 : CLOSED (default) • 2 : NA (automatically selected if property *ChannelType* is "RTD"