



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	2.1.2, 2.69 - 2.78, 2.79	Added M1553B and THG subsystems.
4	Aug 21, 2009	Martin Hadaller	2.19.3.1	Correct available chassis numbers to 1 - 3.
4	Aug 24, 2009	Martin Hadaller	2.1.2, 2.43	PLC Omron added
5	Mar 01, 2010	Martin Hadaller	2.66 2.68	Updated ITSMScanner, ITSMModule
5	Mar 03, 2010	Martin Hadaller	2.1.2 Figure 1 2.85	Added OPC subsystem
5	Apr 21, 2010	Peter Buchholz	2.1.2 Figure 1 2.87 - 2.93	Added VEXA subsystem
5	Jul 05, 2010	Martin Hadaller	2.89.3.3 2.89.3.4 2.91.3.6	Updated Added
5	Jul 28, 2010	Martin Hadaller	2.54.3.2	Updated

5	Aug 23, 2010	Martin Hadaller	2.1.2 Figure 1 2.94 - 2.96	Added DTS3250 subsystem
5	Sep 23, 2010	Martin Hadaller	2.69.3.2 2.69.3.3 2.69.3.4 2.69.3.5	Added M1553 properties
5	Sep 29, 2010	Martin Hadaller	2.27.3.5	Added property PBSBrick::ContPurgeEnable
5	Nov 02, 2010	Martin Hadaller	2.94.3.2 2.94.3.3	Updated DTS3250 methods GetLowLimit, GetHighLimit
5	Nov 23, 2010	Martin Hadaller	2.1.3 2.27.3.5 2.82 2.86.3.2 2.87.4.1 2.88.1.1 2.91.3.2	Reviewed view MDS (FM, RS)
5	Apr 05, 2011	Martin Hadaller	2.25.3.2 2.27.3.1 2.27.3.6 2.27.3.7 2.29.3.1	Added new PBS properties
5	Jul 15, 2011	Martin Hadaller	0	Added new SubsystemPBS method GetMaxPortNumber

TABLE OF CONTENTS

1.	INTRODUCTION	1
1.1	Purpose.....	1
1.2	Scope.....	1
1.3	Applicable Documents.....	1
1.4	Codes and Standards	1
1.5	Abbreviations and Definitions	2
2.	DESIGN	4
2.1	Introduction.....	4
2.2	Interface "SubsystemArinc429"	8
2.3	Interface "ArincPinouts"	9
2.4	Interface "ArincPinout"	11
2.5	Interface "SubsystemAVM"	14
2.6	Interface "AVMModules"	15
2.7	Interface "AVMModule"	17
2.8	Interface "SubsystemDDS"	19
2.9	Interface "DDSDevices"	21
2.10	Interface "DDSDevice"	23
2.11	Interface "DDSControlCHBERTs"	25
2.12	Interface "DDSControlCHBERT"	27
2.13	Interface "DDSControlSTINCAs"	30
2.14	Interface "DDSControlSTINCA"	32
2.15	Interface "SubsystemGASS"	34
2.16	Interface "GASSGroups"	36
2.17	Interface "GASSGroup"	38
2.18	Interface "GASSCards"	40

2.19	Interface "GASSCard"	42
2.20	Interface "GASSSCPs"	45
2.21	Interface "GASSSCP"	47
2.22	Interface "GASSPinouts"	49
2.23	Interface "GASSPinout"	51
2.24	Interface "SubsystemHSS"	54
2.25	Interface "SubsystemPBS"	55
2.26	Interface "PBSBricks"	58
2.27	Interface "PBSBrick"	60
2.28	Interface "PBSPorts"	63
2.29	Interface "PBSPort"	65
2.30	Interface "SubsystemRTP"	67
2.31	Interface "SubsystemSCUTR"	68
2.32	Interface "SCUTRBoxes"	69
2.33	Interface "SCUTRBox"	71
2.34	Interface "SubsystemTruTemp"	73
2.35	Interface "TruTempGateways"	74
2.36	Interface "TruTempGateway"	76
2.37	Interface "TruTempUnits"	78
2.38	Interface "TruTempUnit"	80
2.39	Interface "TruTempScanners"	82
2.40	Interface "TruTempScanner"	84
2.41	Interface "SubsystemHyScan"	85
2.42	Interface "SubsystemModbus"	86
2.43	Interface "SubsystemPLC"	89
2.44	Interface "SubsystemTBDAU"	92

2.45	Interface "SubsystemThrust"	93
2.46	Interface "Subsystem External"	95
2.47	Interface "SubsystemMSS"	97
2.48	Interface "MSSModules"	102
2.49	Interface "MSSModule"	104
2.50	Interface "MSSPorts"	106
2.51	Interface "MSSPort"	108
2.52	Interface "SubsystemUTRH"	110
2.53	Interface "UTRHalfBoxes"	111
2.54	Interface "UTRHalfBox"	113
2.55	Interface "SubsystemDPS"	116
2.56	Interface "DPSScanners"	117
2.57	Interface "DPSScanner"	119
2.58	Interface "DPSPorts"	122
2.59	Interface "DPSPort"	124
2.60	Interface "SubsystemCONSORT"	126
2.61	Interface "SubsystemCEC"	128
2.62	Interface "CECPinouts"	129
2.63	Interface "CECPinout"	131
2.64	Interface "SubsystemTSM"	133
2.65	Interface "TSMScanners"	134
2.66	Interface "TSMScanner"	136
2.67	Interface "TSMModules"	139
2.68	Interface "TSMModule"	141
2.69	Interface "SubsystemM1553"	143
2.70	Interface "M1553Networks"	145

2.71	Interface "M1553Network"	147
2.72	Interface "M1553Nodes"	149
2.73	Interface "M1553Node"	151
2.74	Interface "M1553Messages"	154
2.75	Interface "M1553Message"	156
2.76	Interface "M1553MessageBc"	158
2.77	Interface "M1553MessageBm"	160
2.78	Interface "M1553MessageRt"	162
2.79	Interface "SubsystemThg"	163
2.80	Interface "SubsystemOPC"	164
2.81	Interface "OPCPrefixes"	165
2.82	Interface "OPCPrefix"	166
2.83	Interface "OPCPaths"	168
2.84	Interface "OPCPath"	169
2.85	Interface "OPCTags"	170
2.86	Interface "OPCTag"	171
2.87	Interface "SubsystemVEXA"	172
2.88	Interface "VEXADevices"	174
2.89	Interface "VEXADevice"	175
2.90	Interface "VEXAAAnalogInputs"	178
2.91	Interface "VEXAAAnalogInput"	179
2.92	Interface "VEXASignalConditionerModels"	181
2.93	Interface "VEXASignalConditionerModel"	181
2.94	Interface "SubsystemDTS3250"	184
2.95	Interface "DTS3250Devices"	186
2.96	Interface "DTS3250Device"	187

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-Subsystems.

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 Subsystems 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

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 Totalizer 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
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
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 Boxes
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 This section describes the design of the Subsystems interface and its specific sub-interfaces.

2.1.2 A subsystem interface is obtained by the *New* property of the Subsystems interface described in the Interface Control Document for the Configuration Server. The valid values for the Type parameter of the property are listed in the table below.

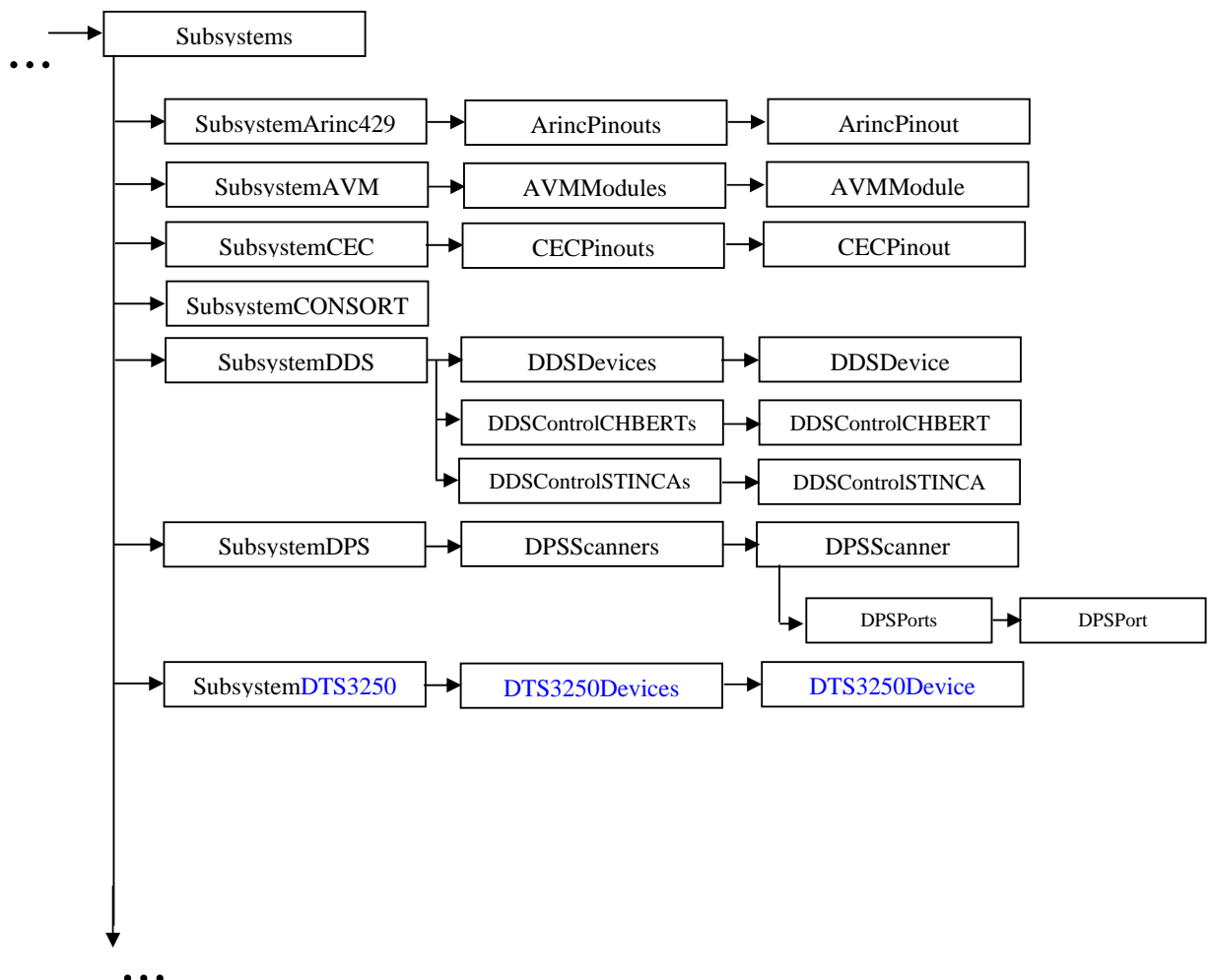
Type	Returned Interface
Calculated	Subsystem
External	SubsystemExternal
Arinc429	SubsystemArinc429
ATH	Subsystem
AVM	SubsystemAVM
CEC	SubsystemCEC
CONSORT	SubsystemCONSORT
DDS_AVM*	SubsystemDDS
DDS	SubsystemDDS
DDTC	SubsystemDDS
DPS	SubsystemDPS
DTS3250	Subsystem DTS3250
GASSAI	SubsystemGASS
GASSAO	SubsystemGASS
GASSDIO	SubsystemGASS
GASSTC	SubsystemGASS
GASSVXI	SubsystemGASS
HSS	SubsystemHSS
HSV	SubsystemGASS
HyScan	SubsystemHyScan
IRIGB	Subsystem
M1553B	SubsystemM1553
ModBusSerial	SubsystemModbus
ModBusEthernet	SubsystemModbus
MSS	SubsystemMSS
OPC	Subsystem OPC
PBS	SubsystemPBS
PLCAB	SubsystemPLC
PLCGE	SubsystemPLC
PLCOmron	SubsystemPLC
PWM	Subsystem
SCUTR	SubsystemSCUTR
RTP	SubsystemRTP
TBDAU	SubsystemTBDAU

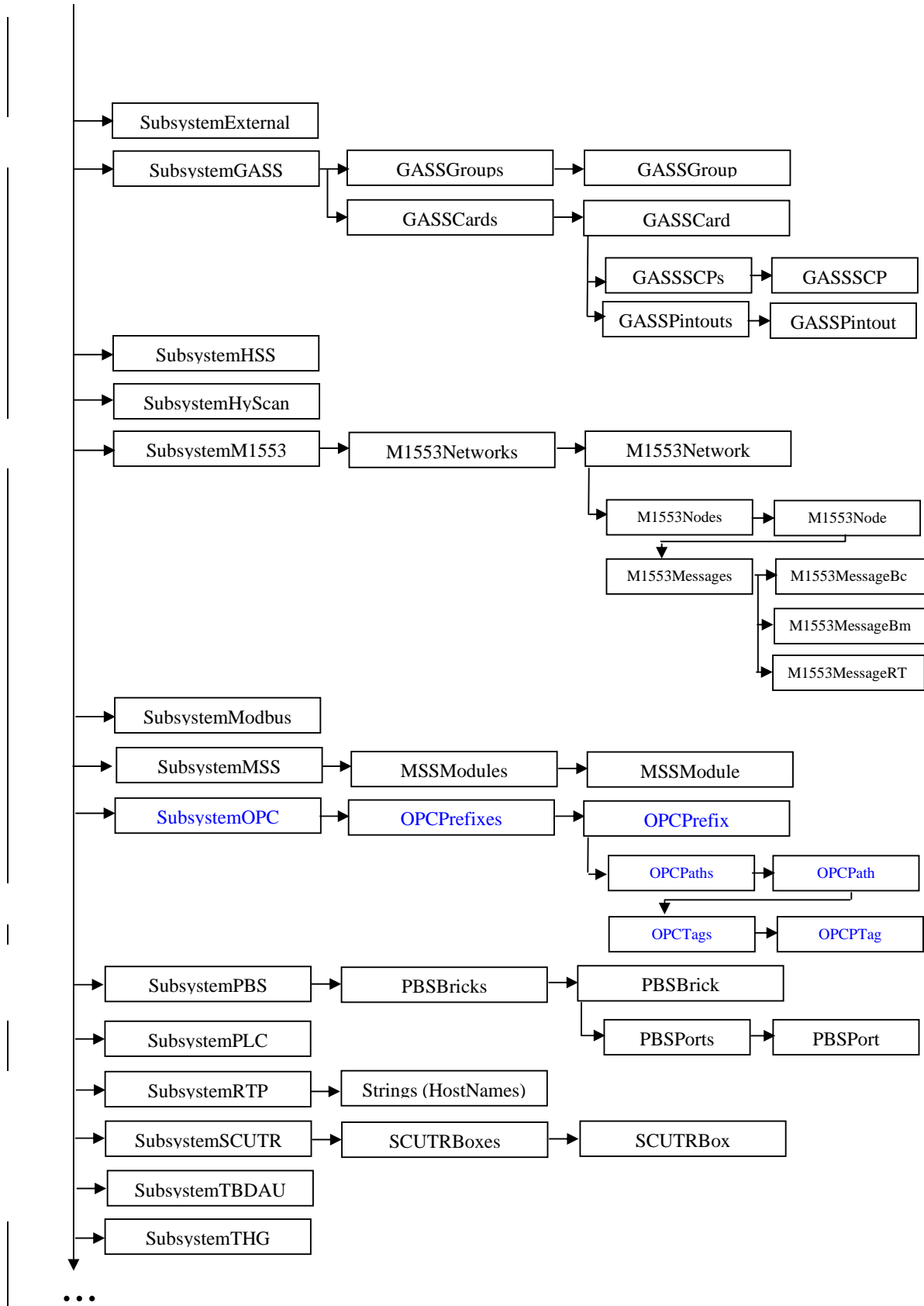
Type	Returned Interface
TDM	Subsystem
THG	SubsystemTHG
ThrustES	SubsystemThrust
ThrustNobel	SubsystemThrust
Thrust55	Subsystem
TruTemp	SubsystemTruTemp
TSM	SubsystemTSM
UTRH	SubsystemUTRH
VEXA	SubsystemVEXA

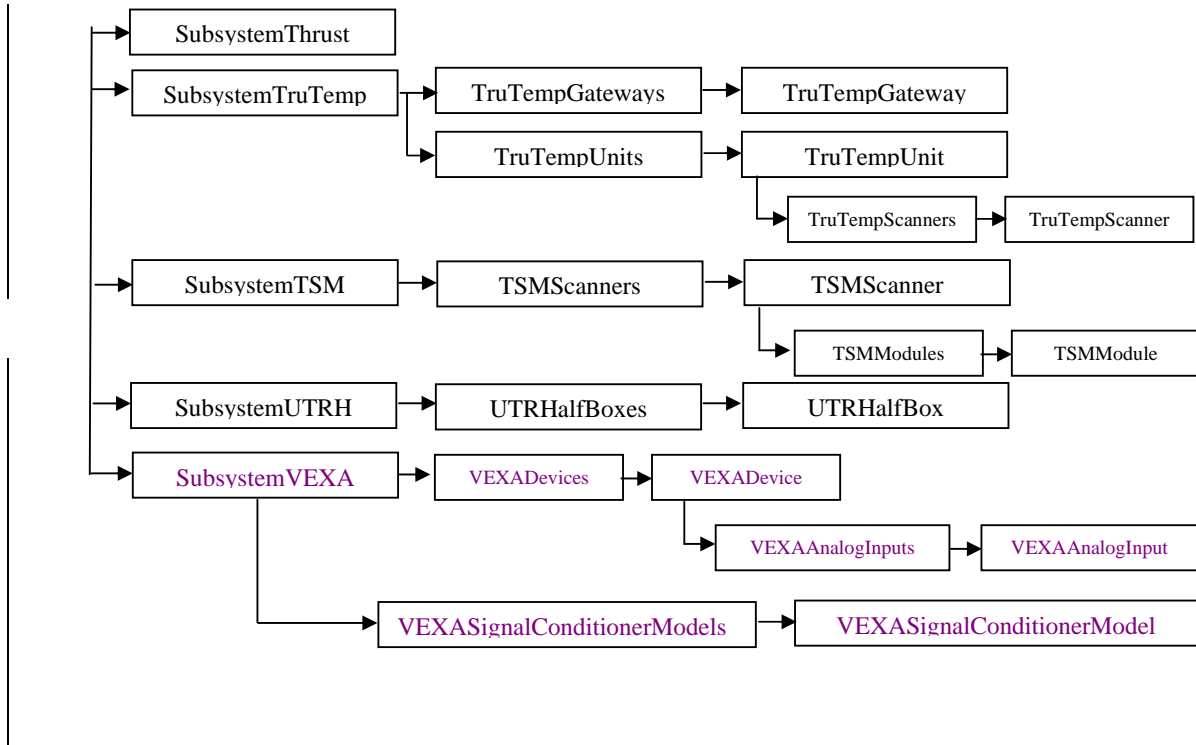
* Vibration/TFIA channels.

2.1.3

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 Subsystem interface is described in the Interface Control Document for the Configuration Server.





**Figure 1: Object Model for Subsystems**

2.2 Interface "SubsystemArinc429"

2.2.1 Description

2.2.1.1 This interface represents the hardware definition 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 *Subsystems*.

2.2.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.2.3 Methods and Properties

2.2.3.1 Property Pinouts

```
// Access to pinouts  
[propget, id(1)]  
HRESULT Pinouts ([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	returns the collection of pinouts of the ARINC subsystem.

2.3 Interface "ArincPinouts"

2.3.1 Description

2.3.1.1 This interface represents the pinouts of an ARINC 429 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 implement the collection interface for *LPDISPATCH*, returning the *ArincPinout* interfaces in the collection.

2.3.2.4 This interface shall not be directly created. It shall be created by the *Pinouts* property of the interface *SubsystemArinc429*.

2.3.3 Methods and Properties

2.3.3.1 Property Find

```
// Find a data record.  
[propget, id(1)]  
HRESULT Find([in] BSTR Name, [out, retval] long *pVal);
```

Argument Name	Description
Name	The Arinc429 Pinout Name to be found.
*pVal	the index of the ArincPinout with name <i>Name</i> , if it is found in the collection, 0 otherwise.

2.3.3.2 Property New

```
// Create a data record.  
[propget, id(1003)]  
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	return the index of the newly created item.

2.3.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.4 Interface "ArincPinout"

2.4.1 Description

2.4.1.1 This interface represents a pinout of an ARINC 429 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 *ArincPinouts*.

2.4.3 Methods and Properties

2.4.3.1 Property Name

```
// Name (Pinout) of the data record  
[propget, id(1)]  
HRESULT Name([out, retval] BSTR *pVal);  
[propput, id(1)]  
HRESULT Name([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the name of the pinout. It shall be a unique identifier given to each connection point. The default value shall be an empty string. If an empty string is set or the name is set to the name of another pinout within the list of pinouts, an error shall be generated.

2.4.3.2 Property Port

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

Argument Name	Description
*pVal, newVal	the port number on the Arinc card. Receive ports shall be designated 1 through 8, while transmit ports shall be designated 1 to 4. The default value shall be 0. If 0 is set, an error shall be generated.

2.4.3.3 Property Source

```
// Source i.e. transfer direction: Input or Output
[propget, id(3)]
HRESULT Source([out, retval] BSTR *pVal);
[propput, id(3)]
HRESULT Source([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the direction of data transfer for the port with possible values <i>I</i> for input and <i>O</i> for Output. Input means data transfer from the EEC to the RTE. The default value shall be an empty string. If an empty string is set, an error shall be generated.

2.4.3.4 Property Speed

```
// Speed High or Low. This shall be the same for all ports in
// the same group
[propget, id(4)]
HRESULT Speed([out, retval] BSTR *pVal);
[propput, id(4)]
HRESULT Speed([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the port speed with possible values <i>L</i> for Low and <i>H</i> for High. The default value shall be an empty string. If an empty string is set, an error shall be generated.

2.4.3.4.1 This must be the same for all channels of the same port. Furthermore, the speeds for channels on adjacent input ports (1 and 2, 3 and 4 ... 7 and 8) must also be the same.

2.4.3.5 Property GroupNumber

```
// Get group number
[propget, id(6)]
HRESULT GroupNumber([out, retval] long *pVal);
```

Argument Name	Description
*pVal	the group number for each transmission direction. The <i>GroupNumber</i> for input channels is ceil (port/2) and is equal to port for the output channels i.e. 1, 2, 3 or 4.

2.4.3.6 Property IsChanged

```
// Flag whether the data have been changed
[propget, id(2000)]
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.5 Interface "SubsystemAVM"

2.5.1 Description

2.5.1.1 This interface describes an Endevco AVM 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 *Subsystems*.

2.5.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.5.3 Methods and Properties

2.5.3.1 Property Modules

```
// Access to the collection of modules  
[propget, id(1)]  
HRESULT Modules ([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	the collection of modules of the AVM subsystem.

2.6 Interface "AVMModules"

2.6.1 Description

2.6.1.1 This interface represents the collection of modules of an Endeveco AVM 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 implement the collection interface for *LPDISPATCH*, returning the *AVMModule* interfaces in the collection.

2.6.2.4 This interface shall not be directly created. It shall be created by the *Modules* property of the interface *SubsystemAVM*.

2.6.3 Methods and Properties

2.6.3.1 Property Find

```
// Find a data record.  
[propget, id(1)]  
HRESULT Find([in] long Chassis, [in] long Slot,  
             [out, retval] long *pVal);
```

Argument Name	Description
Chassis	The Chassis number to be found.
Slot	The Slot number to be found.
*pVal	the index of the AVMModule with the chassis with number <i>Chassis</i> and the slot with number <i>Slot</i> , if it is found in the collection, and 0 otherwise.

2.6.3.2 Property New

```
// Create a data record.  
[propget, id(1003)]  
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created item.

2.6.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.7 Interface "AVMModule"**2.7.1 Description**

2.7.1.1 This interface represents the hardware definition of an AVM subsystem.

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 *AVMModules*.

2.7.3 Methods and Properties**2.7.3.1 Property CardType**

```
// Module i.e. card type
[propget, id(1)]
HRESULT CardType([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT CardType([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the type of module used i.e. Tracking Filter, Vibration Charge Amps or Tracking Filter Input Assembly. The default value shall be an empty string. The corresponding values are:TF, Vib and TFIA. If any other string is set, an error shall be generated.

2.7.3.2 Property Chassis

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

Argument Name	Description
*pVal, newVal	the Endevco chassis number beginning with 1. The chassis slot combination shall be unique. The default value shall be -1. If -1 is set, an error shall be generated.

2.7.3.3 Property Slot

```
// Slot i.e. position in chassis
[propget, id(3)]
HRESULT Slot([out, retval] long *pVal);
[propput, id(3)]
HRESULT Slot([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the slot number in the Endevco chassis beginning with 0. The default value shall be -1. The valid range shall be 1 to 12. If -1 is set, an error shall be generated.

2.7.3.4 Property IsChanged

```
// Flag whether the data have been changed
[propget, id(2000)]
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.7.4 Usage Conditions and Restrictions

- 2.7.4.1 The first slot: Slot 0 is reserved for the controller and it shall not be inserted anywhere else.
- 2.7.4.2 Tracking Filter cards can only be located in slots 1 and 2. If there are no tracking filters in the system, only vibration charge amplifiers shall be available to slots 1 to 12.
- 2.7.4.3 If a Tracking Filter card exists in slot 1 or slot 2, slots 3 and 4 must be empty. Slots 5 through 12 may have a Vibration Charge Amp or a Tracking Filter Input Assembly.
- 2.7.4.4 Any slot, except slot 0, may be left empty.

2.8 Interface "SubsystemDDS"**2.8.1 Description**

2.8.1.1 This interface will represent the DDS subsystem and define the file name for the DDS, DDTC and DDS_AVM subsystems, as well as the devices belonging to these subsystems.

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 *Subsystems*.

2.8.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.8.3 Methods and Properties**2.8.3.1 Property FileName**

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

Argument Name	Description
*pVal, newVal	the name of the DDS configuration file from which the channel configuration for the subsystem can be generated. The property shall be set to the default value i.e. an empty string.

2.8.3.2 Property Devices

```
// Access to the collection of devices
[propget, id(3)]
HRESULT Devices([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	the collection of devices for the DDS subsystem.

2.8.3.3 Property ControlCHBERTs

```
// Access to the collection of CHBERT device controls  
[propget, id(4)]  
HRESULT ControlCHBERTs([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	the collection of CHBERT Controls for the DDTC component of the DDS subsystem.

2.8.3.4 Property ControlSTINCAs

```
// Access to the collection of STINCA device controls  
[propget, id(5)]  
HRESULT ControlSTINCAs([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	the collection of STINCA Controls for the DDTC component of the DDS subsystem.

2.9 Interface "DDSDevices"**2.9.1 Description**

2.9.1.1 This interface describes the devices for the DDS subsystem.

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 *Devices* property of the interface *SubsystemDDS*.

2.9.2.4 This interface shall implement the collection interface for *LPDISPATCH*, returning the *DDSDevice* interfaces in the collection.

2.9.3 Methods and Properties**2.9.3.1 Property Find**

```
// Find a data record.
[propget, id(1)] HRESULT Find([in] BSTR Name, [in] long Port,
                             [out, retval] long *pVal);
```

Argument Name	Description
Name	The name of the DDSDevice to be found.
Port	The port number of the DDSDevice to be found.
*pVal	the index of the DDSDevice with the name <i>Name</i> and the port <i>Port</i> , if it is found in the collection, and 0 otherwise.

2.9.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>DDSDevice</i> .

2.9.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.10 Interface "DDSDevice"**2.10.1 Description**

2.10.1.1 This interface represents a DDS device.

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 *DDSDevices*.

2.10.3 Methods and Properties**2.10.3.1 Property Name**

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

Argument Name	Description
*pVal, newVal	a unique identifier for the DDS Device. The default value shall be an empty string.

2.10.3.2 Property Type

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

Argument Name	Description
*pVal, newVal	the device type. The valid types are PINCAIP, PINCAHP, STINCA, DIGBERT, CHBERT and AVM for the Tracked DDS. The default value shall be an empty string.

2.10.3.3 Property Port

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

Argument Name	Description
*pVal, newVal	the HIPPI Interface unit port number. This must be a number greater than 0. The default value shall be 0.

2.10.3.4 Property DataStream

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

Argument Name	Description
*pVal, newVal	defines which data stream is connected to the <i>Port</i> . The value could be A or B and this value is required for the STINCA, PINCAIP and PINCAHP device types. The default value shall be an empty string.

2.10.3.5 Property IsChanged

```
// Flag whether the data have been changed
[propget, id(2000)]
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.11 Interface "DDSControlCHBERTs"**2.11.1 Description**

2.11.1.1 This interface describes the CHBERT Controls for the DDTC component of the DDS subsystem.

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 *ControlCHBERTs* property of the interface *SubsystemDDS*.

2.11.2.4 This interface shall implement the collection interface for *LPDISPATCH*, returning the *DDSControlCHBERT* interfaces in the collection.

2.11.3 Methods and Properties**2.11.3.1 Property Find**

```
// Find a data record.
[propget, id(1)]
HRESULT Find([in] BSTR Name, [in] long ChannelNumber,
             [out, retval] long *pVal);
```

Argument Name	Description
Name	The name of the CHBERT Device to be found.
ChannelNumber	The channel number on the CHBERT Device to be found.
*pVal	the index of the <i>DDSControlCHBERT</i> with the name <i>Name</i> and the channel number <i>ChannelNumber</i> , if it is found in the collection, and 0 otherwise.

2.11.3.2 Property New

```
// Create a data record.  
[propget, id(1003)]  
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>DDSControlCHBERT</i> .

2.11.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.12 Interface "DDSControlCHBERT"**2.12.1 Description**

2.12.1.1 This interface represents a DDS Control CHBERT channel.

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 *DDSControlCHBERTs*.

2.12.3 Methods and Properties**2.12.3.1 Property Name**

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

Argument Name	Description
*pVal, newVal	a unique identifier for the DDS Control CHBERT device. 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. 2.9), an error shall be generated.

2.12.3.2 Property ChannelNumber

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

Argument Name	Description
*pVal, newVal	the channel number on the device. The default value shall be set to 0. The valid value ranges from 1 to 24. If the value is set outside of this range, an error shall be generated.

2.12.3.3 Property Mode

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

Argument Name	Description
*pVal, newVal	The mode associated to the channel i.e. Acceleration, Velocity or Pressure. The default value shall be an empty string. The corresponding values are: A, V and P. If any other string is set, an error shall be generated.

2.12.3.4 Property Gain

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

Argument Name	Description
*pVal, newVal	The gain associated to a channel. The default value shall be 0. The valid value ranges from 1 to 16 when the mode is set to Acceleration or Pressure and from 1 to 96 when the mode is set to Velocity. If the value is set outside of this range, an error shall be generated.

2.12.3.5 Property IsConnected

```
// IsConnected  
[propget, id(5)]  
HRESULT IsConnected ([out, retval] BOOL *pVal);  
[propput, id(5)]  
HRESULT IsConnected ([in] BOOL newVal);
```

Argument Name	Description
*pVal, newVal	A flag to determine if the channel is connected. The default value shall be <i>TRUE</i> .

2.12.3.6 Property SensorName

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

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

2.12.3.7 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(2000)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.13 Interface "DDSControlSTINCAs"**2.13.1 Description**

2.13.1.1 This interface describes the STINCA Controls for the DDTC component of the DDS 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 *ControlSTINCAs* property of the interface *SubsystemDDS*.

2.13.2.4 This interface shall implement the collection interface for *LPDISPATCH*, returning the *DDSControlSTINCA* interfaces in the collection.

2.13.3 Methods and Properties**2.13.3.1 Property Find**

```
// Find a data record.
[propget, id(1)]
HRESULT Find([in] BSTR Name, [in] BSTR Selection,
             [out, retval] long *pVal);
```

Argument Name	Description
Name	The name of the STINCA Device to be found.
Selection	The Selection on the STINCA Device to be found.
*pVal	the index of the <i>DDSControlSTINCA</i> with the name <i>Name</i> and the selection <i>Selection</i> , if it is found in the collection, and 0 otherwise.

2.13.3.2 Property New

```
// Create a data record.  
[propget, id(1003)]  
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>DDSControlSTINCA</i> .

2.13.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.14 Interface "DDSControlSTINCA"**2.14.1 Description**

2.14.1.1 This interface represents a DDS Control STINCA device.

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 *DDSControlSTINCA*s.

2.14.3 Methods and Properties**2.14.3.1 Property Name**

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

Argument Name	Description
*pVal, newVal	a unique identifier for the DDS Control STINCA device. 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. 2.9), an error shall be generated.

2.14.3.2 Property Selection

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

Argument Name	Description
*pVal, newVal	the selection on the device. The valid values are A, B, C or D. The default value shall be an empty string.

2.14.3.3 Property Gain1

```
// Gain1
[propget, id(3)]
HRESULT Gain1 ([out, retval] long *pVal);
[propput, id(3)]
HRESULT Gain1 ([in] long newVal);;
```

Argument Name	Description
*pVal, newVal	The gain associated to the channels 1 to 12 on the STINCA device. The default value shall be 0. The valid value ranges from 1 to 5. If the value is set outside of this range, an error shall be generated.

2.14.3.4 Property Gain2

```
// Gain2
[propget, id(4)]
HRESULT Gain2 ([out, retval] long *pVal);
[propput, id(4)]
HRESULT Gain2 ([in] long newVal);;
```

Argument Name	Description
*pVal, newVal	The gain associated to the channels 13 to 24 on the STINCA device. The default value shall be 0. The valid value ranges from 1 to 5. If the value is set outside of this range, an error shall be generated.

2.14.3.5 Property IsChanged

```
// Flag whether the data have been changed
[propget, id(2000)]
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.15 Interface "SubsystemGASS"**2.15.1 Description**

2.15.1.1 This interface describes the VXI components of the VXI, GASSAI, GASSAO, GASSDIO, GASSTC and HSV subsystems.

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 The interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.15.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.15.3 Methods and Properties**2.15.3.1 Property Groups**

This property is obsolete. Instead, please refer to the new generic attribute CalGroup.

```
// GASS Groups
[propget, id(1)]
HRESULT Groups([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	the GASS Groups of similar channels for such tasks as configuration and calibration.

2.15.3.2 Property Cards

```
// GASS Cards
[propget, id(2)]
HRESULT Cards([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	the GASS Cards implemented.

2.15.3.3 Property BufferStorageRate

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

Argument Name	Description
*pVal, newVal	Specifies the buffer storage rate for the High Speed VXI (HSV) subsystem. The default value will be 0. Valid values are 250, 500, 769, 1000, 1250, 1666 and 2000 Hz. Any other value will result in an error being generated. An error will also be generated if an attempt is made to define this property for a non HSV subsystem.

2.16 Interface "GASSGroups"

This interface is obsolete. Instead, please refer to the new generic attribute CalGroup.

2.16.1 Description

2.16.1.1 This interface represents a collection of GASS Groups.

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 implement the collection interface for *LPDISPATCH*, returning the *GASSGroup* interfaces in the collection.

2.16.2.4 This interface shall not be directly created. It shall be created by the *Groups* property of the interface *SubsystemGASS*.

2.16.3 Methods and Properties**2.16.3.1 Property Find**

```
// Find a data record.
[propget, id(1)]
HRESULT Find([in] BSTR Name, [out, retval] long *pVal);
```

Argument Name	Description
Name	The name of the GASSGroup to be found.
*pVal	the index of the GASSGroup with the name <i>Name</i> , if it is found in the collection, and 0 otherwise.

2.16.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created GASSGroup.

2.16.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.17 Interface "GASSGroup"

This interface is obsolete. Instead, please refer to the new generic attribute CalGroup.

2.17.1 Description

2.17.1.1 This interface represents a group of a VXI GASS 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 *GASSGroups*.

2.17.3 Methods and Properties**2.17.3.1 Property Name**

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

Argument Name	Description
*pVal, newVal	the VXI group name. The default value shall be an empty string. If an empty string is set or the group name is that of another group within the list of groups, an error shall be generated.

2.17.3.2 Property Id

```
// Group ID
[propget, id(2)]
HRESULT Id ([out, retval] long *pVal);
```

Argument Name	Description
*pVal	a unique key representing the group.

2.17.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(2000)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.18 Interface "GASSCards"**2.18.1 Description**

2.18.1.1 This interface represents a collection of GASS cards. A single GASS card is defined in 2.19.

2.18.1.2 GASS allows a maximum of two Chassis'. Each chassis may contain a maximum of 12 cards i.e. the maximum card number is limited to 24.

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 implement the collection interface for *LPDISPATCH*, returning the *GASSCard* interfaces in the collection.

2.18.2.4 This interface shall not be directly created. It shall be created by the *Cards* property of the interface *SubsystemGASS*.

2.18.3 Methods and Properties**2.18.3.1 Property Find**

```
// Find a data record.
[propget, id(1)]
HRESULT Find([in] long Chassis, [in] long Slot,
             [out, retval] long *pVal);
```

Argument Name	Description
Chassis	The Chassis number to be found.
Slot	The Slot number to be found.
*pVal	the index of the <i>GASSCard</i> with the chassis with number <i>Chassis</i> and the slot with number <i>Slot</i> , if it is found in the collection, and 0 otherwise.

2.18.3.2 Property New

```
// Create a data record.  
[propget, id(1003)]  
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>GASSCard</i> .

2.18.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.19 Interface "GASSCard"**2.19.1 Description**

2.19.1.1 This interface represents a card of a VXI GASS subsystem.

2.19.1.2 The card type depends on the GASS type i.e. not all cards can be applied for each GASS type.

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 *GASSCards*.

2.19.3 Methods and Properties**2.19.3.1 Property Chassis**

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

Argument Name	Description
*pVal, newVal	the VXI chassis number. This shall be either 1, 2 or 3. The default value shall be 0. If 0 is set, an error shall be generated.

2.19.3.2 Property Slot

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

Argument Name	Description
*pVal, newVal	the VXI slot number. This shall be a value between 1 and 12. The default value shall be 0. If 0 is set, an error shall be generated.

2.19.3.3 Property Type

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

Argument Name	Description
*pVal, newVal	the type of card supported by the RTE. These are the Agilent 1413, 1415, 1418, 1419 and 1458 cards. The default value shall be an empty string. If an empty string is set, an error shall be generated.

2.19.3.4 Property SCPs

```
// GASS SCPs
[propget, id(7)]
HRESULT SCPs([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	gives access to the GASSSCP interface.

2.19.3.5 Property Pinouts

```
// GASS Pinouts
[propget, id(8)]
HRESULT Pinouts([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	gives access to the GASSPinout interface.

2.19.3.6 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(2000)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.20 Interface "GASSSCPs"**2.20.1 Description**

2.20.1.1 This interface represents a collection of VXI GASS SCPs. A single SCP is defined in 2.21.

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 implement the collection interface for *LPDISPATCH*, returning the *GASSSCP* interfaces in the collection.

2.20.2.4 This interface shall not be directly created. It shall be created by the *SCPs* property of the interface *GASSCard*.

2.20.3 Methods and Properties**2.20.3.1 Property Find**

```
// Find a data record.
[propget, id(6)]
HRESULT Find([in] long Location, [out, retval] long *pVal);
```

Argument Name	Description
Location	The location of the GASSSCP to be found.
*pVal	the index of the GASSSCP with the location <i>Location</i> , if it is found in the collection, and 0 otherwise.

2.20.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created GASSSCP.

2.20.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.21 Interface "GASSSCP"**2.21.1 Description**

2.21.1.1 This interface represents a VXI GASS SCP, located on a VXI GASS Card.

2.21.1.2 The SCP depends on the GASS type and the card type.

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 *GASSSCPs*.

2.21.3 Methods and Properties**2.21.3.1 Property Location**

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

Argument Name	Description
*pVal, newVal	the location of the SCP on the card. This shall have a value of 0 - 7. The default value shall be -1. If -1 is set, an error shall be generated.

2.21.3.2 Property Type

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

Argument Name	Description
*pVal, newVal	the type of SCP's supported by the RTE. These are the Agilent 1501, 1502, 1503, 1505, 1508, 1509, 1510, 1512, 1534, 1536 and 1538 SCP's. The default value shall be an empty string. If an empty string is set, an error shall be generated.

2.21.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(2000)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.22 Interface "GASSPinouts"**2.22.1 Description**

2.22.1.1 This interface represents a collection of VXI GASS Pinouts. A single VXI GASS Pinout is defined in 2.23.

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 implement the collection interface for *LPDISPATCH*, returning the *GASSPinout* interfaces in the collection.

2.22.2.4 This interface shall not be directly created. It shall be created by the *Pinouts* property of the interface *GASSCard*.

2.22.3 Methods and Properties**2.22.3.1 Property Find**

```
// Find a data record.
[propget, id(6)]
HRESULT Find([in] BSTR Name, [out, retval] long *pVal);
```

Argument Name	Description
Name	The name of the GASSPinout to be found.
*pVal	the index of the GASSPinout with the name <i>Name</i> , if it is found in the collection, and 0 otherwise.

2.22.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created GASSPinout.

2.22.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.23 Interface "GASSPinout"**2.23.1 Description**

2.23.1.1 This interface represents a VXI GASS Pinout, located on a VXI GASS Card.

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 *GASSPinouts*.

2.23.3 Methods and Properties**2.23.3.1 Property Name**

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

Argument Name	Description
*pVal, newVal	the pinout name associated with the hardware channel. The default value shall be an empty string. If an empty string is set, an error shall be generated.

2.23.3.2 Property ChannelNumber

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

Argument Name	Description
*pVal, newVal	the channel number of the card. The valid range shall be 0 – 95 (0 - 63 for most GASS systems, 0 – 95 for GASSDIO and 0 – 15 for GASSAO). The default value shall be -1. If -1 is set, an error shall be generated.

2.23.3.3 Property GroupName

This attribute is obsolete.

```
// Group Name
[propget, id(9)]
HRESULT GroupName ([out, retval] BSTR *pVal);
[propput, id(9)]
HRESULT GroupName ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the name for the specific Group Name. The default value shall be an empty string. If an empty string is set, an error shall be generated.

2.23.3.4 Property CalMax

This attribute is obsolete.

```
// Max. target for the calibration range
[propget, id(6)]
HRESULT CalMax ([out, retval] float *pVal);
[propput, id(6)]
HRESULT CalMax ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	the maximum target for the calibration range of the channel. This value defines the target in volts.

2.23.3.5 Property CalMin

This attribute is obsolete.

```
// Min. target for the calibration range
[propget, id(7)]
HRESULT CalMin ([out, retval] float *pVal);
[propput, id(7)]
HRESULT CalMin ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	the minimum target for the calibration range of the channel. This value defines the target in volts.

2.23.3.6 Property IsCalEndToEnd

This attribute is obsolete.

```
// Flag for End to End Calibration
[propget, id(8)]
HRESULT IsCalEndToEnd ([out, retval] BOOL *pVal);
[propget, id(8)]
HRESULT IsCalEndToEnd ([in] BOOL newVal);
```

Argument Name	Description
*pVal, newVal	whether an end-to-end calibration is necessary. The default value shall be <i>false</i> .

2.23.3.7 Property IsChanged

```
// Flag whether the data have been changed
[propget, id(2000)]
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.23.3.8 Property Description

```
// Description of pinout
[propget, id(10)]
HRESULT Description([out, retval] BSTR *pVal);
[propget, id(10)]
HRESULT Description([in] BSTR newVal );
```

Argument Name	Description
*pVal, newVal	Specifies description of pinout. Default value is "".

2.24 Interface "SubsystemHSS"**2.24.1 Description**

2.24.1.1 This interface will represent the HSS subsystem and define the file name for this 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 *Subsystems*.

2.24.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.24.3 Methods and Properties**2.24.3.1 Property FileName**

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

Argument Name	Description
*pVal, newVal	the name of the HSS configuration file from which the channel configuration for the subsystem can be generated. The property shall be set to the default value i.e. an empty string.

2.25 Interface "SubsystemPBS"**2.25.1 Description**

2.25.1.1 This interface describes the Pressure Bricks 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 The interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.25.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.25.3 Methods and Properties**2.25.3.1 Property Bricks**

```
// Access to the collection of bricks  
[propget, id(1)]  
HRESULT Bricks ([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	the collection of bricks of the PBS subsystem.

2.25.3.2 Method GetScannerTypes

```
[id(2)]  
HRESULT GetScannerTypes ([out] BSTR* pScannerTypes)
```

Argument Name	Description
* pScannerTypes	<p>A tab delimited string with all available PBS brick scanner types.</p> <p>Valid scanner types are:</p> <ul style="list-style-type: none">• 9016• 9116• 9021• 9022• 9032• 9816

2.25.3.3 Method GetMaxPortNumber

```
[id(3)]  
HRESULT GetMaxPortNumber ([in] BSTR type, [out,retval] long*  
pVal);
```

Argument Name	Description
type	The scanner type.
pVal	Returns the maximum value for the port number dependent on the scanner type.

2.26 Interface "PBSBricks"**2.26.1 Description**

2.26.1.1 This interface describes the collection of the Pressure Bricks.

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 *Bricks* property of the interface *SubsystemPBS*.

2.26.2.4 This interface shall implement the collection interface for *LPDISPATCH*, returning the *PBSBrick* interfaces in the collection.

2.26.3 Methods and Properties**2.26.3.1 Property Find**

```
// Find a data record.
[propget, id(1)]
HRESULT Find([in] BSTR SerialNumber, [out, retval] long *pVal);
```

Argument Name	Description
SerialNumber	The Serial Number of the <i>PBSBrick</i> to be found.
*pVal	the index of the <i>PBSBrick</i> with the serial number <i>SerialNumber</i> , if it is found in the collection, and 0 otherwise.

2.26.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>PBSBrick</i> .

2.26.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.27 Interface "PBSBrick"**2.27.1 Description**

2.27.1.1 This interface represents a PSI pressure brick.

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 *PBSBricks*.

2.27.3 Methods and Properties**2.27.3.1 Property SerialNumber**

```
// Serial Number  
[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 a pressure brick on the basis of the serial number . The default value shall be an empty string. If an empty string is set or the serial number is that of another serial number within the list of bricks, an error shall be generated.

2.27.3.2 Property MaxScanRate

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

Argument Name	Description
*pVal, newVal	the maximum scan rate allowed on this brick in Hz. The default value shall be 0. If 0 is set, an error shall be generated. The valid maximum scan rates are defined by the Configuration Server option “ScanRates”. If the option is not defined, the default scan rates 0, 1, 10, 33, 50, 100 and 200 are valid values.

2.27.3.3 Property Ports

```
// PBS Ports
[propget, id(3)]
HRESULT Ports([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	A PBSPorts interface.

2.27.3.4 Property IsChanged

```
// Flag whether the data have been changed
[propget, id(2000)]
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.27.3.5 Property ContPurgeEnable

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

Argument Name	Description
*pVal, newVal	<p>Flag indicating whether Continuous Purge is enabled for the brick. If it is set to true, it will automatically enable purging for the brick.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • 0 = Disabled (default) • 1 = Enabled

2.27.3.6 Property ScannerType

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

Argument Name	Description
*pVal, newVal	<p>Specifies the type of the used PBS scanner.</p> <p>Note: The method <i>GetScannerTypes</i> of the <i>SubsystemPBS</i> interface can be used to receive a list of all available PBS scanners.</p>

2.27.3.7 Property DefaultEu

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

Argument Name	Description
*pVal, newVal	Specifies the default engineering unit for a new PBS channel.

2.28 Interface "PBSPorts"**2.28.1 Description**

2.28.1.1 This interface represents a collection of ports of one PSI pressure brick. A single port is defined in 2.29.

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 implement the collection interface for *LPDISPATCH*, returning the *PBSPort* interfaces in the collection.

2.28.2.4 This interface shall not be directly created. It shall be created by the *Ports* property of the interface *PBSBrick*.

2.28.3 Methods and Properties**2.28.3.1 Property Find**

```
// Find a data record.
[propget, id(1)]
HRESULT Find([in] long Port, [out, retval] long *pVal);
```

Argument Name	Description
Port	The port of the <i>PBSPort</i> to be found.
*pVal	the index of the <i>PBSPort</i> with the port <i>Port</i> , if it is found in the collection, and 0 otherwise.

2.28.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>PBSPort</i> .

2.28.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.29 Interface "PBSPort"**2.29.1 Description**

2.29.1.1 This interface represents a port of a PSI pressure brick.

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 *PBSPorts*.

2.29.3 Methods and Properties**2.29.3.1 Property PortNumber**

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

Argument Name	Description
*pVal, newVal	the pressure port on the brick. The valid range shall be 1 – 16 for a 9016, 9116 or 9816 scanner type, 1 – 12 for a 9021 or 9022 scanner type, and 1 for a 9032 scanner type. The default value shall be 0. If 0 is set, an error shall be generated.

2.29.3.2 Property MaxPressRange

```
// Maximum Pressure Range
[propget, id(2)]
HRESULT MaxPressRange ([out, retval] float *pVal);
[propput, id(2)]
HRESULT MaxPressRange ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	the maximum pressure range allowed on the port. The default shall be 0.0. If 0.0 is set an error shall be generated.

2.29.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(2000)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.29.3.4 Property Description

```
// Description of the port  
[propget, id(3)]  
HRESULT Description([out, retval] BSTR *pVal);  
[propput, id(3)]  
HRESULT Description([in] BSTR newVal );
```

Argument Name	Description
*pVal, newVal	Specifies description of port. Default value = ""

2.30 Interface "SubsystemRTP"**2.30.1 Description**

2.30.1.1 This interface will represent an RTP 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 *Subsystems*.

2.30.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.30.3 Methods and Properties**2.30.3.1 Property HostNames**

```
// Access to the list of host names  
[propget, id(1)]  
HRESULT HostNames([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	the collection of host names using the interface Strings.

2.31 Interface "SubsystemSCUTR"**2.31.1 Description**

2.31.1.1 This interface describes the Scanning UTR 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 The interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.31.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.31.3 Methods and Properties**2.31.3.1 Property Boxes**

```
// Access to the collection of boxes  
[propget, id(1)]  
HRESULT Boxes ([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	the collection of boxes of the SCUTR subsystem.

2.32 Interface "SCUTRBoxes"**2.32.1 Description**

2.32.1.1 This interface describes the collection of boxes of a SCUTR 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 *Boxes* property of the interface *SubsystemSCUTR*.

2.32.2.4 This interface shall implement the collection interface for *LPDISPATCH*, returning the *SCUTRBox* interfaces in the collection.

2.32.3 Methods and Properties**2.32.3.1 Property Find**

```
// Find a data record.
[propget, id(1)]
HRESULT Find([in] BSTR SerialNumber, [out, retval] long *pVal);
```

Argument Name	Description
SerialNumber	The serial number of the <i>SCUTRBox</i> to be found.
*pVal	the index of the <i>SCUTRBox</i> with the serial number <i>SerialNumber</i> , if it is found in the collection, and 0 otherwise.

2.32.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>SCUTRBox</i> .

2.32.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.33 Interface "SCUTRBox"**2.33.1 Description**

2.33.1.1 This interface represents a SCUTR box.

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 *SCUTRBoxes*.

2.33.3 Methods and Properties**2.33.3.1 Property SerialNumber**

```
// Serial Number
[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 a SCUTR box. The default value shall be an empty string. If an empty string is set or the serial number is that of another serial number within the list of SCUTRBoxes, an error shall be generated.

2.33.3.1.1 Please note that matching Sensor Names must be present. The Sensor Names shall be *SerialNumber.RTD1*, *SerialNumber.RTD2* and *SerialNumber.VTCC*.

2.33.3.2 Property IsTransient

```
// transient flag
[propget, id(2)]
HRESULT IsTransient([out, retval] BOOL *pVal);
[propput, id(2)]
HRESULT IsTransient([in] BOOL newVal);
```

Argument Name	Description
*pVal, newVal	whether the SCUTR box supports transient thermocouples. The default value shall be <i>false</i> , i.e. no transient thermocouples are supported.

2.33.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(2000)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.34 Interface "SubsystemTruTemp"**2.34.1 Description**

2.34.1.1 This interface will represent a single Kaye TruTemp Temperature Measurement Subsystem, consisting of gateways, TruTemp units and scanners.

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 *Subsystems*.

2.34.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.34.3 Methods and Properties**2.34.3.1 Property Gateways**

```
// access to TruTemp gateways
[propget, id(1)]
HRESULT Gateways([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	the collection of the gateways of the TruTemp subsystem.

2.34.3.2 Property Units

```
// access to units
[propget, id(2)]
HRESULT Units([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	the collection of the units of the TruTemp subsystem.

2.35 Interface "TruTempGateways"**2.35.1 Description**

2.35.1.1 This interface will represent the collection of gateways of a TruTemp 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 implement the collection interface for *LPDISPATCH*, returning the *TruTempGateway* interfaces in the collection.

2.35.2.4 This interface shall not be directly created. It shall be created by the *Gateways* property of the interface *SubsystemTruTemp*.

2.35.3 Methods and Properties**2.35.3.1 Property Find**

```
// Find a data record.
[propget, id(1)]
HRESULT Find([in] BSTR Name, [out, retval] long *pVal);
```

Argument Name	Description
Name	The name of the <i>TruTempGateway</i> to be found.
*pVal	the index of the <i>TruTempGateway</i> with the name <i>Name</i> , if it is found in the collection, and 0 otherwise.

2.35.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>TruTempGateway</i> .

2.35.3.2.1 The maximum number of gateways per TruTemp subsystem is 5. If an attempt to create more than 5 Gateways occurs, an error shall be generated.

2.35.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.36 Interface "TruTempGateway"**2.36.1 Description**

2.36.1.1 This interface will represent a single gateway of a TruTemp 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 *TruTempGateways*.

2.36.3 Methods and Properties**2.36.3.1 Property Name**

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

Argument Name	Description
*pVal, newVal	the name of the gateway. It shall uniquely identify the gateway within the TruTemp subsystem. The default value shall be an empty string. If an empty string is set or it is set to the name of another gateway of the same TruTemp subsystem an error shall be generated.

2.36.3.2 Property ScanRate

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

Argument Name	Description
*pVal, newVal	the scan rate of the gateway. The default value shall be 0 Hz. The valid values shall be 1 and 10 Hz. If 0 is set, an error shall be generated.

2.36.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(2000)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.37 Interface "TruTempUnits"**2.37.1 Description**

2.37.1.1 This interface will represent the collection of units of a TruTemp 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 implement the collection interface for *LPDISPATCH*, returning the *TruTempUnit* interfaces in the collection.

2.37.2.4 This interface shall not be directly created. It shall be created by the *Units* property of the interface *SubsystemTruTemp*.

2.37.3 Methods and Properties**2.37.3.1 Property Find**

```
// Find a data record.  
[propget, id(1)]  
HRESULT Find([in] long Id, [out, retval] long *pVal);
```

Argument Name	Description
Id	The <i>Id</i> of the <i>TruTempUnit</i> to be found.
*pVal	the index of the <i>TruTempUnit</i> with the identification <i>Id</i> , if it is found in the collection, and 0 otherwise.

2.37.3.2 Property New

```
// Create a data record.  
[propget, id(1003)]  
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>TruTempUnit</i> .

2.37.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.38 Interface "TruTempUnit"**2.38.1 Description**

2.38.1.1 This interface will represent a single unit of a TruTemp subsystem.

2.38.2 Design

2.38.2.1 This interface shall be a dispatch interface.

2.38.2.2 This interface shall be an automation interface.

2.38.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *TruTempUnits*.

2.38.3 Methods and Properties**2.38.3.1 Property Id**

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

Argument Name	Description
*pVal, newVal	the identification number of the unit. It shall be unique within the same TruTemp subsystem. The default value for the unit shall be 0. If 0 or the number of another unit is set, an error shall be generated.

2.38.3.2 Property Scanners

```
// access to scanners
[propget, id(2)]
HRESULT Scanners ([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	the collection of the scanners of the unit.

2.38.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(2000)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.39 Interface "TruTempScanners"**2.39.1 Description**

2.39.1.1 This interface will represent the collection of scanners of a TruTemp unit.

2.39.2 Design

2.39.2.1 This interface shall be a dispatch interface.

2.39.2.2 This interface shall be an automation interface.

2.39.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *TruTempScanner* interfaces in the collection.

2.39.2.4 This interface shall not be directly created. It shall be created by the *Scanners* property of the interface *TruTempUnit*.

2.39.3 Methods and Properties**2.39.3.1 Property Find**

```
// Find a data record.
[propget, id(1)]
HRESULT Find([in] long Address, [out, retval] long *pVal);
```

Argument Name	Description
Address	The <i>Address</i> of the <i>TruTempScanner</i> to be found.
*pVal	the index of the <i>TruTempScanner</i> with the address <i>Address</i> , if it is found in the collection, and 0 otherwise.

2.39.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>TruTempScanner</i> .

2.39.3.2.1 The maximum number of scanners per TruTemp unit is 5.

2.39.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.40 Interface "TruTempScanner"**2.40.1 Description**

2.40.1.1 This interface will represent a single scanner of a TruTemp unit.

2.40.2 Design

2.40.2.1 This interface shall be a dispatch interface.

2.40.2.2 This interface shall be an automation interface.

2.40.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *TruTempScanners*.

2.40.3 Methods and Properties**2.40.3.1 Property Address**

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

Argument Name	Description
*pVal, newVal	the address of the scanner. The address shall be in the range from 1 to 239 and shall be unique within the subsystem. The default value shall be 0. If it is set to 0 or the address of another scanner, an error shall be generated.

2.40.3.2 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(2000)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.41 Interface "SubsystemHyScan"**2.41.1 Description**

2.41.1.1 This interface will represent a single HyScan subsystem containing a number of HyScan PC's.

2.41.2 Design

2.41.2.1 This interface shall be a dispatch interface.

2.41.2.2 This interface shall be an automation interface.

2.41.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.41.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.41.3 Methods and Properties**2.41.3.1 Property NumberOfComputers**

```
// Define number of PC's  
[propget, id(1)]  
HRESULT NumberOfComputers([out, retval] long *pVal);  
[propput, id(1)]  
HRESULT NumberOfComputers([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the number of PC's configured for the HyScan subsystem. The property shall be set to the default value i.e. 0. If 0 is set an error will be generated.

2.42 Interface "SubsystemModbus"**2.42.1 Description**

2.42.1.1 This interface will represent the MODBUS subsystem and define the maximum addresses available for float and Boolean input and output channels. These values are valid for both ETHERNET and SERIAL subsystem types.

2.42.2 Design

2.42.2.1 This interface shall be a dispatch interface.

2.42.2.2 This interface shall be an automation interface.

2.42.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.42.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.42.3 Methods and Properties**2.42.3.1 Property MaxFloatInAddress**

```
// Max. address of float input channels
[propget, id(1)]
HRESULT MaxFloatInAddress([out, retval] long *pVal);
[propput, id(1)]
HRESULT MaxFloatInAddress([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the maximum address available for float data travelling from the MODBUS system to the RTE. The property shall be set to the default value i.e. 0. If 0 is set an error will be generated.

2.42.3.2 Property MaxBooleanInAddress

```
// Max. address of Boolean input channels  
[propget, id(2)]  
HRESULT MaxBooleanInAddress([out, retval] long *pVal);  
[propput, id(2)]  
HRESULT MaxBooleanInAddress([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the maximum address available for Boolean data travelling from the MODBUS system to the RTE. The property shall be set to the default value i.e. 0. If 0 is set an error will be generated.

2.42.3.3 Property MaxFloatOutAddress

```
// Max. address of float output channels  
[propget, id(3)]  
HRESULT MaxFloatOutAddress([out, retval] long *pVal);  
[propput, id(3)]  
HRESULT MaxFloatOutAddress([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the maximum address available for float data travelling from the RTE to the MODBUS system. The property shall be set to the default value i.e. 0. If 0 is set an error will be generated.

2.42.3.4 Property MaxBooleanOutAddress

```
// Max. address of Boolean output channels  
[propget, id(4)]  
HRESULT MaxBooleanOutAddress([out, retval] long *pVal);  
[propput, id(4)]  
HRESULT MaxBooleanOutAddress([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the maximum address available for Boolean data travelling from the RTE to the MODBUS system. The property shall be set to the default value i.e. 0. If 0 is set an error will be generated.

2.43 Interface "SubsystemPLC"**2.43.1 Description**

2.43.1.1 This interface will represent the PLC subsystem and define the maximum addresses available for float and Boolean input and output channels. These values are valid for Allan Bradley, GEFANUC and Omron subsystem types.

2.43.2 Design

2.43.2.1 This interface shall be a dispatch interface.

2.43.2.2 This interface shall be an automation interface.

2.43.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.43.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.43.3 Methods and Properties**2.43.3.1 Property MaxFloatInAddress**

```
// Max. address of float input channels
[propget, id(1)]
HRESULT MaxFloatInAddress([out, retval] long *pVal);
[propput, id(1)]
HRESULT MaxFloatInAddress([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the maximum address available for float data travelling from the PLC system to the RTE. The property shall be set to the default value i.e. 0. If 0 is set an error shall be generated. This property is ignored for the Omron PLC.

2.43.3.2 Property MaxBooleanInAddress

```
// Max. address of Boolean input channels
[propget, id(2)]
HRESULT MaxBooleanInAddress([out, retval] long *pVal);
[propput, id(2)]
HRESULT MaxBooleanInAddress([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the maximum address available for Boolean data travelling from the PLC system to the RTE. The property shall be set to the default value i.e. 0. If 0 is set an error shall be generated.

2.43.3.3 Property MaxFloatOutAddress

```
// Max. address of float output channels
[propget, id(3)]
HRESULT MaxFloatOutAddress([out, retval] long *pVal);
[propput, id(3)]
HRESULT MaxFloatOutAddress([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the maximum address available for float data travelling from the RTE to the PLC system. The property shall be set to the default value i.e. 0. If 0 is set an error shall be generated. This property is ignored for the Omron PLC.

2.43.3.4 Property MaxBooleanOutAddress

```
// Max. address of Boolean output channels  
[propget, id(4)]  
HRESULT MaxBooleanOutAddress([out, retval] long *pVal);  
[propput, id(4)]  
HRESULT MaxBooleanOutAddress([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the maximum address available for Boolean data travelling from the RTE to the PLC system. The property shall be set to the default value i.e. 0. If 0 is set an error shall be generated.

2.44 Interface "SubsystemTBDAU"**2.44.1 Description**

2.44.1.1 This interface will represent a single TBDAU subsystem containing the maximum number of channels for the TBDAU.

2.44.2 Design

2.44.2.1 This interface shall be a dispatch interface.

2.44.2.2 This interface shall be an automation interface.

2.44.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.44.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.44.3 Methods and Properties**2.44.3.1 Property MaxChannels**

```
// Define maximum number of channels
[propget, id(1)]
HRESULT MaxChannels([out, retval] long *pVal);
[propput, id(1)]
HRESULT MaxChannels([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the maximum number of channels (203 at the moment) configured for the TBDAU subsystem. The property shall be set to the default value i.e. 0. If 0 is set an error will be generated.

2.45 Interface "SubsystemThrust"**2.45.1 Description**

2.45.1.1 This interface will represent the Engine Specific and Nobel Thrust subsystems.

2.45.2 Design

2.45.2.1 This interface shall be a dispatch interface.

2.45.2.2 This interface shall be an automation interface.

2.45.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.45.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.45.3 Methods and Properties

2.45.3.1 Property NumberOfWorkingCells

```
// Define number of Working cells
[propget, id(1)]
HRESULT NumberOfWorkingCells([out, retval] long *pVal);
[propput, id(1)]
HRESULT NumberOfWorkingCells([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the number of working load cells configured for the Thrust subsystem. The property shall be set to the default value i.e. 0. If 0 is set an error will be generated.

2.45.3.2 Property NumberOfMasterCells

```
// Define number of Master cells  
[propget, id(2)]  
HRESULT NumberOfMasterCells([out, retval] long *pVal);  
[propput, id(2)]  
HRESULT NumberOfMasterCells([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the number of master load cells configured for the Thrust subsystem. The property shall be set to the default value i.e. 0. If 0 is set an error will be generated.

2.46 Interface “Subsystem External”**2.46.1 Description**

2.46.1.1 This interface will represent the External subsystem and define the executable for this subsystem.

2.46.2 Design

2.46.2.1 This interface shall be a dispatch interface.

2.46.2.2 This interface shall be an automation interface.

2.46.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.46.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.46.3 Methods and Properties**2.46.3.1 Property Executable**

```
// Define the name of the executable
[propget, id(1)]
HRESULT Executable([out, retval] BSTR *pVal);
[propput, id(1)]
HRESULT Executable([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the complete path name of the executable for the subsystem of type <i>External</i> . The property shall be set to the default value i.e. an empty string. If an empty string is set an error will be generated.

2.46.3.2 Property Recalculate

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

Argument Name	Description
*pVal, newVal	Flag whether the external hook program is to be used for recalculation. The default value shall be <i>false</i> .

2.47 Interface “SubsystemMSS”**2.47.1 Description**

2.47.1.1 This interface describes the Mechanical Scanivalve subsystem.

2.47.1.2 Only one instance of the Mechanical Scanivalve subsystem can operate within one test configuration.

2.47.2 Design

2.47.2.1 This interface shall be a dispatch interface.

2.47.2.2 This interface shall be an automation interface.

2.47.2.3 The interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.47.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.47.3 Methods and Properties**2.47.3.1 Property HomeCommand**

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

Argument Name	Description
*pVal, newVal	Specifies the channel that delivers the Home command. The default value is an empty string. If no valid channel is specified, an error will be generated.

2.47.3.2 Property StepCommand

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

Argument Name	Description
*pVal, newVal	Specifies the channel that delivers the Step command. The default value is an empty string. If no valid channel is specified, an error will be generated.

2.47.3.3 Property BCDPosition1

```
[propget, id(3)]  
HRESULT BCDPosition1([out, retval] BSTR *pVal);  
[propput, id(3)]  
HRESULT BCDPosition1([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the channel for the BCD address position 1. The default value is an empty string. If no valid channel is specified, an error will be generated.

2.47.3.4 Property BCDPosition2

```
[propget, id(4)]  
HRESULT BCDPosition2([out, retval] BSTR *pVal);  
[propput, id(4)]  
HRESULT BCDPosition2([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the channel for the BCD address position 2. The default value is an empty string. If no valid channel is specified, an error will be generated.

2.47.3.5 Property BCDPosition4

```
[propget, id(5)]  
HRESULT BCDPosition4([out, retval] BSTR *pVal);  
[propput, id(5)]  
HRESULT BCDPosition4([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the channel for the BCD address position 4. The default value is an empty string. If no valid channel is specified, an error will be generated.

2.47.3.6 Property BCDPosition8

```
[propget, id(6)]  
HRESULT BCDPosition8([out, retval] BSTR *pVal);  
[propput, id(6)]  
HRESULT BCDPosition8([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the channel for the BCD address position 8. The default value is an empty string. If no valid channel is specified, an error will be generated.

2.47.3.7 Property BCDPosition10

```
[propget, id(7)]  
HRESULT BCDPosition10([out, retval] BSTR *pVal);  
[propput, id(7)]  
HRESULT BCDPosition10([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the channel for the BCD address position 10. The default value is an empty string. If no valid channel is specified, an error will be generated.

2.47.3.8 Property BCDPosition20

```
[propget, id(8)]  
HRESULT BCDPosition20([out, retval] BSTR *pVal);  
[propput, id(8)]  
HRESULT BCDPosition20([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the channel for the BCD address position 20. The default value is an empty string. If no valid channel is specified, an error will be generated.

2.47.3.9 Property BCDPosition40

```
[propget, id(9)]  
HRESULT BCDPosition40([out, retval] BSTR *pVal);  
[propput, id(9)]  
HRESULT BCDPosition40([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the channel for the BCD address position 40. The default value is an empty string. If no valid channel is specified, an error will be generated.

2.47.3.10 Property ReadyToReadSignal

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

Argument Name	Description
*pVal, newVal	Specifies the channel that delivers the Ready-To-Read signal. The default value is an empty string.

2.47.3.11 Property Modules

```
[propget, id(11)]  
HRESULT Modules([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	The collection of modules of the MSS subsystem.

2.47.3.12 Property PressureUnit

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

Argument Name	Description
*pVal, newVal	Specifies the EU of the pressure values returned by the MSS modules.

2.48 Interface “MSSModules”**2.48.1 Description**

2.48.1.1 This interface describes the collection of the Mechanical Scanivalve Modules.

2.48.2 Design

2.48.2.1 This interface shall be a dispatch interface.

2.48.2.2 This interface shall be an automation interface.

2.48.2.3 This interface shall not be directly created. It shall be created by the *Modules* property of the interface *SubsystemMSS*.

2.48.2.4 This interface shall implement the collection interface for *LPDISPATCH*, returning the *MSSModule* interfaces in the collection.

2.48.3 Methods and Properties**2.48.3.1 Property Find**

```
// Find a data record.
[propget, id(1)]
HRESULT Find([in] long ModuleNumber, [out, retval] long *pVal);
```

Argument Name	Description
ModuleNumber	The Module Number of the <i>MSSModule</i> to be found.
*pVal	The index of the <i>MSSModule</i> with the module number <i>ModuleNumber</i> , if it is found in the collection, and 0 otherwise.

2.48.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>MSSModule</i> .

2.48.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.49 Interface “MSSModule”**2.49.1 Description**

2.49.1.1 This interface represents a Mechanical Scanivalve Module.

2.49.2 Design

2.49.2.1 This interface shall be a dispatch interface.

2.49.2.2 This interface shall be an automation interface.

2.49.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *MSSModules*.

2.49.3 Methods and Properties**2.49.3.1 Property ModuleNumber**

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

Argument Name	Description
*pVal, newVal	Specifies the module number. The value must be within the range 1 – 12. The default is set to 0. If a value outside of the range is specified, an error will be generated.

2.49.3.2 Property PressureMeasurementChannel

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

Argument Name	Description
*pVal, newVal	Specifies the channel in the GASS subsystem that measures the ports of this module. The default value is an empty string. If an invalid channel is specified, an error will be generated.

2.49.3.3 Property MaximumPressure

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

Argument Name	Description
*pVal, newVal	Specifies the maximum pressure that can be measured using the ports connected to the pressure module. The default is set to 1.0.

2.49.3.4 Property Ports

```
// MSS Ports  
[propget, id(4)]  
HRESULT Ports([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	A MSSPorts interface.

2.50 Interface "MSSPorts"**2.50.1 Description**

2.50.1.1 This interface represents a collection of ports of one Mechanical Scanivalve Module. A single port is defined in 2.51.

2.50.2 Design

2.50.2.1 This interface shall be a dispatch interface.

2.50.2.2 This interface shall be an automation interface.

2.50.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *MSSPort* interfaces in the collection.

2.50.2.4 This interface shall not be directly created. It shall be created by the *Ports* property of the interface *MSSModule*.

2.50.3 Methods and Properties**2.50.3.1 Property Find**

```
// Find a data record.
[propget, id(1)]
HRESULT Find([in] long Port, [out, retval] long *pVal);
```

Argument Name	Description
Port	The port of the <i>MSSPort</i> to be found.
*pVal	the index of the <i>MSSPort</i> with the port <i>Port</i> , if it is found in the collection, and 0 otherwise.

2.50.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>MSSPort</i> .

2.50.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.51 Interface "MSSPort"**2.51.1 Description**

2.51.1.1 This interface represents a port of a Mechanical Scanivalve Module..

2.51.2 Design

2.51.2.1 This interface shall be a dispatch interface.

2.51.2.2 This interface shall be an automation interface.

2.51.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *MSSPorts*.

2.51.3 Methods and Properties**2.51.3.1 Property PortNumber**

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

Argument Name	Description
*pVal, newVal	the pressure port on the module. The valid range shall be 1 – 48. The default value shall be 0. If 0 is set, an error shall be generated.

2.51.3.2 Property Description

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

Argument Name	Description
*pVal, newVal	Specifies description of port. Default value = ""

2.51.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(2000)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.52 Interface “SubsystemUTRH”**2.52.1 Description**

2.52.1.1 This interface describes the UTR Half Box subsystem.

2.52.2 Design

2.52.2.1 This interface shall be a dispatch interface.

2.52.2.2 This interface shall be an automation interface.

2.52.2.3 The interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.52.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.52.3 Methods and Properties**2.52.3.1 Property HalfBoxes**

```
[propget, id(1)]  
HRESULT HalfBoxes([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	The collection of Half Boxes of the UTRH subsystem.

2.53 Interface “UTRHalfBoxes”**2.53.1 Description**

2.53.1.1 This interface describes the collection of the UTR Half Boxes.

2.53.2 Design

2.53.2.1 This interface shall be a dispatch interface.

2.53.2.2 This interface shall be an automation interface.

2.53.2.3 This interface shall not be directly created. It shall be created by the *HalfBoxes* property of the interface *SubsystemUTRH*.

2.53.2.4 This interface shall implement the collection interface for *LPDISPATCH*, returning the *UTRHalfBox* interfaces in the collection.

2.53.3 Methods and Properties**2.53.3.1 Property Find**

```
// Find a data record.  
[propget, id(1)]  
HRESULT Find([in] BSTR HalfBoxName, [out, retval] long *pVal);
```

Argument Name	Description
HalfBoxName	The Half Box Name of the <i>UTRHalfBox</i> to be found.
*pVal	the index of the <i>UTRHalfBox</i> with the name <i>HalfBoxName</i> , if it is found in the collection, and 0 otherwise.

2.53.3.2 Property New

```
// Create a data record.  
[propget, id(1003)]  
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>UTRHalfBox</i> .

2.53.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.54 Interface “UTRHalfBox”**2.54.1 Description**

2.54.1.1 This interface represents a UTR Half Box.

2.54.2 Design

2.54.2.1 This interface shall be a dispatch interface.

2.54.2.2 This interface shall be an automation interface.

2.54.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *UTRHalfBoxes*.

2.54.3 Methods and Properties**2.54.3.1 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 Half Box. The default value is an empty string. If a box name is specified twice, an error will be generated.

2.54.3.2 Property IdentificationResistanceValue

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

Argument Name	Description
*pVal, newVal	Specifies the value of the identification resistance. The default value is -1.0 Ohms. If the value is less than 1.0 an error will be generated.

2.54.3.3 Property IdentificationResistanceTolerance

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

Argument Name	Description
*pVal, newVal	Specifies the tolerance value for the identification resistance. The default value is 5 Ohms. If the value is less than 0 or greater than 1.0e10, an error will be generated.

2.54.3.4 Property ConversionCoefficientA

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

Argument Name	Description
*pVal, newVal	Specifies the coefficient A used to convert the bridge voltage to an RTD resistance. The default value is -1.0. If the value is less than 0 or greater than 1000, an error will be generated.

2.54.3.5 Property ConversionCoefficientB

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

Argument Name	Description
*pVal, newVal	Specifies the coefficient B used to convert the bridge voltage to an RTD resistance. The default value is -1.0. If the value is less than 0 or greater than 1000, an error will be generated.

2.54.3.6 Property ConversionCoefficientC

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

Argument Name	Description
*pVal, newVal	Specifies the coefficient C used to convert the bridge voltage to an RTD resistance. The default value is -1.0. If the value is less than 0 or greater than 1000, an error will be generated.

2.55 Interface “SubsystemDPS”**2.55.1 Description**

2.55.1.1 This interface describes the DSA Pressure Scanner subsystem.

2.55.2 Design

2.55.2.1 This interface shall be a dispatch interface.

2.55.2.2 This interface shall be an automation interface.

2.55.2.3 The interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.55.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.55.3 Methods and Properties**2.55.3.1 Property Scanners**

```
[propget, id(1)]  
HRESULT Scanners([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	The collection of Pressure Scanners of the DPS subsystem.

2.56 Interface “DPSScanners”**2.56.1 Description**

2.56.1.1 This interface describes the collection of the DSA Pressure Scanners.

2.56.2 Design

2.56.2.1 This interface shall be a dispatch interface.

2.56.2.2 This interface shall be an automation interface.

2.56.2.3 This interface shall not be directly created. It shall be created by the *Scanners* property of the interface *SubsystemDPS*.

2.56.2.4 This interface shall implement the collection interface for *LPDISPATCH*, returning the *DPSScanner* interfaces in the collection.

2.56.3 Methods and Properties**2.56.3.1 Property Find**

```
// Find a data record.  
[propget, id(1)]  
HRESULT Find([in] BSTR ScannerName, [out, retval] long *pVal);
```

Argument Name	Description
ScannerName	The name of the <i>DPSScanner</i> to be found.
*pVal	The index of the <i>DPSScanner</i> with the name <i>ScannerName</i> , if it is found in the collection, and 0 otherwise.

2.56.3.2 Property New

```
// Create a data record.  
[propget, id(1003)] HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>DPSScanner</i> .

2.56.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	Indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.57 Interface “DPSScanner”**2.57.1 Description**

2.57.1.1 This interface represents a DSA Pressure Scanner.

2.57.2 Design

2.57.2.1 This interface shall be a dispatch interface.

2.57.2.2 This interface shall be an automation interface.

2.57.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *DPSScanners*.

2.57.3 Methods and Properties**2.57.3.1 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 Scanner. The default value is an empty string. If a scanner name is specified twice, an error will be generated.

2.57.3.2 Property ScannerModel

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

Argument Name	Description
*pVal, newVal	Specifies the DSA Pressure Scanner series i.e. 3000 or 3200. The default value is an empty string. If an invalid value is specified, an error will be generated.

2.57.3.3 Property MaxScanRate

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

Argument Name	Description
*pVal, newVal	Specifies the DSA Pressure Scanner maximum scan rate. The default value is 0. The scan rates are defined by the Configuration Server option "ScanRates". Upper limits for the maximum are 33 Hz for DSA 3000 series and 200 Hz for DSA 3200 series Pressure Scanners. If the scan rate is outside the defined range, an error will be generated.

2.57.3.4 Property AvgSamples

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

Argument Name	Description
*pVal, newVal	Specifies the number of samples averaged for one scan value. Valid values are from 1 to 32767. The default is 0. If the average sample number is less than 1, an error will be generated.

2.57.3.5 Property Unit

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

Argument Name	Description
*pVal, newVal	Specifies the EU of the pressure values returned by the scanner. Valid values are <i>ATM, BAR, CMHG, CMH2O, DECIBAR, FTH2O, GCM2, INHG, INH2O, KGH2O, KGCM2, KGM2, KIPIN2, KNM2, KPA, MBAR, MH2O, MMHG, MPA, NCM2, NM2, OZFT2, OZIN2, PA, PSF, PSI, TORR</i> . If any other value than those listed is entered, <i>PSI</i> will be used. <i>PSI</i> is the default value.

2.57.3.6 Property Zone

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

Argument Name	Description
*pVal, newVal	Specifies the number of the pressure zone to which the DSA Pressure Scanner is assigned. The default value is 1.

2.57.3.7 Property Ports

```
[propget, id(7)] HRESULT Ports([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	The collection of ports defined for the Pressure Scanner.

2.58 Interface “DPSPorts”**2.58.1 Description**

2.58.1.1 This interface describes the collection of the ports for a DSA Pressure Scanner.

2.58.2 Design

2.58.2.1 This interface shall be a dispatch interface.

2.58.2.2 This interface shall be an automation interface.

2.58.2.3 This interface shall not be directly created. It shall be created by the *Ports* property of the interface *DPSScanner*.

2.58.2.4 This interface shall implement the collection interface for *LPDISPATCH*, returning the *DPSPort* interfaces in the collection.

2.58.3 Methods and Properties**2.58.3.1 Property Find**

```
// Find a data record.
[propget, id(1)]
HRESULT Find([in] long PortId, [out, retval] long *pVal);
```

Argument Name	Description
PortId	Port number to be found.
*pVal	The index of the <i>DPSPort</i> with the number <i>PortId</i> , if it is found in the collection, and 0 otherwise.

2.58.3.2 Property New

```
// Create a data record.
[propget, id(1003)] HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>DPSPort</i> .

2.58.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	Indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.59 Interface “DPSPort”**2.59.1 Description**

2.59.1.1 This interface represents a port of a DSA Pressure Scanner.

2.59.2 Design

2.59.2.1 This interface shall be a dispatch interface.

2.59.2.2 This interface shall be an automation interface.

2.59.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *DPSPorts*.

2.59.3 Methods and Properties**2.59.3.1 Property Port**

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

Argument Name	Description
*pVal, newVal	Specifies the pressure port on the DSA Pressure Scanner. The default value is 0. Valid values are from 1 to 16. If a value outside this range is specified, an error will be generated.

2.59.3.2 Property MaxPressure

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

Argument Name	Description
*pVal, newVal	Specifies the maximum pressure allowed on the port.

2.59.3.3 Property Description

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

Argument Name	Description
*pVal, newVal	Specifies description of the port. Default value is ""

2.60 Interface “SubsystemCONSORT”**2.60.1 Description**

2.60.1.1 This interface describes the CONSORT subsystem.

2.60.2 Design

2.60.2.1 This interface shall be a dispatch interface.

2.60.2.2 This interface shall be an automation interface.

2.60.2.3 The interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.60.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.60.3 Methods and Properties**2.60.3.1 Property HostName**

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

Argument Name	Description
*pVal, newVal	Specifies the host name of the CONSORT interface computer. The default value is an empty string. If this value is not defined, an error will be generated.

2.60.3.2 Property SteadyStatePeriod

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

Argument Name	Description
*pVal, newVal	Defines the period in seconds for the CONSORT to average the data. The default value is 5 seconds. Valid values are in the range 1.0 to 60.0 seconds. If a value outside this range is defined, an error will be generated.

2.60.3.3 Property PortLaneA

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

Argument Name	Description
*pVal, newVal	Specifies the port number expected for Lane A. The default value is 0. Valid values are from 0 to 7. If a value outside this range is defined, an error will be generated.

2.60.3.4 Property PortLaneB

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

Argument Name	Description
*pVal, newVal	Specifies the port number expected for Lane B. The default value is 5. Valid values are from 0 to 7. If a value outside this range is defined, an error will be generated.

2.61 Interface “SubsystemCEC”**2.61.1 Description**

2.61.1.1 This interface describes the CEC subsystem.

2.61.2 Design

2.61.2.1 This interface shall be a dispatch interface.

2.61.2.2 This interface shall be an automation interface.

2.61.2.3 The interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.61.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.61.3 Methods and Properties**2.61.3.1 Property Pinouts**

```
[propget] HRESULT Pinouts([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	The collection of pinouts defined for the CEC subsystem.

2.62 Interface “CECPinouts”**2.62.1 Description**

2.62.1.1 This interface represents the pinouts of a CEC subsystem.

2.62.2 Design

2.62.2.1 This interface shall be a dispatch interface.

2.62.2.2 This interface shall be an automation interface.

2.62.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *CECPinout* interfaces in the collection.

2.62.2.4 This interface shall not be directly created. It shall be created by the *Pinouts* property of the interface *SubsystemCEC*.

2.62.3 Methods and Properties**2.62.3.1 Property Find**

```
// Find a data record.
[propget] HRESULT Find([in] BSTR Name, [out, retval] long
*pVal);
```

Argument Name	Description
Name	The name of the <i>CECPinout</i> to be found.
*pVal	the index of the <i>CECPinout</i> with name <i>Name</i> , if it is found in the collection, and 0 otherwise.

2.62.3.2 Property New

```
// Create a data record.
[propget] HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	return the index of the newly created <i>CECPinout</i> .

2.62.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget] HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	Indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.63 Interface “CECPinout”**2.63.1 Description**

2.63.1.1 This interface represents a pinout of a CEC subsystem.

2.63.2 Design

2.63.2.1 This interface shall be a dispatch interface.

2.63.2.2 This interface shall be an automation interface.

2.63.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *CECPinouts*.

2.63.3 Methods and Properties**2.63.3.1 Property Name**

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

Argument Name	Description
*pVal, newVal	Name of the pinout. The maximum length is 40 characters. The default value is an empty string. The name must be unique in the CEC subsystem. If a name with more than 40 characters is tried to be set an error will be generated. If this value is not defined, an error will be generated.

2.63.3.2 Property Chassis

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

Argument Name	Description
*pVal, newVal	Number of the chassis where the pinout is located. Valid values are 1 and 2. The default value will be 1.

2.63.3.3 Property CardType

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

Argument Name	Description
*pVal, newVal	Type of the card where the pinout is located. Valid values are <i>Amplifier</i> and <i>Tachometer</i> . The default value will be an empty string.

2.63.3.4 Property Slot

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

Argument Name	Description
*pVal, newVal	Slot of the card where the pinout is located. It is only required for card type <i>Amplifier</i> . Valid values are numbers from 1 to 14. The default value will be 0.

2.64 Interface “SubsystemTSM”**2.64.1 Description**

2.64.1.1 This interface describes the TSM subsystem.

2.64.2 Design

2.64.2.1 This interface shall be a dispatch interface.

2.64.2.2 This interface shall be an automation interface.

2.64.2.3 The interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.64.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.64.3 Methods and Properties**2.64.3.1 Property Scanners**

```
[propget] HRESULT Scanners([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	The collection of temperature scanners defined for the TSM subsystem.

2.65 Interface “TSMScanners”**2.65.1 Description**

2.65.1.1 This interface represents the collection of TSM subsystem scanners.

2.65.2 Design

2.65.2.1 This interface shall be a dispatch interface.

2.65.2.2 This interface shall be an automation interface.

2.65.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *TSMScanner* interfaces in the collection.

2.65.2.4 This interface shall not be directly created. It shall be created by the *Scanners* property of the interface *SubsystemTSM*.

2.65.3 Methods and Properties**2.65.3.1 Property Find**

```
// Find a data record.  
[propget] HRESULT Find([in] long SerialNumber, [out, retval]  
long *pVal);
```

Argument Name	Description
SerialNumber	The serial number of the <i>TSMScanner</i> to be found.
*pVal	Index of the <i>TSMScanner</i> with serial number <i>SerialNumber</i> , if it is found in the collection, and 0 otherwise.

2.65.3.2 Property New

```
// Create a data record.  
[propget] HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	Index of the newly created <i>TSMScanner</i> .

2.65.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget] HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	Indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.66 Interface “TSMScanner”**2.66.1 Description**

2.66.1.1 This interface represents one scanner of the TSM subsystem.

2.66.2 Design

2.66.2.1 This interface shall be a dispatch interface.

2.66.2.2 This interface shall be an automation interface.

2.66.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *TSMScanners*.

2.66.3 Methods and Properties**2.66.3.1 Property SerialNumber**

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

Argument Name	Description
*pVal, newVal	Serial number of the scanner. Valid values are positive numbers. The default value will be 0. The serial number must be unique.

2.66.3.2 Property DefaultEngineeringUnit

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

Argument Name	Description
*pVal, newVal	Default engineering unit of the scanner. Valid values are <i>C</i> for Celsius and <i>F</i> for Fahrenheit and <i>K</i> for Kelvin (only for TSM 9146 scanners). The default value will be <i>C</i> .

2.66.3.3 Property ADSamples

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

Argument Name	Description
*pVal, newVal	The number of A/D samples to be averaged by the scanner, before it returns the channel data. Valid values are 1, 2, 4, 8, 16, 32 and 64. The default value will be 64 for the 9046 scanners and 32 for the 9146 scanners.

2.66.3.4 Property ReferenceAlarmLimit

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

Argument Name	Description
*pVal, newVal	Cold junction reference temperature deviation limit in degrees Celsius. This is the amount one reference temperature can deviate from the average of all reference temperatures. Valid values are numbers from 1 to 20. The default value will be 10.

2.66.3.5 Property ReferenceScanInterval

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

Argument Name	Description
*pVal, newVal	Scan interval in seconds for the thermal update task, which updates the cold junction temperature and voltages for other sensors. Valid values are numbers from 1 to 15. The default value will be 1.

2.66.3.6 Property Modules

```
[propget] HRESULT Modules([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	The collection of modules defined for the TSM scanner.

2.66.3.7 Property ScannerModel

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

Argument Name	Description
*pVal, newVal	Specifies the TSM Scanner series i.e. 9046 or 9146. The default value is an empty string. If an invalid value is specified, an error will be generated.

2.66.3.8 Property MaxScanRate

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

Argument Name	Description
*pVal, newVal	Specifies the TSM Scanner maximum scan rate. The default value is 0. The scan rates are defined by the Configuration Server option "ScanRates". The upper limit for maximum is 10 Hz for TSM 9046 series scanners. For TSM 9146 series scanners all available scan rates are allowed.

2.67 Interface “TSMModules”**2.67.1 Description**

2.67.1.1 This interface represents the collection of modules of a TSM subsystem scanner.

2.67.2 Design

2.67.2.1 This interface shall be a dispatch interface.

2.67.2.2 This interface shall be an automation interface.

2.67.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *TSMModule* interfaces in the collection.

2.67.2.4 This interface shall not be directly created. It shall be created by the *Modules* property of the interface *TSMScanner*.

2.67.3 Methods and Properties**2.67.3.1 Property Find**

```
// Find a data record.
[propget] HRESULT Find([in] long ChannelNumber, [out, retval]
long *pVal);
```

Argument Name	Description
ChannelNumber	The channel number of the <i>TSMModule</i> on the scanner to be found.
*pVal	Index of the <i>TSMModule</i> with channel number <i>ChannelNumber</i> , if it is found in the collection, and 0 otherwise.

2.67.3.2 Property New

```
// Create a data record.
[propget] HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	Index of the newly created <i>TSMModule</i> .

2.67.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget] HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	Indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.68 Interface “TSMModule”**2.68.1 Description**

2.68.1.1 This interface represents one module of a TSM subsystem scanner.

2.68.2 Design

2.68.2.1 This interface shall be a dispatch interface.

2.68.2.2 This interface shall be an automation interface.

2.68.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *TSMModules*.

2.68.3 Methods and Properties**2.68.3.1 Property ChannelNumber**

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

Argument Name	Description
*pVal, newVal	Number of the corresponding channel on the scanner. Valid values are numbers from 1 to 16. The default value will be 0. The number must be unique for the scanner.

2.68.3.2 Property Type

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

Argument Name	Description
*pVal, newVal	The type of module installed in the scanner. Valid values are <i>TC</i> for general thermocouple, <i>K</i> , <i>N</i> , <i>J</i> , <i>T</i> , <i>B</i> , <i>E</i> , <i>R</i> and <i>S</i> for the respective thermocouple type and <i>V</i> for Volts and <i>U</i> for User Defined (only for 9146 scanners). The default value will be an empty string.

2.68.3.3 Property Description

```
[propget, id(3)] HRESULT Description([out, retval] BSTR* pVal);
```

```
[propput, id(3)] HRESULT Description([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Specifies description of channel. Default value is "".

2.69 Interface "SubsystemM1553"**2.69.1 Description**

2.69.1.1 This interface describes the components of the M1553B subsystem.

2.69.2 Design

2.69.2.1 This interface shall be a dispatch interface.

2.69.2.2 This interface shall be an automation interface.

2.69.2.3 The interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.69.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.69.3 Methods and Properties**2.69.3.1 Property Networks**

```
[propget, id(1)]
HRESULT Networks([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	the networks defined in the subsystem. There is a maximum of 2 networks.

2.69.3.2 Property MaxStartWord

```
[propget, id(2)]
HRESULT MaxStartWord([in] BSTR Message, [out, retval] long
*pVal);
```

Argument Name	Description
Message	The name of the M1553B message.
*pVal	The max value of the start word for this message.

2.69.3.3 Property InputChannelRequired

```
[propget, id(3)]
HRESULT InputChannelRequired([in] BSTR Message, [out, retval]
long *pVal);
```

Argument Name	Description
Message	The name of the M1553B message.
*pVal	1 – Input Channel is required for the message. 0 – Input channel is not required.

2.69.3.4 Property OutputChannelRequired

```
[propget, id(4)]
HRESULT InputChannelRequired([in] BSTR Message, [out, retval]
long *pVal);
```

Argument Name	Description
Message	The name of the M1553B message.
*pVal	1 – Output Channel is required for the message. 0 – Output channel is not required.

2.69.3.5 Property AllMessagesOfType

```
[propget, id(4)]
HRESULT AllMessagesOfType ([in] long type, [out, retval]
VARIANT *pArrayVal);
```

Argument Name	Description
type	The type of the message to get. Valid values are: 0 – All messages which require an input channel. 1 – All messages feasible for input channels. 2 – All messages which require an output channel. 3 – All messages feasible for output channels. 4 – All messages
* pArrayVal	Array with message names.

2.70 Interface "M1553Networks"**2.70.1 Description**

2.70.1.1 This interface represents a collection of M1553 networks. A single M1553B network is defined in 2.71.

2.70.1.2 M1553B allows a maximum of two networks. Each network may contain a maximum of 31 nodes.

2.70.2 Design

2.70.2.1 This interface shall be a dispatch interface.

2.70.2.2 This interface shall be an automation interface.

2.70.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *M1553Network* interfaces in the collection.

2.70.2.4 This interface shall not be directly created. It shall be created by the *Networks* property of the interface *SubsystemM1553*.

2.70.3 Methods and Properties**2.70.3.1 Property Find**

```
// Find a data record.  
[propget, id(1)]  
HRESULT Find([in] long Id, [out, retval] long *pVal);
```

Argument Name	Description
Id	The Id of the network
*pVal	the index of the <i>M1553Network</i> , if it is found in the collection, and 0 otherwise.

2.70.3.2 Property New

```
// Create a data record.  
[propget, id(1003)]  
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created <i>M1553Network</i> .

2.70.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.71 Interface "M1553Network"**2.71.1 Description**

2.71.1.1 This interface represents a network of a M1553B subsystem.

2.71.2 Design

2.71.2.1 This interface shall be a dispatch interface.

2.71.2.2 This interface shall be an automation interface.

2.71.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *M1553Networks*.

2.71.3 Methods and Properties**2.71.3.1 Property PrimaryBus**

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

Argument Name	Description
*pVal, newVal	The primary bus of the network. This can be 'A' or 'B'

2.71.3.2 Property Id

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

Argument Name	Description
*pVal, newVal	The ID of the network. This property is only for internal use.

2.71.3.3 Property Nodes

```
[propget, id(3)]  
HRESULT Nodes([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	gives access to the M1553Nodes interface.

2.71.3.4 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(2000)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.72 Interface "M1553Nodes"**2.72.1 Description**

2.72.1.1 This interface represents a collection of M1553B nodes. A single node is defined in 2.73.

2.72.2 Design

2.72.2.1 This interface shall be a dispatch interface.

2.72.2.2 This interface shall be an automation interface.

2.72.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *M1553Node* interfaces in the collection.

2.72.2.4 This interface shall not be directly created. It shall be created by the *Nodes* property of the interface *M1553Network*.

2.72.3 Methods and Properties**2.72.3.1 Property Find**

```
[propget, id(1)]
HRESULT Find([in] BSTR Name, [out, retval] long *pVal);
```

Argument Name	Description
Name	The name of the M1553B node to be found.
*pVal	the index of the node with the specific name, if it is found in the collection, and 0 otherwise.

2.72.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created M1553Node.

2.72.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.73 Interface "M1553Node"**2.73.1 Description**

2.73.1.1 This interface represents an M1553B node, located on a M1553Network.

2.73.2 Design

2.73.2.1 This interface shall be a dispatch interface.

2.73.2.2 This interface shall be an automation interface.

2.73.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *M1553Nodes*.

2.73.3 Methods and Properties**2.73.3.1 Property Name**

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

Argument Name	Description
*pVal, newVal	the name of the node. The name must be unique across the entire SS.

2.73.3.2 Property Address

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

Argument Name	Description
*pVal, newVal	The address is a number from 0 to 30 inclusive. Duplicate addresses are not allowed. BC and BM nodes have an internal address of 31.

2.73.3.3 Property Type

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

Argument Name	Description
*pVal, newVal	the type of the node. Must be one of the following: BC, BM or RT. Maximum of 1 BC node, 1 BM node and 31 RT nodes.

2.73.3.4 Property IsSimulated

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

Argument Name	Description
*pVal, newVal	Indicates whether the node is simulated by proDAS or not. A BC node and a BM node cannot both be simultaneously selected for simulation by proDAS.

2.73.3.5 Property IsChanged

```
[propget, id(2000)]
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.73.3.6 Property Messages

```
[propget, id(5)]  
HRESULT Messages([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	gives access to the M1553Messages interface.

2.73.3.7 Property ScheduledIndex

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

Argument Name	Description
*pVal	Result of the calculation of the scheduled index on a per network/port basis. For BM and RT nodes this value is always 0, only for simulated BC nodes this value is greater than 0.

2.74 Interface "M1553Messages"**2.74.1 Description**

2.74.1.1 This interface represents a collection of M1553B messages. A single M1553B message is defined in 2.75.

2.74.2 Design

2.74.2.1 This interface shall be a dispatch interface.

2.74.2.2 This interface shall be an automation interface.

2.74.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *M1553Message* interfaces in the collection.

2.74.2.4 This interface shall not be directly created. It shall be created by the *Messages* property of the interface *M1553Node*.

2.74.3 Methods and Properties**2.74.3.1 Property Find**

```
// Find a data record.  
[propget, id(1)]  
HRESULT Find([in] BSTR Name, [out, retval] long *pVal);
```

Argument Name	Description
Name	The name of the M1553B message to be found.
*pVal	the index of the M1553B message with the name <i>Name</i> , if it is found in the collection, and 0 otherwise.

2.74.3.2 Property New

```
// Create a data record.  
[propget, id(1003)]  
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created message.

2.74.3.3 Property IsChanged

```
// Flag whether the data have been changed  
[propget, id(1005)]  
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.74.3.4 Property NodeType

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

Argument Name	Description
*pVal, newVal	the type of the parent node. Must be one of the following: BC, BM or RT and will be set automatically when setting the type of the parent node. This is an internal property to control the creation of new messages with the New property.

2.75 Interface "M1553Message"**2.75.1 Description**

2.75.1.1 This interface represents a M1553B message which is the base class of the interfaces M1553MessageBc, M1553MessageBm and M1553MessageRt.

2.75.2 Design

2.75.2.1 This interface shall be a dispatch interface.

2.75.2.2 This interface shall be an automation interface.

2.75.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *M1553Messages*.

2.75.3 Methods and Properties**2.75.3.1 Property Name**

```
[propget, id(100)]
HRESULT Name ([out, retval] BSTR *pVal);
[propput, id(100)]
HRESULT Name ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the name of the message. Must be unique across the entire SS.

2.75.3.2 Property Type

```
[propget, id(101)]
HRESULT Type ([out, retval] BSTR *pVal);
[propput, id(101)]
HRESULT Type ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	the type of the message. Must be one of the following: BCRT, RTBC, RTRT, RT_Tx, RT_Rx.

2.75.3.3 Property NumberOfWords

```
[propget, id(102)]  
HRESULT NumberOfWords ([out, retval] long *pVal);  
[propput, id(102)]  
HRESULT NumberOfWords ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the number of words of the message. Values range from 1 to 32 inclusive.

2.76 Interface "M1553MessageBc"**2.76.1 Description**

2.76.1.1 This interface represents a M1553B BC message which is derived from the interface M1553Message.

2.76.2 Design

2.76.2.1 This interface shall be a dispatch interface.

2.76.2.2 This interface shall be an automation interface.

2.76.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *M1553Messages*.

2.76.3 Methods and Properties**2.76.3.1 Property From**

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

Argument Name	Description
*pVal, newVal	The name of the From node. The type of the From node and the type of the To node set the type of the message, eg. BCRT.

2.76.3.2 Property FromSubAddress

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

Argument Name	Description
*pVal, newVal	Number from 1 to 30 inclusive

2.76.3.3 Property To

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

Argument Name	Description
*pVal, newVal	The name of the To node. The type of the From node and the type of the To node set the type of the message, eg. BCRT.

2.76.3.4 Property ToSubAddress

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

Argument Name	Description
*pVal, newVal	Number from 1 to 30 inclusive

2.76.3.5 Property Period

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

Argument Name	Description
*pVal, newVal	Any number in ms. All values must be integer multiples of the smallest positive value. Zero is a valid number.

2.77 Interface "M1553MessageBm"**2.77.1 Description**

2.77.1.1 This interface represents a M1553B BM message which is derived from the interface M1553Message.

2.77.2 Design

2.77.2.1 This interface shall be a dispatch interface.

2.77.2.2 This interface shall be an automation interface.

2.77.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *M1553Messages*.

2.77.3 Methods and Properties**2.77.3.1 Property From**

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

Argument Name	Description
*pVal, newVal	The name of the From node. The type of the From node and the type of the To node set the type of the message, eg. BCRT.

2.77.3.2 Property FromSubAddress

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

Argument Name	Description
*pVal, newVal	Number from 1 to 30 inclusive

2.77.3.3 Property To

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

Argument Name	Description
*pVal, newVal	The name of the To node. The type of the From node and the type of the To node set the type of the message, eg. BCRT.

2.77.3.4 Property ToSubAddress

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

Argument Name	Description
*pVal, newVal	Number from 1 to 30 inclusive

2.78 Interface "M1553MessageRt"**2.78.1 Description**

2.78.1.1 This interface represents a M1553B RT message which is derived from the interface M1553Message.

2.78.2 Design

2.78.2.1 This interface shall be a dispatch interface.

2.78.2.2 This interface shall be an automation interface.

2.78.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *M1553Messages*.

2.78.3 Methods and Properties

2.78.3.1 Property SubAddress

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

Argument Name	Description
*pVal, newVal	Number from 1 to 30 inclusive

2.79 Interface "SubsystemThg"**2.79.1 Description**

2.79.1.1 This interface will represent a THG subsystem.

2.79.2 Design

2.79.2.1 This interface shall be a dispatch interface.

2.79.2.2 This interface shall be an automation interface.

2.79.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.79.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.79.3 Methods and Properties**2.79.3.1 Property NumberOfWorkingCells**

```
// Define number of Working cells  
[propget, id(1)]  
HRESULT NumberOfWorkingCells([out, retval] long *pVal);  
[propput, id(1)]  
HRESULT NumberOfWorkingCells([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the number of working load cells configured for the THG subsystem. The property shall be set to the default value i.e. 0. If 0 is set an error will be generated.

2.79.3.2 Property NumberOfMasterCellsForward

```
// Define number of forward Master cells
[propget, id(2)]
HRESULT NumberOfMasterCellsForward([out, retval] long *pVal);
[propput, id(2)]
HRESULT NumberOfMasterCellsForward([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the number of forward master load cells configured for the THG subsystem. The property shall be set to the default value i.e. 0. If 0 is set an error will be generated.

2.79.3.3 Property NumberOfMasterCellsReverse

```
// Define number of reverse Master cells
[propget, id(3)]
HRESULT NumberOfMasterCellsReverse([out, retval] long *pVal);
[propput, id(3)]
HRESULT NumberOfMasterCellsReverse([in] long newVal);
```

Argument Name	Description
*pVal, newVal	the number of reverse master load cells configured for the THG subsystem. The property shall be set to the default value i.e. 0.

2.80 Interface "SubsystemOPC"**2.80.1 Description**

2.80.1.1 This interface will represent an OPC subsystem.

2.80.2 Design

2.80.2.1 This interface shall be a dispatch interface.

2.80.2.2 This interface shall be an automation interface.

2.80.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *Subsystems*.

2.80.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.80.3 Methods and Properties

2.80.3.1 Property Prefixes

```
[propget, id(3010)]  
HRESULT Prefixes([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	The prefixes defined in the subsystem. Gives access to the <i>OPCPrefixes</i> interface.

2.81 Interface “OPCPrefixes”

2.81.1 Description

2.81.1.1 This interface represents a collection of OPC prefixes. A single OPC prefix is defined in 2.82.

2.81.2 Design

2.81.2.1 This interface shall be a dispatch interface.

2.81.2.2 This interface shall be an automation interface.

2.81.2.3 This interface shall implement the collection interface for LPDISPATCH, returning the OPCPrefix interfaces in the collection.

2.81.2.4 This interface shall not be directly created. It shall be created by the Prefixes property of the interface “SubsystemOPC” (c.f. 2.80.3.1).

2.81.3 Methods and Properties

2.81.3.1 Property Find

```
// Find a data record.  
[propget, id(1012)]  
HRESULT Find([in] BSTR PrefixName, [out, retval] long *pVal);
```

Argument Name	Description
PrefixName	The name of the prefix

*pVal	The one-based index of the <i>OPCPrefix</i> if it is found in the collection, and 0 otherwise.
-------	--

2.81.3.2 Property New

```
// Create a data record for a prefix.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The one-based index of the newly created <i>OPCPrefix</i> .

2.81.3.3 Property IsChanged

```
// Flag whether the data have been changed
[propget, id(1005)]
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	Indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.82 Interface "OPCPrefix"

2.82.1 Description

2.82.1.1 This interface represents a prefix of an OPC subsystem.

2.82.2 Design

2.82.2.1 This interface shall be a dispatch interface.

2.82.2.2 This interface shall be an automation interface.

2.82.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *OPCPrefixes*.

2.82.3 Methods and Properties

2.82.3.1 Property Name

```
[propget, id(2006)]
HRESULT Name([out, retval] BSTR *pVal);
[propput, id(2006)]
HRESULT Name([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	The prefix. No spaces permitted. The maximum length is 19 characters.

2.82.3.2 Property Comment

```
[propget, id(2007)]
HRESULT Comment ([out, retval] BSTR *pVal);
[propput, id(2007)]
HRESULT Comment ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	A comment for describing the prefix.

2.82.3.3 Property Paths

```
[propget, id(2008)]
HRESULT Paths([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	Gives access to the <i>OPCPaths</i> interface.

2.82.3.4 Property IsChanged

```
// Flag whether the data have been changed
[propget, id(2000)]
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	Whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.83 Interface "OPCPaths"**2.83.1 Description**

2.83.1.1 This interface represents a collection of OPC paths. A single OPC path is defined in 2.84.

2.83.2 Design

2.83.2.1 This interface shall be a dispatch interface.

2.83.2.2 This interface shall be an automation interface.

2.83.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *OPCPath* interfaces in the collection.

2.83.2.4 This interface shall not be directly created. It shall be created by the *Paths* property of the interface *OPCPrefix* (c.f. 2.82.3.3).

2.83.3 Methods and Properties**2.83.3.1 Property Find**

```
// Find a data record.
[propget, id(1012)]
HRESULT Find([in] BSTR Name, [out, retval] long *pVal);
```

Argument Name	Description
Name	The name of the OPC path to be found.
*pVal	The index of the OPC path if it is found in the collection, and 0 otherwise.

2.83.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The one-based index of the newly created path.

2.83.3.3 Property IsChanged

```
// Flag whether the data have been changed
[propget, id(1005)]
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	Indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.84 Interface "OPCPath"**2.84.1 Description**

2.84.1.1 This interface represents an OPC path.

2.84.2 Design

2.84.2.1 This interface shall be a dispatch interface.

2.84.2.2 This interface shall be an automation interface.

2.84.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *OPCPaths*.

2.84.3 Methods and Properties**2.84.3.1 Property Name**

```
[propget, id(2006)]
HRESULT Name ([out, retval] BSTR *pVal);
[propput, id(2006)]
HRESULT Name ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	The name of the path. Branches shall be delimited using the period character ".".

2.84.3.2 Property Tags

```
[propget, id(2007)]
HRESULT Tags([out, retval] LPDISPATCH *pVal);
```


Argument Name	Description
*pVal	Gives access to the <i>OPCTags</i> interface.

2.84.3.3 Property IsChanged

```
// Flag whether the data have been changed
[propget, id(1005)]
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	Indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.85 Interface "OPCTags"

2.85.1 Description

2.85.1.1 This interface represents a collection of OPC tags. A single OPC tag is defined in 2.86.

2.85.2 Design

2.85.2.1 This interface shall be a dispatch interface.

2.85.2.2 This interface shall be an automation interface.

2.85.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *OPCTag* interfaces in the collection.

2.85.2.4 This interface shall not be directly created. It shall be created by the *Tags* property of the interface *OPCPath* (c.f.2.84.3.2).

2.85.3 Methods and Properties

2.85.3.1 Property Find

```
// Find a data record.
[propget, id(1012)]
HRESULT Find([in] BSTR Name, [out, retval] long *pVal);
```

Argument Name	Description
---------------	-------------

Name	The name of the OPC tag to be found.
*pVal	The index of the OPC tag if it is found in the collection, and 0 otherwise.

2.85.3.2 Property New

```
// Create a data record.
[propget, id(1003)]
HRESULT New([out, retval] long *pVal);
```

Argument Name	Description
*pVal	The index of the newly created tag.

2.85.3.3 Property IsChanged

```
// Flag whether the data have been changed
[propget, id(1005)]
HRESULT IsChanged([out, retval] BOOL *pVal);
```

Argument Name	Description
*pVal	Indicates whether the data have been changed since the data have been written back to the subsystem collection the last time. It shall initially be set to <i>false</i> .

2.86 Interface "OPCTag"

2.86.1 Description

2.86.1.1 This interface represents a tag of an OPC subsystem.

2.86.2 Design

2.86.2.1 This interface shall be a dispatch interface.

2.86.2.2 This interface shall be an automation interface.

2.86.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *OPCTags*.

2.86.3 Methods and Properties

2.86.3.1 Property Name

```
[propget, id(2006)]
HRESULT Name([out, retval] BSTR *pVal);
[propget, id(2006)]
HRESULT Name([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	The name of the tag. Must be unique across the current path. “#” is not allowed.

2.86.3.2 Property Comment

```
[propget, id(2007)]
HRESULT Comment ([out, retval] BSTR *pVal);
[propget, id(2007)]
HRESULT Comment ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	A comment for describing the tag.

2.86.4 Usage Conditions and Restrictions

2.86.4.1 The maximum characters allowed for the combination of the path and tag name is 127 characters.

2.87 Interface “SubsystemVEXA”

2.87.1 Description

2.87.1.1 This interface will represent VEXA subsystems.

2.87.2 Design

2.87.2.1 This interface shall be a dispatch interface.

2.87.2.2 This interface shall be an automation interface.

2.87.2.3 This interface shall not be directly instantiated. It shall be instantiated by the *Item* property of the interface *Subsystems*.

2.87.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.87.3 Methods and Properties**2.87.3.1 Property SignalConditionerModels**

```
[propget, id(4001)]
HRESULT SignalConditionerModels ([out, retval] LPDISPATCH
*pVal);
[propget, id(4001)]
HRESULT SignalConditionerModels ([in] LPDISPATCH newVal);
```

Argument Name	Description
*pVal, newVal	Definitions of signal conditioners models that can be applied to digital inputs. Returned instance implements interface <i>SignalConditionerModels</i>

2.87.3.2 Method LockSignalConditionerModels

```
[id(4002)]
HRESULT LockSignalConditionerModels ([in] BOOL lock);
```

Argument Name	Description
lock	0: unlock property SignalConditionerModels 1: lock property SignalConditionerModels

2.87.3.3 Property Devices

```
[propget, id(4003)]
HRESULT Devices ([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	The EX10xxA devices defined in the subsystem. Returned instance implements interface <i>VEXADevices</i>

2.87.4 Usage Conditions and Restrictions

- 2.87.4.1 Signal conditioners are independent from the SubsystemVEXA instance. Extending the list of conditioners will allow the use of the new signal conditioners in other VEXA subsystems.
- 2.87.4.2 If the property SignalConditionerModels ist set that data will be immediately made persistent (stored into a file).
- 2.87.4.3 Before setting the property SignalConditionerModels the method LockSignalConditionerModels has to be invoked with the parameter set to 1 (TRUE). Afterwards it should be invoked with the parameter set to 0 (FALSE).

2.88 Interface “VEXADevices”**2.88.1 Description**

- 2.88.1.1 This interface represents collections of EX10xxA devices. An EX10xxA device is defined in 2.89.

2.88.2 Design

- 2.88.2.1 This interface shall be a dispatch interface.
- 2.88.2.2 This interface shall be an automation interface.
- 2.88.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *VEXADevices* items in the collection.
- 2.88.2.4 This interface shall not be directly instantiated. It shall be created by the *Devices* property of the interface *SubsystemVEXA* (c.f.2.87.3.2).
- 2.88.2.5 The interface *Collection* shall be implemented as part of this interface by aggregation.

2.88.3 Methods and Properties**2.88.3.1 Property Find**

```
[propget, id(2001)]
HRESULT Find([in] BSTR HostName, [out, retval] long *pVal);
```

Argument Name	Description
HostName	Network host name of device

2.89 **Interface “VEXADevice”**

2.89.1 **Description**

2.89.1.1 This interface represents EX10xxA devices.

2.89.2 **Design**

2.89.2.1 This interface shall be a dispatch interface.

2.89.2.2 This interface shall be an automation interface.

2.89.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *VEXADevices*.

2.89.2.4 The interface *Item* shall be implemented as part of this interface by aggregation.

2.89.2.5 The methods *GetNumberOfChannels* and *GetValidChannelNumber* can be used to determine which channel numbers are valid for a certain category. Especially for CJC channels this would be convenient for clients.

2.89.3 **Methods and Properties**

2.89.3.1 Property HostName

```
[propget, id(3001)]
HRESULT HostName ([out, retval] BSTR *pVal);
[propput, id(3001)]
HRESULT HostName ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Network host name of the device. Must be defined in the /etc/hosts file of the RTE machine. . The maximum length is 39 characters.

2.89.3.2 Property Model

```
[propget, id(3002)]
HRESULT Model ([out, retval] long *pVal);
[propput, id(3002)]
HRESULT Model ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Device model type: <ul style="list-style-type: none"> EX1000A = 0 EX1000A-TC = 1 EX1016A = 2 EX1032A = 3 EX1048A = 4

2.89.3.3 Property UnitScanRate

```
[propget, id(3003)]
HRESULT UnitScanRate ([out, retval] long *pVal);
[propput, id(3003)]
HRESULT UnitScanRate ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Maximum scan rate of the device up to 200 Hz (default).

2.89.3.4 Property DefaultEu

```
[propget, id(3004)]
HRESULT DefaultEu ([out, retval] BSTR *pVal);
[propput, id(3004)]
HRESULT DefaultEu ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Engineering units of values returned for the thermocouple channels. Valid values are C (default) and F.

2.89.3.5 Property AnalogInputs

```
[propget, id(3010)]  
HRESULT AnalogInputs ([out, retval] BSTR *pVal);
```

Argument Name	Description
*pVal	Analog inputs of the device. Returned instance implements interface <i>VEXAAnalogInputs</i>

2.89.3.6 Method GetNumberOfChannels

```
[id(3020)]  
HRESULT GetNumberOfChannels ([in] long category, [out, retval]  
long* pVal);
```

Argument Name	Description
category	Channel category/type for which the number of channels shall be returned <ul style="list-style-type: none">Analog input = 0 (voltage or thermocouple input)CJC = 1DIO = 2
*pVal	Number of channels available for the category

2.89.3.7 Method GetValidChannelNumber

```
[id(3021)]  
HRESULT GetValidChannelNumber ([in] long category, [in] long  
index, [out, retval] long* pVal);
```


Argument Name	Description
category	Channel category/type for which the number of channels shall be returned <ul style="list-style-type: none"> Analog input = 0 (voltage or thermocouple input) CJC = 1 DIO = 2
index	1-based index of the channel number to be queried
*pVal	A valid channel number

2.90 Interface “VEXAAAnalogInputs”

2.90.1 Description

2.90.1.1 This interface represents collections of analog inputs of EX10xxA devices. A analog input is defined in 2.91.

2.90.2 Design

2.90.2.1 This interface shall be a dispatch interface.

2.90.2.2 This interface shall be an automation interface.

2.90.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *IVEXAAAnalogInput* items in the collection.

2.90.2.4 This interface shall not be directly instantiated. It shall be created by the *AnalogInputs* property of the interface *VEXADevice* (c.f.2.89.3.4).

2.90.2.5 The interface *Collection* shall be implemented as part of this interface by aggregation.

Note The items (analog inputs) is 1-based, but the channel number of an analog input is 0-based. Hence the index for channel 0 is 1. The 1-based indexing is inherited by the base interface *Collection*.

2.90.3 Methods and Properties

N/A

2.91 Interface “VEXAAAnalogInput”**2.91.1 Description**

2.91.1.1 This interface represents analog inputs of EX10xxA devices.

2.91.2 Design

2.91.2.1 This interface shall be a dispatch interface.

2.91.2.2 This interface shall be an automation interface.

2.91.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *VEXAAAnalogInputs*.

2.91.2.4 The interface *Item* shall be implemented as part of this interface by aggregation.

2.91.3 Methods and Properties**2.91.3.1 Property Enabled**

```
[propget, id(3001)]  
HRESULT Enabled ([out, retval] BOOL *pVal);  
[propput, id(3001)]  
HRESULT Enabled ([in] BOOL newVal);
```

Argument Name	Description
*pVal, newVal	Flag whether input is enabled and available for measurements or not

2.91.3.2 Property Type

```
[propget, id(3002)]  
HRESULT Type ([out, retval] long*pVal);  
[propput, id(3002)]  
HRESULT Type ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	<p>Type of the input. Following values are defined:</p> <ul style="list-style-type: none"> Voltage = 0 Thermocouple = 1 <p>The default selection of Voltage or Thermocouple will be determined by the ChannelNumber and the selected Model of the VEXA Device as per the table in the ES for the Configuration Server ES78031.3070 in section 3.1.48.17.</p>

2.91.3.3 Property SignalConditionerModel

```
[propget, id(3003)]
HRESULT SignalConditionerModel ([out, retval] BSTR *pVal);
[propput, id(3003)]
HRESULT SignalConditionerModel ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Model/type of the signal conditioner.

2.91.3.4 Property Description

```
[propget, id(3004)]
HRESULT Description ([out, retval] BSTR *pVal);
[propput, id(3004)]
HRESULT Description ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Description of the analog input channel

2.91.3.5 Property ChannelNumber

```
[propget, id(3005)]
HRESULT ChannelNumber ([out, retval] long* pVal);
```

Argument Name	Description
*pVal, newVal	Channel number of the analog input on the device

2.91.3.6 Property PhysicalType

```
[propget, id(3006)]
HRESULT PhysicalType([out, retval] long* pVal);
```

Argument Name	Description
*pVal, newVal	Physical type of the analog input on the device <ul style="list-style-type: none"> Voltage = 0 Thermocouple = 1

2.92 Interface “VEXASignalConditionerModels”**2.92.1 Description**

2.92.1.1 This interface represents collections of models/types of signal conditioners for EX10xxA devices. A single signal conditioner is defined in 2.93.

2.92.2 Design

2.92.2.1 This interface shall be a dispatch interface.

2.92.2.2 This interface shall be an automation interface.

2.92.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *SignalConditionerModel* interfaces in the collection.

2.92.2.4 This interface shall not be directly created. It shall be created by the *SignalConditionerModels* property of the interface *SubsystemVEXA* (c.f.2.87.3.1).

2.92.2.5 The interface *DataRecords* shall be implemented as part of this interface by aggregation.

2.92.3 Methods and Properties

N/A

2.93 Interface “VEXASignalConditionerModel”**2.93.1 Description**

2.93.1.1 This interface represents signal conditioner models/types for EX10xxA devices.

2.93.2 Design

2.93.2.1 This interface shall be a dispatch interface.

2.93.2.2 This interface shall be an automation interface.

2.93.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *VEXASignalConditionerModels*.

2.93.2.4 The interface *DataRecord* shall be implemented as part of this interface by aggregation.

2.93.3 Methods and Properties**2.93.3.1 Property Type**

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

Argument Name	Description
*pVal, newVal	Type of signal of signal conditioner input, free text, e.g. "Current input", "RMS voltage"

2.93.3.2 Property InputUnit

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

Argument Name	Description
*pVal, newVal	Unit of the input signal of the signal conditioner

2.93.3.3 Property InputRangeMin

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

Argument Name	Description
*pVal, newVal	Lower limit of input range

2.93.3.4 Property InputRangeMax

```
[propget, id(4005)]  
HRESULT InputRangeMax ([out, retval] float *pVal);  
[propput, id(4005)]  
HRESULT InputRangeMax ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	Upper limit of input range

2.93.3.5 Property OutputRangeMin

```
[propget, id(4006)]  
HRESULT OutputRangeMin ([out, retval] float *pVal);  
[propput, id(4006)]  
HRESULT OutputRangeMin ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	Lower limit of output range

2.93.3.6 Property OutputRangeMax

```
[propget, id(4007)]  
HRESULT OutputRangeMax ([out, retval] float *pVal);  
[propput, id(4007)]  
HRESULT OutputRangeMax ([in] float newVal);
```

Argument Name	Description
*pVal, newVal	Upper limit of output range

2.94 Interface “SubsystemDTS3250”**2.94.1 Description**

2.94.1.1 This interface will represent DTS3250 subsystems.

2.94.2 Design

2.94.2.1 This interface shall be a dispatch interface.

2.94.2.2 This interface shall be an automation interface.

2.94.2.3 This interface shall not be directly instantiated. It shall be instantiated by the *Item* property of the interface *Subsystems*.

2.94.2.4 The interface *Subsystem* shall be implemented as part of this interface by aggregation.

2.94.3 Methods and Properties**2.94.3.1 Property Devices**

```
[propget, id(4001)]  
HRESULT Devices ([out, retval] LPDISPATCH *pVal);
```

Argument Name	Description
*pVal	The DTS3250/xxTx devices defined in the subsystem. Returned instance implements interface <i>DTS3250Devices</i>

2.94.3.2 Method GetLowLimit

```
[id(4020)]  
HRESULT GetLowLimit([in] long tcType, [in] long limitType, [out,  
retval] float* pVal);
```

Argument Name	Description
tcType	Type of thermocouple input. Valid values are: <ul style="list-style-type: none"> • 'B' = 66 • 'E' = 69 • 'J' = 74 • 'K' = 75 • 'N' = 78 • 'R' = 82 • 'S' = 83 • 'T' = 84 (The ASCII code is used).
limitType	The default or overall low limit. Valid values are: <ul style="list-style-type: none"> • 0 = Overall low limit • 1 = Default low limit
*pVal	The default or overall low limit for the thermocouple type in Deg C.

2.94.3.3 Method GetHighLimit

```
[id(4021)]
HRESULT GetHighLimit([in] long tcType,[in] long limitType,
[out, retval] float* pVal);
```


Argument Name	Description
tcType	Type of thermocouple input. Valid values are: <ul style="list-style-type: none"> • 'B' = 66 • 'E' = 69 • 'J' = 74 • 'K' = 75 • 'N' = 78 • 'R' = 82 • 'S' = 83 • 'T' = 84 (The ASCII code is used).
limitType	The default or overall high limit. Valid values are: <ul style="list-style-type: none"> • 0 = Overall high limit • 1 = Default high limit
*pVal	The default or overall high limit for the thermocouple type in Deg C.

2.95 Interface “DTS3250Devices”

2.95.1 Description

2.95.1.1 This interface represents collections of DTS3250/xxTx devices. A DTS3250/xxTx device is defined in 2.96.

2.95.2 Design

2.95.2.1 This interface shall be a dispatch interface.

2.95.2.2 This interface shall be an automation interface.

2.95.2.3 This interface shall implement the collection interface for *LPDISPATCH*, returning the *DTS3250Devices* items in the collection.

2.95.2.4 This interface shall not be directly instantiated. It shall be created by the *Devices* property of the interface *SubsystemDTS3250* (c.f.2.94.3.1).

2.95.2.5 The interface *Collection* shall be implemented as part of this interface by aggregation.

2.95.3 Methods and Properties**2.95.3.1 Property Find**

```
[propget, id(2001)]
HRESULT Find([in] BSTR HostName, [out, retval] long *pVal);
```

Argument Name	Description
HostName	Network host name of device

2.96 Interface “DTS3250Device”**2.96.1 Description**

2.96.1.1 This interface represents a single DTS3250/xxTx device.

2.96.2 Design

2.96.2.1 This interface shall be a dispatch interface.

2.96.2.2 This interface shall be an automation interface.

2.96.2.3 This interface shall not be directly created. It shall be created by the *Item* property of the interface *DTS3250Devices*.

2.96.2.4 The interface *Item* shall be implemented as part of this interface by aggregation.

2.96.2.5 The method *GetNumberOfChannels* can be used to determine the maximum number of channels for a certain category (Thermocouple or RTD).

2.96.3 Methods and Properties**2.96.3.1 Property HostName**

```
[propget, id(3001)]
HRESULT HostName ([out, retval] BSTR *pVal);
[propput, id(3001)]
HRESULT HostName ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	Network host name of the device. Must be defined in the /etc/hosts file of the RTE machine. . The maximum length is 39 characters.

2.96.3.2 Property NumberOfChannels

```
[propget, id(3002)]  
HRESULT NumberOfChannels ([out, retval] long *pVal);  
[propput, id(3002)]  
HRESULT NumberOfChannels ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Number of channels on the scanner. <ul style="list-style-type: none">• 16 (DTS3250/16Tx) - Default• 32 (DTS3250/32Tx)• 64 (DTS3250/64Tx)

2.96.3.3 Property UnitScanRate

```
[propget, id(3003)]  
HRESULT UnitScanRate ([out, retval] long *pVal);  
[propput, id(3003)]  
HRESULT UnitScanRate ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Maximum scan rate of the device up to: <ul style="list-style-type: none">• 40 Hz for DTS3250/16Tx• 20 Hz for DTS3250/32Tx• 10 Hz for DTS3250/64Tx

2.96.3.4 Property AverageSamples

```
[propget, id(3004)]  
HRESULT AverageSamples ([out, retval] long *pVal);  
[propput, id(3004)]  
HRESULT AverageSamples ([in] long newVal);
```

Argument Name	Description
*pVal, newVal	Specifies the number of samples to be averaged for one scan value. Valid values are from 1 to 240

2.96.3.5 Property DefaultEu

```
[propget, id(3005)]
HRESULT DefaultEu ([out, retval] BSTR *pVal);
[propget, id(3005)]
HRESULT DefaultEu ([in] BSTR newVal);
```

Argument Name	Description
*pVal, newVal	<p>Engineering unit of values returned by the scanner. Valid values are:</p> <ul style="list-style-type: none"> • C (Degrees Celsius) - Default • F (Degrees Fahrenheit) • K (Degrees Kelvin) • R (Degrees Rankin) • V (Volts, raw uncorrected) • A (Volts, corrected by UTR temp) • 0 (Zero, Raw counts)

2.96.3.6 Method GetNumberOfChannels

```
[id(3020)]
HRESULT GetNumberOfChannels ([in] long category, [out, retval]
long* pVal);
```

Argument Name	Description
category	<p>Channel category/type for which the number of channels shall be returned</p> <ul style="list-style-type: none"> • Thermocouple = 0 • RTD = 1
*pVal	Number of channels available for the category