dSPACE

Platform Management API Reference

Release 2021-A – May 2021

dSPACE

How to Contact dSPACE

Mail:	dSPACE GmbH Rathenaustraße 26 33102 Paderborn Germany
Tel.:	+49 5251 1638-0
Fax:	+49 5251 16198-0
E-mail:	info@dspace.de
Web:	http://www.dspace.com

How to Contact dSPACE Support

If you encounter a problem when using dSPACE products, contact your local dSPACE representative:

- Local dSPACE companies and distributors: http://www.dspace.com/go/locations
- For countries not listed, contact dSPACE GmbH in Paderborn, Germany.
  Tel.: +49 5251 1638-941 or e-mail: support@dspace.de

You can also use the support request form: http://www.dspace.com/go/supportrequest. If you are logged on to mydSPACE, you are automatically identified and do not need to add your contact details manually.

If possible, always provide the relevant dSPACE License ID or the serial number of the CmContainer in your support request.

Software Updates and Patches

dSPACE strongly recommends that you download and install the most recent patches for your current dSPACE installation. Visit http://www.dspace.com/go/patches for software updates and patches.

Important Notice

Contents

About This Reference

Content

This reference gives you detailed information on the automation API for platform management (Automation API Version 2.0).

Note

The PDF version of this reference does not contain graphical representations of the API elements. For graphical representations, refer to dSPACE Help.

Required knowledge

You should be familiar with performing the tasks to be automated. For example, you should know how to manage a platform using AutomationDesk or ControlDesk.

Knowledge in handling the host PC and the Microsoft Windows operating system is presupposed. You should also be familiar with a programming language such as Python, C, or C#.

Symbols

dSPACE user documentation uses the following symbols:

Symbol	Description
DANGER	Indicates a hazardous situation that, if not avoided, will result in death or serious injury.
WARNING	Indicates a hazardous situation that, if not avoided, could result in death or serious injury.
CAUTION	Indicates a hazardous situation that, if not avoided, could result in minor or moderate injury.
NOTICE	Indicates a hazard that, if not avoided, could result in property damage.
Note	Indicates important information that you should take into account to avoid malfunctions.
Tip	Indicates tips that can make your work easier.

Symbol	Description
	Indicates a link that refers to a definition in the glossary, which you can find at the end of the document unless stated otherwise.
	Precedes the document title in a link that refers to another document.

Naming conventions

dSPACE user documentation uses the following naming conventions:

%name% Names enclosed in percent signs refer to environment variables for file and path names.

< > Angle brackets contain wildcard characters or placeholders for variable file and path names, etc.

Special folders

Some software products use the following special folders:

Common Program Data folder A standard folder for application-specific configuration data that is used by all users.

```
%PROGRAMDATA%\dSPACE\<InstallationGUID>\<ProductName>
```

or

```
%PROGRAMDATA%\dSPACE\<ProductName>\<VersionNumber>
```

Documents folder A standard folder for user-specific documents.

```
%USERPROFILE%\Documents\dSPACE\<ProductName>\
<VersionNumber>
```

Local Program Data folder A standard folder for application-specific configuration data that is used by the current, non-roaming user.

```
%USERPROFILE%\AppData\Local\dSPACE\<InstallationGUID>\
<ProductName>
```

Accessing dSPACE Help and PDF Files

After you install and decrypt dSPACE software, the documentation for the installed products is available in dSPACE Help and as PDF files.

dSPACE Help (local) You can open your local installation of dSPACE Help:

- On its home page via Windows Start Menu
- On specific content using context-sensitive help via **F1**

dSPACE Help (Web) You can access the Web version of dSPACE Help at www.dspace.com/go/help.

To access the Web version, you must have a *mydSPACE* account.

PDF files You can access PDF files via the  icon in dSPACE Help. The PDF opens on the first page.

Safety Precautions

Introduction	Read and follow the safety precautions carefully.

General Warning When Using the Platform Management API

Introduction	Note the following warning when using the Platform Management API.
Danger potential	**Risk of serious injury and/or property damage** Using this product can be dangerous. You must observe the following safety instructions and the relevant instructions in the user documentation. Using the dSPACE Platform Management API can have a direct effect on electronic systems connected to the platform accessed via the automation interface. Improper or negligent use can result in serious personal injury and/or property damage. Only persons who are qualified to use this software, and who have been informed of the above dangers and possible consequences, are permitted to use this product.
Liability	It is your responsibility to adhere to instructions and warnings. Any unskilled operation or other improper use of this product in violation of the respective safety instructions, warnings, or other instructions contained in the user documentation constitutes contributory negligence, which may lead to a limitation of liability by dSPACE GmbH, its representatives, agents and regional dSPACE companies, to the point of total exclusion, as the case may be. Any exclusion or limitation of liability according to other applicable regulations, individual agreements, and applicable general terms and conditions remain unaffected.

Data loss during operating system shutdown

The shutdown procedure of Microsoft Windows operating systems causes some required processes to be aborted although they are still being used by dSPACE software. To avoid data loss, the dSPACE software must be terminated manually before a PC shutdown is performed.

Introduction

Introduction

The dSPACE Platform Management API can be used to automate access to a dSPACE platform. You can register a platform and load a real-time application without applying another dSPACE software product, for example, ControlDesk.

Where to go from here

Information in this section

Basics on the Platform Management API

Introduction

Information on the supported platforms and limitations.

Supported platforms

The dSPACE Platform Management API supports the following dSPACE platforms:

- DS1006
- DS1007
- DS1104
- DS1403 (MicroAutoBox III)
- DS1202 (MicroLabBox)
- MABX (DS1401/MicroAutoBox II)
- SCALEXIO
- Multiprocessor systems (based on DS1006, DS1007, or SCALEXIO)
- VEOS (dSPACE platform for offline simulation), VEOS MC (multicore)

Tip

The supported platforms can be divided into two categories depending on the type of real-time application object they support:

- RealTimeApplication
  This object is used by DS1006, DS1104, MicroAutoBox II and multiprocessor systems based on DS1006.
- ControllableRealTimeApplication
  In contrast to the RealTimeApplication object, this object provides further methods, for example, `Start`, `Stop` and `Unload`.
  This object is used by DS1007, MicroLabBox, MicroAutoBox III, SCALEXIO and VEOS.

Limitations

PlatformManagement properties are not provided Without the context of another experiment software tool, the dSPACE Platform Manager server does not provide the following properties:

- `BalanceConnectedParametersOnly`
- `ResynchronizationEnabled`
- `ResynchronizationRate`

Migrating applications

Replacing the program ID You have to change the progam ID in applications to migrate to the Platform Management API coming with dSPACE Release 2016-B and later:

Replace `DSPlatformManagementAPI` or `DSPlatformManagementAPI2` by `DSPlatformManagementAPI3`

Basics on the Object Model

Introduction

The object model of the platform management consists of different kinds of elements.

Collection elements

Collection elements provide access to a list of elements of the same type.

Collection elements can provide methods such as the following:

- `Add`: Lets you create and add an element to the collection.
- `Contains`: Lets you check wether a specific element is a member of the collection.
- `Item`: Lets you access a specific element of the collection by it's name or index.

Collection elements can provide properties such as the following:

- `Count`: Provides the number of elements of the collection.
- `Elements`: Lets you access all the elements of the collection.

Example For example, you can use the `SeekedPlatforms` collection to get a specific platform object. Collection element names are plural.

The following listing shows how you can get the object of a registered platform.

```
# Get a registered platform using the Item method of the
SeekedPlatforms collection.
MyPlatform = SeekedPlatforms.Item("<UniqueName>")
```

Interface elements

Interface elements provide access to properties and methods of an object.

Methods provide a functionality, for example, to load a real-time application to the platform, and can return objects and values.

Properties can be used to get a value from or set a value to a specific value of the interface element.

Example To get the board details of a DS1006 board, you can use the `DS1006BoardDetails` property.

To stop the running real-time application, you can use the `StopRTP` method.

Enumerations

Enumeration elements provide access to a set of named constants. Each constant can be accessed via its value or via its name.

Example To select a platform/device type, for example, you use a constant of the PlatformType enumeration element:

Name	Description	Value
MABX	MicroAutoBox platform (MicroAutoBox II)	0
DS1006	DS1006 platform	17
...	...	...

The following listing shows how you can select a DS1006 platform via its name.

```
import dspace.com
PlatformManagement =
win32com.client.Dispatch("DSPlatformManagementAPI3")

# Define Enums object
PMEnums = dspace.com.Enums(PlatformManagement)

# Create RegistrationInfo object
RegInfo = PlatformManagement.CreatePlatformRegistrationInfo(
     PMEnums.PlatformType.DS1006)

# Set ConnectionType
RegInfo.ConnectionType = PMEnums.InterfaceConnectionType.Bus

# Set PortAddress
RegInfo.PortAddress = 0x300

# Register platform
myPlatform = PlatformManagement.RegisterPlatform(RegInfo)
```

Syntax

Introduction	Examples of accessing properties and methods of Platform Management elements in Python and C#.

Python	**Getting the value of a property**
	`value = RegistrationInfo.PortAddress`
	Setting the value of a property
	`RegistrationInfo.PortAddress = value`
	Calling a method without parameters
	`RealTimeApplication.Start()`
	Calling a method with parameters
	`returnValue = PlatformManagement.RegisterPlatform(RegInfo)`

C#	**Getting the value of a property**
	`value = RegistrationInfo.PortAddress;`
	Setting the value of a property
	`RegistrationInfo.PortAddress = value;`
	Calling a method without parameters
	`RealTimeApplication.Start();`

Calling a method with parameters

```
returnValue = PlatformManagement.RegisterPlatform(RegInfo);
```

Basics on Types in the Object Model

Introduction

Platform management elements have properties and provide methods that work with different types.

Using platform management features

To use platform management features via automation you have to access elements, specify element settings and execute element methods. To perform these tasks the automation interface provides properties and methods for platform management elements.

Properties and methods provide access to information via objects and let you get and/or modify the information value.

- Properties can get objects and set and/or get values.
- Methods can return objects or values.
- Methods can require parameters that are objects or values.

Types of handles

The dSPACE Platform Management API provides you access to information, like the names of all the elements of a collection, or to elements, like a platform, via handles.

The dSPACE Platform Management API provides you access to the following types via objects:

- Platform management elements
- String lists

Types of values

The dSPACE Platform Management API lets you get and/or modify information that is represented by the following types:

- bool
- double
- float
- int
- string
- date

Automating Platform Management with Different Languages

Introduction

To automate platform management via its automation interface using different programming languages.

Specific instructions

You find specific information for the following programming languages:

- Python
  Refer to Using Python on page 18.
- C#
  Refer to Using C# on page 20.
- Visual Basic
  Refer to Using VB on page 20.
- MATLAB M code
  Refer to Using M on page 20.

Using Python

Migrating to Python 3.9 The support of Python 3.6 was discontinued with dSPACE Release 2021-A. Python 3.9 is now supported.

For information on changes and migration aspects of Python scripts in dSPACE products, refer to Migrating Python Scripts from Python 2.7 to Python 3.6 (New Features and Migration 🕮).

Importing required modules to the interpreter namespace The following listing shows how you can import required modules to the interpreter namespace.

```
#-------------------------------------------------------------------------------
# Import Python library modules to be used.
# The Dispatch class is used to create objects.

# The Enums class is used to create an object with all enumerations for a
# typelibrary given by an automation object.
# The os module is used for path concatenations.
# The sys module is used to get the command line arguments.
# The exceptions module is used to catch unexpected exceptions.
#-------------------------------------------------------------------------------
from win32com.client import Dispatch
from dspace.com import Enums
import os
import sys
import exceptions
```

Opening a COM connection to the platform management server The following listing shows how you can open a COM connection to the platform management server to use its API functions.

```
class PlatformManagement():
    def __init__(self):
        # the PlatformManagement object
        self.PlatformManagement = None

        # the enums for PlatformManagement object model
        self.Enums = None

    def Initialize(self):
        print("Init")
        # get PlatformManagement object
        self.PlatformManagement = Dispatch("DSPlatformManagementAPI3")
        self.PlatformManagement.RefreshPlatformConfiguration()
        # get the enums object
        self.Enums = Enums(self.PlatformManagement)

    def GetInfoOfRegisteredPlatforms(self):
        print("Registered platforms count: %i"
%(self.PlatformManagement.Platforms.Count))
        for eachPlatform in self.PlatformManagement.Platforms:
            print("-------------------------------------")
            print("UniqueName: %s" %(eachPlatform.UniqueName))
            print("PlatformType: %s\n"
%(self.Enums.PlatformType(eachPlatform.Type)))

def ExecuteDemo():
    myPlatformManagement = None

    myPlatformManagement = PlatformManagement()
    myPlatformManagement.Initialize()
    myPlatformManagement.GetInfoOfRegisteredPlatforms()


# Main program
if __name__ == "__main__":
    ExecuteDemo()
```

The `Dispatch("DSPlatformManagementAPI3")` command opens a COM connection to the PlatformManagement server. When you execute a demo including the listing above in an external Python interpreter, the PlatformManagement server is started if it is not yet running.

Structuring Python scripts The following script structure is useful for scripts that automate platform management:

1. Import the required modules, such as `os` or `win32com`.
2. Define an `Enums` object like this:

   ```
   self.Enums = Enums(self.PlatformManagement)
   ```

3. Define functions using platform management features.

   Define objects using the `Enums` object:

   ```
   DS1006RegInfo =
   self.PlatformManagement.CreatePlatformRegistrationInfo(
             self.Enums.PlatformType.DS1006)
   ```

4. Call the required functions in a `Main()` routine.

Using C#

Automating platform management using Microsoft Visual Studio and C# The following instructions show how you can instantiate platform management to use its API functions.

To automate platform management using Microsoft Visual Studio and C#:

1. Open Visual Studio (Express) and select **File – New Project** from the menu.
2. Specify the project and select **Visual C#** as the language and an application type template (the demos use Windows Forms).
3. In Visual Studio's **Solution Explorer**, right-click **References** and browse for the DLLs in the `./PlatformManagementAPI/Main/bin/AutomationAssemblies` folder of your XIL API .NET installation. Add all assemblies to your project.
4. Add the following listing to your project's program code:

```
// Start the platform management server
Type serverType =
    Type.GetTypeFromProgID("DSPlatformManagementAPI3");
this.PlatformManagement =
    Activator.CreateInstance(serverType) as IPmPlatformManagement;
```

5. Build and run the solution to instantiate the platform manager server via its automation interface.

Using VB

The following listing shows how you can open a COM connection to the platform management server to use its API functions.

```
' The PlatformManagement
Private application As IPmPlatformManagement

' Creates the connection to the platform management server.
Set PlatformManagement = CreateObject("DSPlatformManagementAPI3")
```

Using M

The following listing shows how you can open a COM connection to the platform management server to use its API functions.

```
PlatformManagement = actxserver('DSPlatformManagementAPI3')
```

The `actxserver('DSPlatformManagementAPI3')` command opens a COM connection to the platform management server. When you execute the listing above in MATLAB's Command Window, the platform management server is started if it is not yet running.

Using the dSPACE Platform Management API Reference when Programming with Python

Introduction

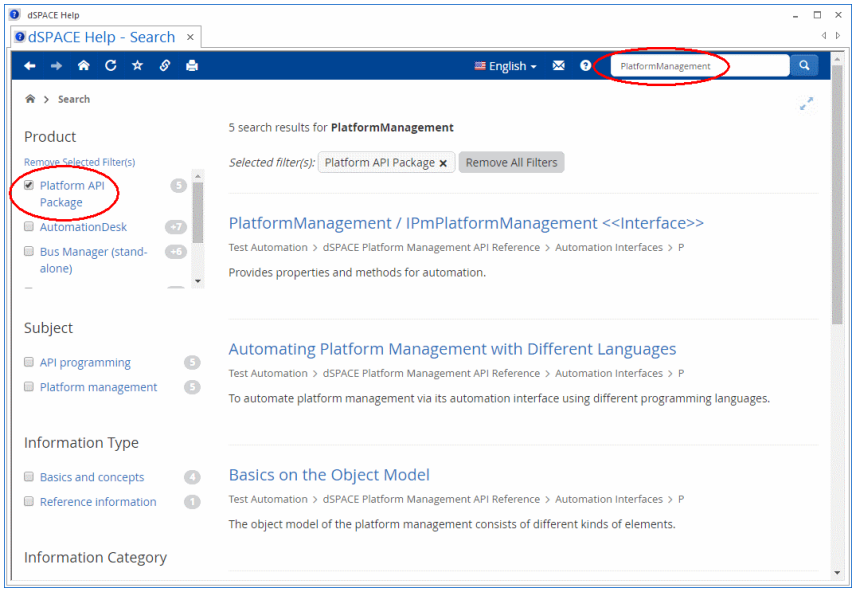
In scripting languages such as Python, interface types such as `IPmPlatformManagement` are not used. You usually work with object *properties* and *methods* instead.

Finding information on interfaces

As a Python programmer you find the description of the appropriate interface by recapitulating the way you have created the element, starting with the IPmPlatformManagement interface.

Tip

You can search in dSPACE Help to find a specific interface, e.g., the `IPmPlatformManagement` interface. Enter the interface name without prefix and start the search. Then set Platform API Package as the product filter.



Tip

- To find the interface name in your Python code, you can use the `GetComIdentity` function in the `dspace.com` module. As an example, the code below outputs the interface name of the `MyDS1006Platform` object:

```
import dspace.com
print(dspace.com.GetComIdentity(MyDS1006Platform))
Output: IPmDS1006Platform
```

- To work with enumerations in Python, you can use the `Enums` function in the `dspace.com` module. As an example, the code below outputs the current value of the `PlatformType` enumeration of the specified platform.

```
import dspace.com
Enums = dspace.com.Enums(PlatformManagement)
print(Enums.PlatformType(PlatformManagement.Platforms[0].Type))
Output: DS1006
```

Automation Interfaces

Where to go from here

Information in this section

A

Where to go from here

Information in this section

ApplicationPart / IPmApplicationPart <<Interface>>

Description

Interface for accessing an application part.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
ID	Gets the application part ID generated at compile time.	Get	*String*
IPAddress	Gets the IP address of the processing unit running the application part.	Get	*String*
Name	Gets the name of the real-time application.	Get	*String*
Properties	Provides access to the supported property collection of the realtimeapplication properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RuntimeID	Gets the run-time ID of the application part generated during application load.	Get	*Unsigned 32 Bit Integer*

Name	Description	Get/Set	Type
State	Gets the state of the real-time application.	Get	ApplicationState (refer to ApplicationState <<Enumeration>> on page 27)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmApplicationParts (refer to ApplicationParts / IPmApplicationParts <<Collection>> on page 25)

ApplicationParts / IPmApplicationParts <<Collection>>

Description

IPmRealTimeApplication Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of real-time applications in the collection. [0..System.Int32 -1]	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Returns True if an element that is accessible by the specified string is in the collection.	▪ *<String>* `RTAppName`: Name to access a real-time application	True if the real-time application exists. ▪ *Boolean*
Item	Returns a real-time application from the collection.	▪ *<Object>* `Index`: System.Object; either integer index or string with I/O platform name	Return value of the method. ▪ IPmApplicationPart (refer to ApplicationPart / IPmApplicationPart <<Interface>> on page 24)

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmControllableRealTimeApplication (refer to ControllableRealTimeApplication / IPmControllableRealTimeApplication <<Interface>> on page 35)
- IPmDS1007ProcessingUnit (refer to DS1007ProcessingUnit / IPmDS1007ProcessingUnit <<Interface>> on page 43)
- IPmDS1202ProcessingUnit (refer to DS1202ProcessingUnit / IPmDS1202ProcessingUnit <<Interface>> on page 48)
- IPmDS1403ProcessingUnit (refer to DS1403ProcessingUnit / IPmDS1403ProcessingUnit <<Interface>> on page 52)
- IPmSCALEXIOProcessingUnit (refer to SCALEXIOProcessingUnit / IPmSCALEXIOProcessingUnit <<Interface>> on page 148)
- IPmVEOSApplication (refer to VEOSApplication / IPmVEOSApplication <<Interface>> on page 160)
- IPmVEOSProcessingUnit (refer to VEOSProcessingUnit / IPmVEOSProcessingUnit <<Interface>> on page 164)

ApplicationProcess / IPmApplicationProcess <<Interface>>

Description

Application process interface for platform automation interfaces.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
ActiveVariableDescription	Gets the platform's active variable description.	Get	*dSPACE.ToolAutomation.ControlDeskNG.IXaActiveVariableDescription*
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*
Name	Gets the name of the platform used in an experiment.	Get	*String*
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Name	Description	Get/Set	Type
VariableDescriptions	Gets the list of variable descriptions available for the platform.	Get	*dSPACE.ToolAutomation.ControlDeskNG.IXaVariableDescriptions*

Methods

The element has no methods.

ApplicationState <<Enumeration>>

Description

Real-time application state enumeration

Enumeration values

The enumeration has the following values:

Name	Description	Value
Undefined	Real-time application state is undefined	0
Running	The real-time application is running	1
Stopped	The real-time application is stopped	2
Terminated	The real-time application is terminated	3
RunningFromFlash	The real-time application is running from flash.	4
Paused	The offline simulation application is paused (used by VEOS).	5
Unknown	The real-time application state is unknown.	6
StoppedFromFlash	The real-time application state is stopped from flash.	7
Initialized	The real-time application state is initialized.	8

Returned by

The element is returned by properties or methods of the following elements:

- IPmApplicationPart (refer to ApplicationPart / IPmApplicationPart <<Interface>> on page 24)
- IPmControllableRealTimeApplication (refer to ControllableRealTimeApplication / IPmControllableRealTimeApplication <<Interface>> on page 35)
- IPmVEOSApplication (refer to VEOSApplication / IPmVEOSApplication <<Interface>> on page 160)

AssignmentMode <<Enumeration>>

Description Assignment mode enumeration

Enumeration values The enumeration has the following values:

Name	Description	Value
UseFirstAvailablePlatform	Assign to first available platform	0
UseEqualPlatform	Assign to any equal platform	1
UseIdenticalPlatform	Assign to identical platform	2

AutomaticReconnectBehavior <<Enumeration>>

Description AutomaticReconnectBehavior Type Enumeration

Enumeration values The enumeration has the following values:

Name	Description	Value
ResumeMeasurementAndCalibration	Measurement and calibration are resumed if hardware is reconnected.	1
ResumeMeasurement	Only measurement is resumed if hardware is reconnected. Calibration is disabled.	2
DisconnectDevice	Device will be disconnected if connection is lost	4

Returned by The element is returned by properties or methods of the following elements:

- IPmCalibrationPlatformGeneralSettings (refer to CalibrationPlatformGeneralSettings / IPmCalibrationPlatformGeneralSettings <<Interface>> on page 30)

AutomationAPIVersion <<Enumeration>>

Description

AutomationAPIVersion enumeration type.

Enumeration values

The enumeration has the following values:

Name	Description	Value
APIVersion1	Use deprecated Automation API Version 1.0	1
APIVersion2	Use current Automation API Version 2.0	2

Returned by

The element is returned by properties or methods of the following elements:

- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)

C

Where to go from here

Information in this section

CalibrationPlatformGeneralSettings / IPmCalibrationPlatformGeneralSettings <<Interface>>

Description

IPmCalibrationPlatformGeneralSettings Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
AutomaticReconnectBehavior	Gets or sets the platform's behavior when the interface connection is lost/reestablished.	Get/Set	AutomaticReconnectBehavior (refer to AutomaticReconnectBehavior <<Enumeration>> on page 28)
ConfirmAutomaticPageSwitch	Gets or sets the ConfirmAutomaticPageSwitch.	Get/Set	*Boolean*
CrossRead	Gets or sets the CrossRead.	Get/Set	*Boolean*
DisplayStatusInformation	Gets or sets whether dialogs are displayed when the interface connection is lost or reestablished.	Get/Set	*Boolean*
EnablePlatform	Enables / disables the platform / device.	Get/Set	*Boolean*

Name	Description	Get/Set	Type
EnableResynchronizePlatform	Gets or sets the platform's local resynchonization when global resynchonization is enabled.	Get/Set	*Boolean*
InitialPage	InitialPageType enumeration type.	Get/Set	InitialPageType (refer to InitialPageType <<Enumeration>> on page 68)
ResumeOnlineCalibrationBehavior	Gets or sets the platform's behavior when resuming online calibration after reestablishing the interface connection.	Get/Set	OnlineCalibrationBehavior (refer to OnlineCalibrationBehavior <<Enumeration>> on page 86)
StartOnlineCalibrationBehavior	OnlineCalibrationBehavior type enumeration. Gets or sets the platform's behavior during online transition.	Get/Set	OnlineCalibrationBehavior (refer to OnlineCalibrationBehavior <<Enumeration>> on page 86)
VerifyPageConsistency	Gets or sets the VerifyPageConsistency.	Get/Set	*Boolean*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
SupportsResumeOnlineCalibrationBehavior	Checks whether the specified resuming behavior is supported by the platform.	▪ <OnlineCalibrationBehavior (refer to OnlineCalibrationBehavior <<Enumeration>> on page 86)> `Behavior`: The desired resuming transition behavior.	Returns True if the desired resuming behavior is possible. Otherwise, returns False. ▪ *Boolean*
SupportsStartOnlineCalibrationBehavior	Checks whether the specified online transition behavior is supported by the platform.	▪ <OnlineCalibrationBehavior (refer to OnlineCalibrationBehavior <<Enumeration>> on page 86)> `Behavior`: The desired online transition behavior.	Returns True if the desired online transition behavior is possible. Otherwise, returns False. ▪ *Boolean*

1) <Type> `Name`: Description

CalibrationPlatformProtocolLogging / IPmCalibrationPlatformProtocolLogging <<Interface>>

Description IPmCalibrationPlatformProtocolLogging Interface

Properties The element has the following properties:

Name	Description	Get/Set	Type
Communication	Protocol communication logging enumeration.	Get/Set	ProtocolCommunicationLogging (refer to ProtocolCommunicationLogging <<Enumeration>> on page 116)
Configuration	Protocol configuration logging enumeration.	Get/Set	ProtocolConfigurationLogging (refer to ProtocolConfigurationLogging <<Enumeration>> on page 116)
LogFilesDirectoryName	Gets or sets the LogFilesDirectoryName.	Get/Set	*String*
MaximumFileSize	Gets or sets the MaximumFileSize.	Get/Set	*Signed 32 Bit Integer*

Methods The element has no methods.

CalibrationPlatformTransportLayerLogging / IPmCalibrationPlatformTransportLayerLogging <<Interface>>

Description IPmCalibrationPlatformTransportLayerLogging Interface

Properties The element has the following properties:

Name	Description	Get/Set	Type
Communication	Transport layer communication logging enumeration.	Get/Set	TransportLayerCommunicationLogging (refer to TransportLayerCommunicationLogging <<Enumeration>> on page 155)
Configuration	Transport layer configuration logging enumeration.	Get/Set	TransportLayerConfigurationLogging (refer to TransportLayerConfigurationLo

Name	Description	Get/Set	Type
			gging <<Enumeration>> on page 156)
LogFilesDirectoryName	Gets or sets the LogFilesDirectoryName.	Get/Set	*String*
MaximumFileSize	Gets or sets the MaximumFileSize.	Get/Set	*Signed 32 Bit Integer*

Methods

The element has no methods.

Capabilities <<Enumeration>>

Description

Capabilities Type Enumeration

The enumeration describes the capability of a platform to calibrate and/or to measure.

Enumeration values

The enumeration has the following values:

Name	Description	Value
None	The platform can neither calibrate nor measure.	0
Calibration	The platform can only calibrate.	1
Measurement	The platform can only measure.	2
MeasurementAndCalibration	The platform can calibrate and measure.	3

Returned by

The element is returned by properties or methods of the following elements:

- IPmSubstitutePlatform (refer to SubstitutePlatform / IPmSubstitutePlatform <<Interface>> on page 153)

ConnectionState <<Enumeration>>

Description

Connection State Enumeration

Enumeration values

The enumeration has the following values:

Name	Description	Value
Connected	Platform is connected to hardware or data source. This state is a precondition for calibration and measurement.	0
Disconnected	Platform is not connected to hardware or data source. Only offline calibration is possible.	1

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1007ProcessingUnit (refer to DS1007ProcessingUnit / IPmDS1007ProcessingUnit <<Interface>> on page 43)
- IPmDS1202ProcessingUnit (refer to DS1202ProcessingUnit / IPmDS1202ProcessingUnit <<Interface>> on page 48)
- IPmDS1403ProcessingUnit (refer to DS1403ProcessingUnit / IPmDS1403ProcessingUnit <<Interface>> on page 52)
- IPmRegisteredDS1006Platform (refer to RegisteredDS1006Platform / IPmRegisteredDS1006Platform <<Interface>> on page 123)
- IPmRegisteredDS1104Platform (refer to RegisteredDS1104Platform / IPmRegisteredDS1104Platform <<Interface>> on page 128)
- IPmRegisteredMABXPlatform (refer to RegisteredMABXPlatform / IPmRegisteredMABXPlatform <<Interface>> on page 134)
- IPmRegisteredMultiProcessorPlatform (refer to RegisteredMultiProcessorPlatform / IPmRegisteredMultiProcessorPlatform <<Interface>> on page 136)
- IPmSCALEXIOProcessingUnit (refer to SCALEXIOProcessingUnit / IPmSCALEXIOProcessingUnit <<Interface>> on page 148)
- IPmSubstitutePlatform (refer to SubstitutePlatform / IPmSubstitutePlatform <<Interface>> on page 153)
- IPmVEOSProcessingUnit (refer to VEOSProcessingUnit / IPmVEOSProcessingUnit <<Interface>> on page 164)

ControllableRealTimeApplication / IPmControllableRealTimeApplication <<Interface>>

Description IPmControllableRealTimeApplication Interface

Properties The element has the following properties:

Name	Description	Get/Set	Type
ApplicationParts	Returns the application parts of the real-time application.	Get	IPmApplicationParts (refer to ApplicationParts / IPmApplicationParts <<Collection>> on page 25)
BuildDateTime	Returns the time when the real-time application was built.	Get	*Date Time*
FullPath	Returns the path of the loaded real-time application The returned path is the original file path to the loaded application, which might not be suitable for the current file system.	Get	*String*
Name	Returns the name of the real-time application.	Get	*String*
Properties	Provides access to the supported property collection of the realtimeapplication properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
State	Returns the state of the real-time application.	Get	ApplicationState (refer to ApplicationState <<Enumeration>> on page 27)

Methods The element has the following methods:

Name	Description	Parameter[1)]	Returns
Start	Starts the real-time application. The method just returns whether the real-time application already started without an exception. To start the real-time application, you need shared access to it.	None	None
Stop	Stops the real-time application The method just returns whether the real-time application already stopped without an exception. To stop the real.time application, you need shared access to it.	None	None

Name	Description	Parameter[1)]	Returns
Unload	Unloads the real-time application To unload the real-time application, you need exclusive access to it.	None	None

[1)] <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmRealTimeApplications (refer to RealTimeApplications / IPmRealTimeApplications <<Collection>> on page 119)

D

Where to go from here

Information in this section

DAQIntelByteOrder <<Enumeration>>

Description DAQIntelByteOrder enumeration type.

Enumeration values The enumeration has the following values:

Name	Description	Value
Forward	Forward	0
Sequential	Sequential	1

DAQMotorolaByteOrder <<Enumeration>>

Description DAQMotorolaByteOrder enumeration

Enumeration values The enumeration has the following values:

Name	Description	Value
ForwardMSB	Forward MSB	0
ForwardLSB	Forward LSB	1
Backward	Backward	2
Sequential	Sequential	3

DS1006BoardDetails / IPmDS1006BoardDetails <<Interface>>

Description IPmDS1006BoardDetails Interface (RCP/HIL Device)

Properties The element has the following properties:

Name	Description	Get/Set	Type
BoardVersion	Returns the board version of the hardware.	Get	*String*
BusFrequency	Returns the bus frequency of the hardware.	Get	*Double*
IsMultiCoreHardware	Gets a value indicating whether the hardware is multicore.	Get	*Boolean*
NumberOfAvailableCores	Gets the number of available cores.	Get	*Signed 32 Bit Integer*
PortAddress	Property to get/set the port address	Get	*Signed 32 Bit Integer*
ProcessorFrequency	Returns the processor frequency of the hardware.	Get	*Double*
ProcessorState	Returns the processor state of the hardware.	Get	ProcessorState (refer to ProcessorState <<Enumeration>> on page 113)
ProcessorTemperature	Returns the processor temperature.	Get	*Double*
ProcessorType	Returns the processor type of the hardware.	Get	*String*
SerialNumber	Returns the serial number of the hardware.	Get	*Signed 32 Bit Integer*

Methods The element has no methods.

Returned by The element is returned by properties or methods of the following elements:

- IPmRegisteredDS1006Platform (refer to RegisteredDS1006Platform / IPmRegisteredDS1006Platform <<Interface>> on page 123)

DS1006MemoryInfo / IPmDS1006MemoryInfo <<Interface>>

Description IPmDS1006MemoryInfo Interface (RCP/HIL Device)

Properties The element has the following properties:

Name	Description	Get/Set	Type
FlashEPROMSize	Returns the flash EPROM size of the DS1006 hardware	Get	*Signed 32 Bit Integer*
GlobalRAMSize	Returns the global RAM size of the DS1006 hardware	Get	*Signed 32 Bit Integer*
L2CacheSize	Returns the L2 cache size of the DS1006 hardware	Get	*Signed 32 Bit Integer*

Methods The element has no methods.

Returned by The element is returned by properties or methods of the following elements:

- IPmRegisteredDS1006Platform (refer to RegisteredDS1006Platform / IPmRegisteredDS1006Platform <<Interface>> on page 123)

DS1006RegisterInfo / IPmDS1006RegisterInfo <<Interface>>

Description IPmDS1006RegisterInfo Interface

Properties The element has the following properties:

Name	Description	Get/Set	Type
ConnectionType	Property to specify the connection type that is used for assignment	Get/Set	InterfaceConnectionType (refer to InterfaceConnectionType

Name	Description	Get/Set	Type
			<<Enumeration>> on page 69)
NetClient	Property to specify the net client that is used for assignment	Get/Set	*String*
PortAddress	Gets or sets the port address.	Get/Set	*Signed 32 Bit Integer*
ProcessorName	Property to specify the processor name.	Get/Set	*String*
Type	Returns the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)

DS1007HardwareInformation / IPmDS1007HardwareInformation <<Interface>>

Description

Interface to access the DS1007 hardware information.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
CPU	Gets the CPU type of the processor board hardware.	Get	*String*
FlashDriveSize	Gets the flash drive size of the processor board hardware.	Get	*Signed 64 Bit Integer*
Frequency	Gets the frequency of the processor board hardware.	Get	*Double*
IsMultiCoreHardware	Gets a value indicating whether the hardware is multicore.	Get	*Boolean*
NumberOfAvailableCores	Gets the number of available cores.	Get	*Signed 32 Bit Integer*
ProductVersion	Gets the product version of the processor board hardware.	Get	*String*
RAMSize	Gets the RAM size of the processor board hardware.	Get	*Signed 32 Bit Integer*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1007ProcessingUnit (refer to DS1007ProcessingUnit / IPmDS1007ProcessingUnit <<Interface>> on page 43)

DS1007IdentificationInformation / IPmDS1007IdentificationInformation <<Interface>>

Description

Interface to access the DS1007 identification information.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BoardType	Gets the board type of the processor board hardware.	Get	*String*
IPAddress	Gets the IP address of the processor board hardware.	Get	*String*
MACAddress	Gets the MAC address of the processor board hardware.	Get	*String*
SerialNumber	Gets the serial number of the processor board hardware.	Get	*Signed 32 Bit Integer*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1007ProcessingUnit (refer to DS1007ProcessingUnit / IPmDS1007ProcessingUnit <<Interface>> on page 43)

DS1007ProcessingUnit / IPmDS1007ProcessingUnit <<Interface>>

Description Interface to access the DS1007 computation node.

Properties The element has the following properties:

Name	Description	Get/Set	Type
ApplicationParts	Returns the currently loaded real-time application collection.	Get	IPmApplicationParts (refer to ApplicationParts / IPmApplicationParts <<Collection>> on page 25)
BoardHardware	Gets the hardware information object.	Get	IPmDS1007HardwareInformation (refer to DS1007HardwareInformation / IPmDS1007HardwareInformation <<Interface>> on page 41)
ConnectionState	Gets the current connection state to indicate whether the device is connected. Not visible in the property grid, visualized only by different icons.	Get	ConnectionState (refer to ConnectionState <<Enumeration>> on page 34)
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*
Identification	Gets the identification information object.	Get	IPmDS1007IdentificationInformation (refer to DS1007IdentificationInformation / IPmDS1007IdentificationInformation <<Interface>> on page 42)
IOPlatforms	Gets the connected I/O platforms.	Get	IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)
Properties	Provides access to the supported property collection of the pu properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
Software	Gets the software information object.	Get	IPmDS1007SoftwareInformation (refer to DS1007SoftwareInformation / IPmDS1007SoftwareInformation <<Interface>> on page 45)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Name	Description	Get/Set	Type
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods The element has no methods.

DS1007RegistrationInfo / IPmDS1007RegistrationInfo <<Interface>>

Description Interface to access the DS1007 registration information.

Properties The element has the following properties:

Name	Description	Get/Set	Type
Name	Property to specify the the custom platform, processing unit or processor board name.	Get/Set	*String*
RegistrationInfos	Gets the registration information of the platform's member processing units.	Get	IPmProcessingUnitRegisterInfos (refer to ProcessingUnitRegisterInfos / IPmProcessingUnitRegisterInfos <<Collection>> on page 110)
Type	Returns the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Methods The element has no methods.

Returned by The element is returned by properties or methods of the following elements:

- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)

DS1007SoftwareInformation / IPmDS1007SoftwareInformation <<Interface>>

Description

Interface to access the DS1007 software information.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
FirmwareVersion	Gets the version of the firmware.	Get	*String*
FPGACoreVersion	Gets the version of the core FPGA.	Get	*String*
FPGAVersion	Gets the version of the FPGA.	Get	*String*
LastFirmwareUpdate	Gets the date of the last firmware update as a string.	Get	*String*
LastFPGAUpdate	Gets the date of the last FPGA update as a string.	Get	*String*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1007ProcessingUnit (refer to DS1007ProcessingUnit / IPmDS1007ProcessingUnit <<Interface>> on page 43)

DS1104BoardDetails / IPmDS1104BoardDetails <<Interface>>

Description

IPmDS1104BoardDetails Interface (RCP/HIL Device)

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BoardVersion	Returns the board version of the hardware.	Get	*String*
BusFrequency	Returns the bus frequency of the hardware.	Get	*Double*
ProcessorFrequency	Returns the processor frequency of the hardware.	Get	*Double*
ProcessorState	Returns the processor state of the hardware.	Get	ProcessorState (refer to ProcessorState <<Enumeration>> on page 113)
ProcessorType	Returns the processor type of the hardware.	Get	*String*
SerialNumber	Returns the serial number of the hardware.	Get	*Signed 32 Bit Integer*

Name	Description	Get/Set	Type
SlaveProcessorState	Returns the slave processor state of the hardware.	Get	ProcessorState (refer to ProcessorState <<Enumeration>> on page 113)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmRegisteredDS1104Platform (refer to RegisteredDS1104Platform / IPmRegisteredDS1104Platform <<Interface>> on page 128)

DS1104MemoryInfo / IPmDS1104MemoryInfo <<Interface>>

Description

IPmDS1104MemoryInfo Interface (RCP/HIL Device)

Properties

The element has the following properties:

Name	Description	Get/Set	Type
FlashEPROMSize	Returns the flash EPROM size of the DS1104 hardware	Get	*Signed 32 Bit Integer*
GlobalRAMSize	Returns the global RAM size of the DS1104 hardware	Get	*Signed 32 Bit Integer*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmRegisteredDS1104Platform (refer to RegisteredDS1104Platform / IPmRegisteredDS1104Platform <<Interface>> on page 128)

DS1202HardwareInformation / IPmDS1202HardwareInformation <<Interface>>

Description

Interface to access the DS1202 hardware information.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
CPU	Gets the CPU type of the processor board hardware.	Get	*String*
FlashDriveSize	Gets the flash drive size of the processor board hardware.	Get	*Signed 64 Bit Integer*
Frequency	Gets the frequency of the processor board hardware.	Get	*Double*
IsMultiCoreHardware	Gets a value indicating whether the hardware is multicore.	Get	*Boolean*
NumberOfAvailableCores	Gets the number of available cores.	Get	*Signed 32 Bit Integer*
ProductVersion	Gets the product version of the processor board hardware.	Get	*String*
RAMSize	Gets the RAM size of the processor board hardware.	Get	*Signed 32 Bit Integer*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1202ProcessingUnit (refer to DS1202ProcessingUnit / IPmDS1202ProcessingUnit <<Interface>> on page 48)

DS1202IdentificationInformation / IPmDS1202IdentificationInformation <<Interface>>

Description

Interface to access the DS1202 identification information.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BoardType	Gets the board type of the processor board hardware.	Get	*String*
IPAddress	Gets the IP address of the processor board hardware.	Get	*String*
MACAddress	Gets the MAC address of the processor board hardware.	Get	*String*
SerialNumber	Gets the serial number of the processor board hardware.	Get	*Signed 32 Bit Integer*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1202ProcessingUnit (refer to DS1202ProcessingUnit / IPmDS1202ProcessingUnit <<Interface>> on page 48)

DS1202ProcessingUnit / IPmDS1202ProcessingUnit <<Interface>>

Description

Interface to access the DS1202 processing unit.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
ApplicationParts	Returns the currently loaded real-time application collection.	Get	IPmApplicationParts (refer to ApplicationParts / IPmApplicationParts <<Collection>> on page 25)
BoardHardware	Gets the hardware information object.	Get	IPmDS1202HardwareInformation (refer to DS1202HardwareInformation / IPmDS1202HardwareInformation <<Interface>> on page 47)

Name	Description	Get/Set	Type
ConnectionState	Gets the current connection state to indicate whether the device is connected. Not visible in the property grid, visualized only by different icons.	Get	ConnectionState (refer to ConnectionState <<Enumeration>> on page 34)
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*
Identification	Gets the identification information object.	Get	IPmDS1202IdentificationInformation (refer to DS1202IdentificationInformation / IPmDS1202IdentificationInformation <<Interface>> on page 48)
IOPlatforms	Gets the connected I/O platforms.	Get	IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)
Properties	Provides access to the supported property collection of the pu properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
Software	Gets the software information object.	Get	IPmDS1202SoftwareInformation (refer to DS1202SoftwareInformation / IPmDS1202SoftwareInformation <<Interface>> on page 50)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods

The element has no methods.

DS1202RegistrationInfo / IPmDS1202RegistrationInfo <<Interface>>

Description Interface to access the DS1202 registration information.

Properties The element has the following properties:

Name	Description	Get/Set	Type
Name	Property to specify the the custom platform, processing unit or processor board name.	Get/Set	*String*
RegistrationInfos	Gets the registration information of the platform's member processing units.	Get	IPmProcessingUnitRegisterInfos (refer to ProcessingUnitRegisterInfos / IPmProcessingUnitRegisterInfos <<Collection>> on page 110)
Type	Returns the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Methods The element has no methods.

DS1202SoftwareInformation / IPmDS1202SoftwareInformation <<Interface>>

Description Interface to access the DS1202 software information.

Properties The element has the following properties:

Name	Description	Get/Set	Type
FirmwareVersion	Gets the version of the firmware.	Get	*String*
FPGACoreVersion	Gets the version of the core FPGA.	Get	*String*
FPGAVersion	Gets the version of the FPGA.	Get	*String*
LastFirmwareUpdate	Gets the date of the last firmware update as a string.	Get	*String*
LastFPGAUpdate	Gets the date of the last FPGA update as a string.	Get	*String*

Methods The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1202ProcessingUnit (refer to DS1202ProcessingUnit / IPmDS1202ProcessingUnit <<Interface>> on page 48)

DS1403HardwareInformation / IPmDS1403HardwareInformation <<Interface>>

Description

Interface to access the DS1403 hardware information.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
CPU	Gets the CPU type of the ds1403 hardware.	Get	*String*
FlashDriveSize	Gets the flash drive size of the ds1403 hardware.	Get	*Signed 64 Bit Integer*
Frequency	Gets the frequency of the ds1403 hardware.	Get	*Double*
IsMultiCoreHardware	Gets a value indicating whether the hardware is multicore.	Get	*Boolean*
NumberOfAvailableCores	Gets the number of available cores.	Get	*Signed 32 Bit Integer*
ProductVersion	Gets the product version of the ds1403 hardware.	Get	*String*
RAMSize64	Gets the RAM size of the ds1403 hardware.	Get	*Signed 64 Bit Integer*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1403ProcessingUnit (refer to DS1403ProcessingUnit / IPmDS1403ProcessingUnit <<Interface>> on page 52)

DS1403IdentificationInformation / IPmDS1403IdentificationInformation <<Interface>>

Description

IPmds1403IdentificationInformation interface (RCP/HIL device).

Properties The element has the following properties:

Name	Description	Get/Set	Type
BoardName	Gets the board name of the ds1403 hardware.	Get	*String*
DSNumber	Gets the DS number of the ds1403 hardware.	Get	*String*
Identifier	Gets the serial number of the ds1403 hardware.	Get	*Signed 32 Bit Integer*
IPAddress	Gets the IP address of the ds1403 hardware.	Get	*String*
MACAddress	Gets the MAC address of the ds1403 hardware.	Get	*String*

Methods The element has no methods.

Returned by The element is returned by properties or methods of the following elements:

- IPmDS1403ProcessingUnit (refer to DS1403ProcessingUnit / IPmDS1403ProcessingUnit <<Interface>> on page 52)

DS1403ProcessingUnit / IPmDS1403ProcessingUnit <<Interface>>

Description IPmds1403ProcessingUnitInternal Interface.

Properties The element has the following properties:

Name	Description	Get/Set	Type
ApplicationParts	Returns the currently loaded real-time application parts collection.	Get	IPmApplicationParts (refer to ApplicationParts / IPmApplicationParts <<Collection>> on page 25)
BoardHardware	Gets the hardware information object.	Get	IPmDS1403HardwareInformation (refer to DS1403HardwareInformation / IPmDS1403HardwareInformation <<Interface>> on page 51)
BoardName	Gets the board name.	Get	*String*
ConnectionState	Gets the current connection state to indicate whether the device is connected. Not visible in the property grid, visualized only by different icons.	Get	ConnectionState (refer to ConnectionState <<Enumeration>> on page 34)

Name	Description	Get/Set	Type
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*
EthernetAdapters	Returns the ethernet adapter collection of the ethernet adapters.	Get	IPmEthernetAdapters (refer to EthernetAdapters / IPmEthernetAdapters <<Collection>> on page 61)
EthernetSwitches	Returns a collection including the ethernet switches of processing unit's IO platforms.	Get	IPmEthernetSwitches (refer to EthernetSwitches / IPmEthernetSwitches <<Collection>> on page 63)
HostInterface	Gets the host interface information object.	Get	IPmHostInterfaceInformation (refer to HostInterfaceInformation / IPmHostInterfaceInformation <<Interface>> on page 67)
Identification	Gets the identification information object.	Get	IPmDS1403IdentificationInformation (refer to DS1403IdentificationInformation / IPmDS1403IdentificationInformation <<Interface>> on page 51)
IOPlatforms	Gets the connected I/O platforms.	Get	IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)
Properties	Provides access to the supported property collection of the pu properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
Software	Gets the software information object.	Get	IPmDS1403SoftwareInformation (refer to DS1403SoftwareInformation / IPmDS1403SoftwareInformation <<Interface>> on page 55)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods The element has the following methods:

Name	Description	Parameter[1)]	Returns
SetBoardName	Sets the board name.	▪ *<String>* `Name`: Board name to set.	None

1) <Type> `Name`: Description

DS1403RegistrationInfo / IPmDS1403RegistrationInfo <<Interface>>

Description IPmDS1403RegistrationInfo Interface

Properties The element has the following properties:

Name	Description	Get/Set	Type
Name	Property to specify the the custom platform, processing unit or processor board name.	Get/Set	*String*
RegistrationInfos	Registration infos of member processing units of platform.	Get	IPmProcessingUnitRegisterInfos (refer to ProcessingUnitRegisterInfos / IPmProcessingUnitRegisterInfos <<Collection>> on page 110)
Type	Returns the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Methods The element has no methods.

DS1403SoftwareInformation / IPmDS1403SoftwareInformation <<Interface>>

Description

IPmDS1403SoftwareInformation interface (RCP/HIL device).

Properties

The element has the following properties:

Name	Description	Get/Set	Type
FirmwareVersion	Gets the version of the firmware.	Get	*String*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1403ProcessingUnit (refer to DS1403ProcessingUnit / IPmDS1403ProcessingUnit <<Interface>> on page 52)

DS2301RegisterInfo / IPmDS2301RegisterInfo <<Interface>>

Description

IPmDS2301RegisterInfo Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
ConnectionType	Property to specify the connection type that is used for assignment	Get/Set	InterfaceConnectionType (refer to InterfaceConnectionType <<Enumeration>> on page 69)
NetClient	Property to specify the net client that is used for assignment	Get/Set	*String*
PortAddress	Gets or sets the port address.	Get/Set	*Signed 32 Bit Integer*
Type	Returns the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)

DS2302RegisterInfo / IPmDS2302RegisterInfo <<Interface>>

Description

IPmDS2302RegisterInfo Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
ConnectionType	Property to specify the connection type that is used for assignment	Get/Set	InterfaceConnectionType (refer to InterfaceConnectionType <<Enumeration>> on page 69)
NetClient	Property to specify the net client that is used for assignment	Get/Set	*String*
PortAddress	Gets or sets the port address.	Get/Set	*Signed 32 Bit Integer*
Type	Returns the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)

DS230xIOPlatform / IPmDS230xIOPlatform <<Interface>>

Description IPmDS230xIOPlatform Interface (

Properties The element has the following properties:

Name	Description	Get/Set	Type
BoardRevision	Gets the revision of the I/O platform. Returns 0 if the I/O platform does not support the board revision.	Get	*String*
Description	Gets a description of the I/O platform, e.g. PHS-bus address, interrupt, slot, ...	Get	*String*
DisplayName	Gets the name of the I/O platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*
Name	Gets the name of the I/O platform used as a unique identifier.	Get	*String*
PHSBusAddress	Gets the PHS-bus address of the I/O board.	Get	*String*
PortAddress	Gets and sets the port address of the I/O board.	Get/Set	*Signed 32 Bit Integer*
Properties	Provides access to the supported property collection of the I/O platform.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RealTimeApplication	Returns the currently loaded real-time application.	Get	IPmRealTimeApplication (refer to RealTimeApplication / IPmRealTimeApplication <<Interface>> on page 118)
SerialNumber	Gets the serial number of the I/O platform. Returns 0 if the I/O platform does not support the serial number.	Get	*Signed 32 Bit Integer*
Type	Gets the type of the I/O platform.	Get	IOPlatformType (refer to IOPlatformType <<Enumeration>> on page 74)

Methods The element has the following methods:

Name	Description	Parameter[1)]	Returns
GetRealTimeApplication	Returns the real-time application loaded on the specified channel.	▪ *<Signed 32 Bit Integer>* `Channel`: The parameter Channel.	System.String ▪ *String*

Name	Description	Parameter[1)]	Returns
LoadRealtimeApplication	Loads the real-time application specified by ApplicationFullPath. If the real-time application is already loaded, this method reloads it. The real-time application is also loaded if another real-time application is currently running.	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None
ModuleDescription	Returns a description of the specified channel.	▪ *<Signed 32 Bit Integer>* `Channel`: The parameter Channel.	System.String ▪ *String*
StopRTP	Stops the real-time processor. If the real-time processor was already stopped, the method just returns without any exception.	None	None

1) <Type> `Name`: Description

DS4505IOPlatform / IPmDS4505IOPlatform <<Interface>>

Description IPmDS4505IOPlatform Interface (

Properties The element has the following properties:

Name	Description	Get/Set	Type
BoardRevision	Gets the revision of the I/O platform. Returns 0 if the I/O platform does not support the board revision.	Get	*String*
Description	Gets a description of the I/O platform, e.g. PHS-bus address, interrupt, slot, ...	Get	*String*
DisplayName	Gets the name of the I/O platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*
Name	Gets the name of the I/O platform used as a unique identifier.	Get	*String*
PHSBusAddress	Gets the PHS-bus address of the I/O board.	Get	*String*
Properties	Provides access to the supported property collection of the I/O platform.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)

Name	Description	Get/Set	Type
SerialNumber	Gets the serial number of the I/O platform. Returns 0 if the I/O platform does not support the serial number.	Get	*Signed 32 Bit Integer*
Type	Gets the type of the I/O platform.	Get	IOPlatformType (refer to IOPlatformType <<Enumeration>> on page 74)

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
ModuleDescription	Returns a description of the specified channel.	▪ *<Signed 32 Bit Integer>* `Channel`: The parameter Channel.	System.String ▪ *String*

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)

E

Where to go from here

Information in this section

EmbeddedIOPlatform / IPmEmbeddedIOPlatform <<Interface>>

Description

IPmEmbeddedIOPlatform Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
IOModules	Gets the connected I/O modules.	Get	IPmIOModules (refer to IOModules / IPmIOModules <<Collection>> on page 71)
Properties	Provides access to the supported property collection of the I/O platform.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
Type	Gets the type of the I/O platform.	Get	IOPlatformType (refer to IOPlatformType <<Enumeration>> on page 74)

Methods

The element has no methods.

EthernetAdapter / IPmEthernetAdapter <<Interface>>

Description

Interface for accessing an Ethernet adapter.

Interface representing an Ethernet adapter.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
ID	Gets the ID.	Get	*String*
MACAddress	Gets the MAC address.	Get	*String*
Name	Gets or sets the name.	Get/Set	*String*
Owner	Gets the owner processing unit if it's an onboard ethernet adapter. Otherwise gets the owner IO platform.	Get	*Object*
Properties	Gets the property collection of the Ethernet adapter.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmEthernetAdapters (refer to EthernetAdapters / IPmEthernetAdapters <<Collection>> on page 61)

EthernetAdapters / IPmEthernetAdapters <<Collection>>

Description

Interface representing a collection of Ethernet adapters.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Gets the number of Ethernet adapters in the collection.	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Checks whether an Ethernet adapter with a specified index exists in the collection.	▪ *<Object>* `Index`: Index or object for accessing an Ethernet adapter.	True if the Ethernet adapter exists. ▪ *Boolean*
Item	Returns the Ethernet adapter from the collection specified by the index. The index can be the zero-based integer index.	▪ *<Object>* `Index`: Index of the Ethernet adapter in the collection.	Returns the requested Ethernet adapter if it exists. Otherwise, returns null. ▪ IPmEthernetAdapter (refer to EthernetAdapter / IPmEthernetAdapter <<Interface>> on page 61)

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1403ProcessingUnit (refer to DS1403ProcessingUnit / IPmDS1403ProcessingUnit <<Interface>> on page 52)
- IPmSCALEXIOProcessingUnit (refer to SCALEXIOProcessingUnit / IPmSCALEXIOProcessingUnit <<Interface>> on page 148)

EthernetProtocol <<Enumeration>>

Description

EthernetProtocol enumeration type.

Enumeration values

The enumeration has the following values:

Name	Description	Value
UDP	Ethernet protocol is UDP	0
TCP	Ethernet protocol is TCP	1

EthernetSwitch / IPmEthernetSwitch <<Interface>>

Description

Interface representing an ethernet switch.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Name	Gets or sets the name of the ethernet switch.	Get/Set	*String*
Owner	Gets the IO platform the ethernet switch belongs to.	Get	IPmIOPlatform (refer to IOPlatform / IPmIOPlatform <<Interface>> on page 72)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmEthernetSwitches (refer to EthernetSwitches / IPmEthernetSwitches <<Collection>> on page 63)

EthernetSwitches / IPmEthernetSwitches <<Collection>>

Description

Interface representing a collection of ethernet switches.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Gets the number of ethernet switches in the collection.	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Checks whether an athernet switch with the specified index exists in the collection.	▪ *<Object>* `Index`: The index of the requested ethernet switch.	True if an ethernet siwtch with the passed index exists. False otherwise. ▪ *Boolean*

Name	Description	Parameter[1)]	Returns
Item	Returns the ethernet switch from the collection specified by the index. The index can be a zero-based integer.	▪ *<Object>* `Index`: The index of the ethernet switch in the collection.	Returns the requested ethernet switch if it exists. Otherwise, returns null. ▪ IPmEthernetSwitch (refer to EthernetSwitch / IPmEthernetSwitch <<Interface>> on page 63)

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1403ProcessingUnit (refer to DS1403ProcessingUnit / IPmDS1403ProcessingUnit <<Interface>> on page 52)
- IPmSCALEXIOProcessingUnit (refer to SCALEXIOProcessingUnit / IPmSCALEXIOProcessingUnit <<Interface>> on page 148)

ExperimentPlatformsCollection / IPmExperimentPlatformsCollection <<Interface>>

Description

This interface is to access the platform smart extension.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of platforms in the collection.	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Checks whether a platform exists in the platforms collection. The index can be the zero-based integer index or the name of the platform.	▪ *<Object>* `Index`: Index or name of platform.	Returns True if the platform exists. Otherwise, returns False. ▪ *Boolean*
Item	Returns the platform from the collection specified by the index. The index can be the zero-based integer index or the name of the platform.	▪ *<Object>* `Index`: Index or name of platform.	Returns the requested platform, if it exists. Otherwise, returns null. ▪ *dSPACE.PlatformManagement.Automation.IPmDS1006Platform*

Name	Description	Parameter[1)]	Returns
			▪ *dSPACE.PlatformManagement.Automation.IPmDS1007Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmDS1104Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmMABXPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmMultiProcessorPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmSCALEXIOPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmVEOSPlatform* ▪ IPmRegisteredDS1006Platform (refer to RegisteredDS1006Platform / IPmRegisteredDS1006Platform <<Interface>> on page 123) ▪ IPmRegisteredDS1104Platform (refer to RegisteredDS1104Platform / IPmRegisteredDS1104Platform <<Interface>> on page 128) ▪ IPmRegisteredMultiProcessorPlatform (refer to RegisteredMultiProcessorPlatform / IPmRegisteredMultiProcessorPlatform <<Interface>> on page 136) ▪ IPmRegisteredSCALEXIOPlatform (refer to RegisteredSCALEXIOPlatform / IPmRegisteredSCALEXIOPlatform <<Interface>> on page 138) ▪ IPmRegisteredVEOSPlatform (refer to RegisteredVEOSPlatform / IPmRegisteredVEOSPlatform <<Interface>> on page 141)
Rename	Renames an existing platform/device. Index can be the zero-based integer index from platform	▪ <*Object*> `Index`: Index or name of platform.	None

Name	Description	Parameter[1)]	Returns
	management or the name of the platform.	▪ *<String>* `NewPlatformName`: New name of platform.	

1) <Type> `Name`: Description

H

HostInterfaceInformation / IPmHostInterfaceInformation <<Interface>>

Description

IPmHostInterfaceInformation interface.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
IPAddress	Gets the IP address of the hardware board's host interface.	Get	*String*
IPMode	Gets the IP mode of the hardware board's host interface.	Get	*String*
MACAddress	Gets the MAC address of the hardware board's host interface.	Get	*String*
SubnetMask	Gets the subnet mask of the hardware board's host interface.	Get	*String*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1403ProcessingUnit (refer to DS1403ProcessingUnit / IPmDS1403ProcessingUnit <<Interface>> on page 52)

Where to go from here

Information in this section

InitialPageType <<Enumeration>>

Description

InitialPageType enumeration type.

Enumeration values

The enumeration has the following values:

Name	Description	Value
ECUDefined	Initial page type is ECU-defined	0
WorkingPage	Initial page type is working page	1

Name	Description	Value
ReferencePage	Initial page type is reference page	2
ToolDefined	Initial page type is tool-defined	3

Returned by

The element is returned by properties or methods of the following elements:

- IPmCalibrationPlatformGeneralSettings (refer to CalibrationPlatformGeneralSettings / IPmCalibrationPlatformGeneralSettings <<Interface>> on page 30)

InterfaceConnectionType <<Enumeration>>

Description

Connection Type Enumeration

Enumeration values

The enumeration has the following values:

Name	Description	Value
Bus	Bus connection	0
Net	Net connection	1

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1006RegisterInfo (refer to DS1006RegisterInfo / IPmDS1006RegisterInfo <<Interface>> on page 40)
- IPmDS2301RegisterInfo (refer to DS2301RegisterInfo / IPmDS2301RegisterInfo <<Interface>> on page 55)
- IPmDS2302RegisterInfo (refer to DS2302RegisterInfo / IPmDS2302RegisterInfo <<Interface>> on page 56)
- IPmMultiprocessorRegisterInfo (refer to MultiprocessorRegisterInfo / IPmMultiprocessorRegisterInfo <<Interface>> on page 84)
- IPmRecentHardwareItem (refer to RecentHardwareItem / IPmRecentHardwareItem <<Interface>> on page 120)
- IPmRegisteredDS1006Platform (refer to RegisteredDS1006Platform / IPmRegisteredDS1006Platform <<Interface>> on page 123)
- IPmRegisteredMABXPlatform (refer to RegisteredMABXPlatform / IPmRegisteredMABXPlatform <<Interface>> on page 134)

InventoryInformation / IPmInventoryInformation <<Interface>>

Description

IPmInventoryInformation Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
ModuleVersion	Returns the module version.	Get	*String*
XMLInventoryInformation	Returns the version information as an XML formatted string.	Get	*String*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmRegisteredDS1006Platform (refer to RegisteredDS1006Platform / IPmRegisteredDS1006Platform <<Interface>> on page 123)
- IPmRegisteredDS1104Platform (refer to RegisteredDS1104Platform / IPmRegisteredDS1104Platform <<Interface>> on page 128)
- IPmRegisteredMABXPlatform (refer to RegisteredMABXPlatform / IPmRegisteredMABXPlatform <<Interface>> on page 134)

IOModule / IPmIOModule <<Interface>>

Description

IPmIOModule Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Properties	Provides access to the supported property collection of the I/O platform.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)

Methods

The element has no methods.

IOModuleOwningPlatform / IPmIOModuleOwningPlatform <<Interface>>

Description IPmIOModuleOwningPlatform Interface

Properties The element has the following properties:

Name	Description	Get/Set	Type
IOModules	Gets the connected I/O modules.	Get	IPmIOModules (refer to IOModules / IPmIOModules <<Collection>> on page 71)

Methods The element has no methods.

IOModules / IPmIOModules <<Collection>>

Description IPmIOModules Interface

Properties The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of I/O modules in the collection. [0..System.Int32 -1]	Get	*Signed 32 Bit Integer*

Methods The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Checks whether an I/O module with the specified name is in the collection.	▪ *<String>* `IOModuleName`: IO module name	True if the I/O module exists. ▪ *Boolean*
Item	Returns the item specified by index. The index can be an integer or a string value.	▪ *<String>* `Index`: An integer index or string with I/O module name.	None

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmEmbeddedIOPlatform (refer to EmbeddedIOPlatform / IPmEmbeddedIOPlatform <<Interface>> on page 60)
- IPmIOModuleOwningPlatform (refer to IOModuleOwningPlatform / IPmIOModuleOwningPlatform <<Interface>> on page 71)

IOPlatform / IPmIOPlatform <<Interface>>

Description

IPmIOPlatform Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BoardRevision	Gets the revision of the I/O platform. Returns 0 if the I/O platform does not support the board revision.	Get	*String*
Description	Gets a description of the I/O platform, e.g. PHS-bus address, interrupt, slot, ...	Get	*String*
DisplayName	Gets the name of the I/O platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*
Name	Gets the name of the I/O platform used as a unique identifier.	Get	*String*
Properties	Provides access to the supported property collection of the I/O platform.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
SerialNumber	Gets the serial number of the I/O platform. Returns 0 if the I/O platform does not support the serial number.	Get	*Signed 32 Bit Integer*
Type	Gets the type of the I/O platform.	Get	IOPlatformType (refer to IOPlatformType <<Enumeration>> on page 74)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmEthernetSwitch (refer to EthernetSwitch / IPmEthernetSwitch <<Interface>> on page 63)

- IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)

IOPlatforms / IPmIOPlatforms <<Collection>>

Description

IPmIOPlatforms Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of I/O platforms in the collection. [0..System.Int32 -1]	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Checks whether an I/O platform with the specified name is in the collection.	▪ *<String>* `IOPlatformName`: IO platform name	True if the I/O platform exists. ▪ *Boolean*
Item	Returns the item specified by index. The index can be an integer or a string value.	▪ *<String>* `Index`: An integer index or string with I/O platform name.	The item specified by index. ▪ *dSPACE.PlatformManagement.Automation.IPmRapidProIOPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmSCALEXIOPlatform* ▪ IPmDS4505IOPlatform (refer to DS4505IOPlatform / IPmDS4505IOPlatform <<Interface>> on page 58) ▪ IPmIOPlatform (refer to IOPlatform / IPmIOPlatform <<Interface>> on page 72) ▪ *IPmIOPlatforms* ▪ IPmPHSIOPlatform (refer to PHSIOPlatform / IPmPHSIOPlatform <<Interface>> on page 89)

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1007ProcessingUnit (refer to DS1007ProcessingUnit / IPmDS1007ProcessingUnit <<Interface>> on page 43)
- IPmDS1202ProcessingUnit (refer to DS1202ProcessingUnit / IPmDS1202ProcessingUnit <<Interface>> on page 48)
- IPmDS1403ProcessingUnit (refer to DS1403ProcessingUnit / IPmDS1403ProcessingUnit <<Interface>> on page 52)
- IPmIOUnit (refer to IOUnit / IPmIOUnit <<Interface>> on page 78)
- IPmRegisteredDS1006Platform (refer to RegisteredDS1006Platform / IPmRegisteredDS1006Platform <<Interface>> on page 123)
- IPmRegisteredDS1007Platform (refer to RegisteredDS1007Platform / IPmRegisteredDS1007Platform <<Interface>> on page 126)
- IPmRegisteredDS1202Platform (refer to RegisteredDS1202Platform / IPmRegisteredDS1202Platform <<Interface>> on page 130)
- IPmRegisteredDS1403Platform (refer to RegisteredDS1403Platform / IPmRegisteredDS1403Platform <<Interface>> on page 132)
- IPmRegisteredMABXPlatform (refer to RegisteredMABXPlatform / IPmRegisteredMABXPlatform <<Interface>> on page 134)
- IPmRegisteredSCALEXIOPlatform (refer to RegisteredSCALEXIOPlatform / IPmRegisteredSCALEXIOPlatform <<Interface>> on page 138)
- IPmSCALEXIOProcessingUnit (refer to SCALEXIOProcessingUnit / IPmSCALEXIOProcessingUnit <<Interface>> on page 148)

IOPlatformType <<Enumeration>>

Description

IOPlatformType enumeration

Enumeration values

The enumeration has the following values:

Name	Description	Value
Unknown	Enumeration value for unknown board type.	0
DS2001	Enumeration value DS2001 of enumeration type IOPlatformType.	1
DS2002	Enumeration value DS2002 of enumeration type IOPlatformType.	2
DS2003	Enumeration value DS2003 of enumeration type IOPlatformType.	3
DS2004	Enumeration value DS2004 of enumeration type IOPlatformType.	4
DS2101	Enumeration value DS2101 of enumeration type IOPlatformType.	5

Name	Description	Value
DS2102	Enumeration value DS2102 of enumeration type IOPlatformType.	6
DS2103	Enumeration value DS2103 of enumeration type IOPlatformType.	7
DS2201	Enumeration value DS2201 of enumeration type IOPlatformType.	8
DS2202	Enumeration value DS2202 of enumeration type IOPlatformType.	9
DS2210	Enumeration value DS2210 of enumeration type IOPlatformType.	10
DS2211	Enumeration value DS2211 of enumeration type IOPlatformType.	11
DS2301	Enumeration value DS2301 of enumeration type IOPlatformType.	12
DS2302	Enumeration value DS2302 of enumeration type IOPlatformType.	13
DS2401	Enumeration value DS2401 of enumeration type IOPlatformType.	14
DS3001	Enumeration value DS3001 of enumeration type IOPlatformType.	15
DS3002	Enumeration value DS3002 of enumeration type IOPlatformType.	16
DS4001	Enumeration value DS4001 of enumeration type IOPlatformType.	17
DS4002	Enumeration value DS4002 of enumeration type IOPlatformType.	18
DS4003	Enumeration value DS4003 of enumeration type IOPlatformType.	19
DS4101	Enumeration value DS4101 of enumeration type IOPlatformType.	20
DS4110	Enumeration value DS4110 of enumeration type IOPlatformType.	21
DS4120	Enumeration value DS4120 of enumeration type IOPlatformType.	22
DS4121	Enumeration value DS4121 of enumeration type IOPlatformType.	23
DS4201	Enumeration value DS4201 of enumeration type IOPlatformType.	24
DS4301	Enumeration value DS4301 of enumeration type IOPlatformType.	25
DS4302	Enumeration value DS4302 of enumeration type IOPlatformType.	26
DS4330	Enumeration value DS4330 of enumeration type IOPlatformType.	27
DS4401	Enumeration value DS4401 of enumeration type IOPlatformType.	28

Name	Description	Value
DS4501	Enumeration value DS4501 of enumeration type IOPlatformType.	29
DS4502	Enumeration value DS4502 of enumeration type IOPlatformType.	30
DS4503	Enumeration value DS4503 of enumeration type IOPlatformType.	31
DS5001	Enumeration value DS5001 of enumeration type IOPlatformType.	32
DS5101	Enumeration value DS5101 of enumeration type IOPlatformType.	33
DS5201	Enumeration value DS5201 of enumeration type IOPlatformType.	34
DS5202	Enumeration value DS5202 of enumeration type IOPlatformType.	35
DS4504	Enumeration value DS4504 of enumeration type IOPlatformType.	36
DS4505	Enumeration value DS4505 of enumeration type IOPlatformType.	37
DS4201S	Enumeration value DS4201S of enumeration type IOPlatformType.	38
DS2601	Enumeration value DS2601 of enumeration type IOPlatformType.	39
DS2621	Enumeration value DS2621 of enumeration type IOPlatformType.	40
DS2642	Enumeration value DS2642 of enumeration type IOPlatformType.	41
DS2671	Enumeration value DS2671 of enumeration type IOPlatformType.	42
DS2901	Enumeration value DS2901 of enumeration type IOPlatformType.	43
DS5203	Enumeration value DS5203 of enumeration type IOPlatformType.	44
DS4004	Enumeration value DS4004 of enumeration type IOPlatformType.	45
DS802	Enumeration value DS802 of enumeration type IOPlatformType.	46
DS2551	Enumeration value DS2551 of enumeration type IOPlatformType.	47
DS2502	Enumeration value DS2502 of enumeration type IOPlatformType.	48
DS2672	Enumeration value DS2672 of enumeration type IOPlatformType.	49
DS2680	Enumeration value DS2680 of enumeration type IOPlatformType.	50
DS2680RT	Enumeration value DS2680RT of enumeration type IOPlatformType.	51

Name	Description	Value
DS2907	Enumeration value DS2907 of enumeration type IOPlatformType.	52
DS2690	Enumeration value DS2690 of enumeration type IOPlatformType.	53
DS2702	Enumeration value DS2702 of enumeration type IOPlatformType.	54
DS2703	Enumeration value DS2703 of enumeration type IOPlatformType.	55
DS2655	Enumeration value DS2655 of enumeration type IOPlatformType.	56
DS2908	Enumeration value DS2908 of enumeration type IOPlatformType.	57
DS2656	Enumeration value DS2656 of enumeration type IOPlatformType.	58
DS6301	Enumeration value DS6301 of enumeration type IOPlatformType.	59
DS6072	Enumeration value DS6072 of the IOPlatformType enumeration type.	60
DS6331	Enumeration value DS6331 of the IOPlatformType enumeration type.	61
DS6332	Enumeration value DS6332 of the IOPlatformType enumeration type.	62
DS6341	Enumeration value DS6341 of the IOPlatformType enumeration type.	63
DS6351	Enumeration value DS6351 of the IOPlatformType enumeration type.	64
DS6411	Enumeration value DS6411 of the IOPlatformType enumeration type.	65
DS6221	Enumeration value DS6221 of the IOPlatformType enumeration type.	66
DS6101	Enumeration value DS6101 of the IOPlatformType enumeration type.	67
DS6201	Enumeration value DS6201 of the IOPlatformType enumeration type.	68
DS6202	Enumeration value DS6202 of the IOPlatformType enumeration type.	69
DS6311	Enumeration value DS6311 of the IOPlatformType enumeration type.	70
DS6071	Enumeration value DS6071 of the IOPlatformType enumeration type.	71
DS6241	Enumeration value DS6241 of the IOPlatformType enumeration type.	72
DS6333	Enumeration value DS6333 of enumeration type IOPlatformType.	73
DS6334	Enumeration value DS6334 of enumeration type IOPlatformType.	74

Name	Description	Value
DS6335	Enumeration value DS6335 of enumeration type IOPlatformType.	75
DS6601	Enumeration value DS6601 of enumeration type IOPlatformType.	76
DS6602	Enumeration value DS6602 of enumeration type IOPlatformType.	77
DS6121	Enumeration value DS6121 of enumeration type IOPlatformType.	78
DS6321	Enumeration value DS6321 of enumeration type IOPlatformType.	79
DS1511	Enumeration value DS1511 of enumeration type IOPlatformType.	80
DS1513	Enumeration value DS1513 of enumeration type IOPlatformType.	81
DS1514	Enumeration value DS1514 of enumeration type IOPlatformType.	82
DS1521	Enumeration value DS1521 of enumeration type IOPlatformType.	83
DS6342	Enumeration value DS6342 of enumeration type IOPlatformType.	84
DS6651	Enumeration value DS6651 of enumeration type IOPlatformType.	85
DS6336	Enumeration value DS6336 of the IOPlatformType enumeration type.	86

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS230xIOPlatform (refer to DS230xIOPlatform / IPmDS230xIOPlatform <<Interface>> on page 57)
- IPmDS4505IOPlatform (refer to DS4505IOPlatform / IPmDS4505IOPlatform <<Interface>> on page 58)
- IPmEmbeddedIOPlatform (refer to EmbeddedIOPlatform / IPmEmbeddedIOPlatform <<Interface>> on page 60)
- IPmIOPlatform (refer to IOPlatform / IPmIOPlatform <<Interface>> on page 72)
- IPmPHSIOPlatform (refer to PHSIOPlatform / IPmPHSIOPlatform <<Interface>> on page 89)

IOUnit / IPmIOUnit <<Interface>>

Description

Interface to access a IO unit.

Interface representing I/O units of smart platforms.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
IOPlatforms	Gets the connected I/O platforms.	Get	IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)
Name	Gets the name of the unit.	Get	*String*
Properties	Provides access to the supported property collection of the platform properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RackName	Gets the name of the rack the hardware element belongs to.	Get	*String*
Type	Gets the type of the unit.	Get	*String*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
SetName	Sets the name of the unit.	▪ <*String*> `Name`: Unit name to set.	None
SetRackName	Sets the name of the rack the hardware element belongs to.	▪ <*String*> `Name`: Rack name to set.	None

1) <Type> `Name`: Description

M

Where to go from here

Information in this section

MABXBoardDetails / IPmMABXBoardDetails <<Interface>>

Description

IPmMABXBoardDetails Interface (RCP/HIL Device)

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BatteryVoltage	Gets the battery voltage of the MicroAutoBox hardware.	Get	*Double*
BoardTemperature	Gets the board temperature of the MicroAutoBox hardware.	Get	*Double*
BoardVersion	Returns the board version of the hardware.	Get	*String*
BusFrequency	Returns the bus frequency of the hardware.	Get	*Double*
PortAddress	Property to get/set the port address	Get	*Signed 32 Bit Integer*
ProcessorFrequency	Returns the processor frequency of the hardware.	Get	*Double*
ProcessorState	Returns the processor state of the hardware.	Get	ProcessorState (refer to ProcessorState

Name	Description	Get/Set	Type
			<<Enumeration>> on page 113)
ProcessorTemperature	Gets the processor temperature of the MicroAutoBox hardware.	Get	*Double*
ProcessorType	Returns the processor type of the hardware.	Get	*String*
SerialNumber	Returns the serial number of the hardware.	Get	*Signed 32 Bit Integer*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmRegisteredMABXPlatform (refer to RegisteredMABXPlatform / IPmRegisteredMABXPlatform <<Interface>> on page 134)

MABXMemoryInfo / IPmMABXMemoryInfo <<Interface>>

Description

IPmMABXMemoryInfo Interface (RCP/HIL Device)

Properties

The element has the following properties:

Name	Description	Get/Set	Type
FlashEPROMSize	Gets the size of the flash EPROM of the MABX hardware	Get	*Signed 32 Bit Integer*
GlobalRAMSize	Gets the global RAM size of the MABX hardware	Get	*Signed 32 Bit Integer*
LocalRAMSize	Gets the local RAM size of the MABX hardware	Get	*Signed 32 Bit Integer*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmRegisteredMABXPlatform (refer to RegisteredMABXPlatform / IPmRegisteredMABXPlatform <<Interface>> on page 134)

MABXRegisterInfo / IPmMABXRegisterInfo <<Interface>>

Description

IPmMABXRegisterInfo Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
NetClient	Property to specify the net client that is used for assignment	Get/Set	*String*
Type	Returns the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)

MeasurementServiceType <<Enumeration>>

Description

Measurement service types enumeration.

Enumeration values

The enumeration has the following values:

Name	Description	Value
CLIB_SERVICE	Host service	0
DS_DAQ_SERVICE	dsDAQ service	1

MeasurementState <<Enumeration>>

Description MeasurementState State Enumeration

Enumeration values The enumeration has the following values:

Name	Description	Value
Stopped	Measurement is stopped. Initial state.	0
Running	Measurement is running.	1

MemorySegmentType <<Enumeration>>

Description Enumeration describing possible memory segment content types

Enumeration values The enumeration has the following values:

Name	Description	Value
DATA	Memory segment content type	0
CODE	Memory segment content type	1
OFFLINE_DATA	Memory segment content type	2
VARIABLES	Memory segment content type	3
CALIBRATION_VARIABLES	Memory segment content type	4
SERAM	Memory segment content type	5
EXCLUDE_FROM_FLASH	Memory segment content type	6
RESERVED	Memory segment content type	7

MultiprocessorRegisterInfo / IPmMultiprocessorRegisterInfo <<Interface>>

Description

IPmMultiprocessorRegisterInfo Interface (RCP/HIL Device).

Properties

The element has the following properties:

Name	Description	Get/Set	Type
ConnectionType	Property to specify the connection type that is used for assignment	Get/Set	InterfaceConnectionType (refer to InterfaceConnectionType <<Enumeration>> on page 69)
Name	Property to specify the the custom platform, processing unit or processor board name.	Get/Set	*String*
NetClient	Property to specify the net client that is used for assignment	Get/Set	*String*
RegisterInfos	Gets the RegisterInfos.	Get	*Object*
TopologyCheck	Gets or sets a value indicating whether to specify the topology check.	Get/Set	*Boolean*
Type	Returns the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Add	This is the Add method of the IPmMultiprocessorRegisterInfo interface.	▪ <PlatformType (refer to PlatformType <<Enumeration>> on page 107)> `PlatformType`: The parameter PlatformType.	Return value of the method. ▪ *Object*
AddWithCustomName	This is the AddWithCustomName method of the IPmMultiprocessorRegisterInfo interface.	▪ <PlatformType (refer to PlatformType <<Enumeration>> on page 107)> `PlatformType`: The parameter PlatformType. ▪ <*String*> `ProcessorName`: The parameter ProcessorName.	Return value of the method. ▪ *Object*

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)

O

OnlineCalibrationBehavior <<Enumeration>>

Description

OnlineCalibrationBehavior type enumeration.

Enumeration values

The enumeration has the following values:

Name	Description	Value
PromptUser	Always prompt user if differences are detected.	0
UploadWorkingPageDownloadReferencePage	Upload working page and download reference page if differences are detected.	1
DownloadWorkingPageUploadReferencePage	Download the working page and upload the reference page if differences are detected.	2
DownloadWorkingPageDownloadReferencePage	Download the working page and the reference page if differences are detected.	3
UploadWorkingPageUploadReferencePage	Upload working page and upload reference page if differences are detected.	4
UploadConnectedVariables	Upload connected variables only.	5
IgnoreDifferences	Ignores all difference and skips balancing. Calibration is disabled.	6
Upload	Uploads all variables or memory segments	7
Download	Downloads all variables or memory segments	8
DownloadConnectedVariables	Downloads connected variables only.	9

Returned by

The element is returned by properties or methods of the following elements:

- IPmCalibrationPlatformGeneralSettings (refer to CalibrationPlatformGeneralSettings / IPmCalibrationPlatformGeneralSettings <<Interface>> on page 30)

P

Where to go from here

Information in this section

PageAccessType <<Enumeration>>

Description Enumeration of platform page access kinds

Enumeration values The enumeration has the following values:

Name	Description	Value
ReadOnly	Platform page is 'read only'	0
WriteOnly	Platform page is 'write only'	1
ReadWrite	Platform page is 'read/write'	2

PageConcept <<Enumeration>>

Description

PageConcept type enumeration.

Enumeration describes the page concept of the platform.

Enumeration values

The enumeration has the following values:

Name	Description	Value
None	Platform has no page concept.	0
ReferencePage	Platform has a reference page only.	1
WorkingPage	Platform has a working page only.	2
ReferencePage_WorkingPage	Platform has a working page only.	3

PageType <<Enumeration>>

Description

Enumeration of platform page types

Enumeration values

The enumeration has the following values:

Name	Description	Value
ReferencePage	Reference page	0
WorkingPage	Working page	1

PHSIOPlatform / IPmPHSIOPlatform <<Interface>>

Description

IPmPHSIOPlatform Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BoardRevision	Gets the revision of the I/O platform. Returns 0 if the I/O platform does not support the board revision.	Get	*String*
Description	Gets a description of the I/O platform, e.g. PHS-bus address, interrupt, slot, ...	Get	*String*

Name	Description	Get/Set	Type
DisplayName	Gets the name of the I/O platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*
Name	Gets the name of the I/O platform used as a unique identifier.	Get	*String*
PHSBusAddress	Gets the PHS-bus address of the I/O board.	Get	*String*
Properties	Provides access to the supported property collection of the I/O platform.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
SerialNumber	Gets the serial number of the I/O platform. Returns 0 if the I/O platform does not support the serial number.	Get	*Signed 32 Bit Integer*
Type	Gets the type of the I/O platform.	Get	IOPlatformType (refer to IOPlatformType <<Enumeration>> on page 74)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)

PlatformCalibrationState <<Enumeration>>

Description

PlatformCalibrationState Type Enumeration

Enumeration values

The enumeration has the following values:

Name	Description	Value
Stopped	Online calibration is stopped. Initial state.	0
Started	Online calibration is started.	1

PlatformManagement / IPmPlatformManagement <<Interface>>

Description

IPmPlatformManagement Interface

Provides access to platforms and platform-specific objects

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BalanceConnectedParametersOnly	Enable or disable balancing of connected parameters only. Specifies parameter balancing outside of memory segments. Balancing of connected parameters only is enabled if value is true, otherwise balancing is done for all parameters outside of memory segments.	Get/Set	*Boolean*
DisplayPlatformMessageDialogs	Enable or disable displaying of platform message dialogs. Specifies whether to display platform message dialogs for asynchronous messages. Asynchronous messages can arise from a running real-time application or they appear when a platform is unplugged.	Get/Set	*Boolean*
IsPlatformSearchFinished	Returns True if the search has finished. Otherwise, returns False. An event is also available.	Get	*Boolean*
PlatformAutomationAPIVersion	Selects the platform interface API version provided by platform management API version 1.0 interfaces are deprecated but provided for script compatibility.	Get/Set	AutomationAPIVersion (refer to AutomationAPIVersion <<Enumeration>> on page 29)
Platforms	Returns the platforms collection. The platforms collection is never null, but can be empty.	Get	IPmSeekedPlatforms (refer to SeekedPlatforms / IPmSeekedPlatforms <<Collection>> on page 151)
RecentPlatformConfiguration	Gets the collection of recent platforms.	Get	IPmRecentPlatformConfiguration (refer to RecentPlatformConfiguration / IPmRecentPlatformConfiguration <<Collection>> on page 121)
ResynchronizationEnabled	Enables or disables resynchronization of measurement signals on all platforms. If resynchronization is enabled, all hardware time stamps are normalized to host time stamps. If True, resynchronization is enabled. Otherwise, resynchronization is disabled.	Get/Set	*Boolean*
ResynchronizationRate	Gets or sets the resynchronization rate. Value is interpreted as milliseconds	Get/Set	*Signed 32 Bit Integer*

Name	Description	Get/Set	Type
SeekNetworkPlatformsOnAutomatedStartup	Enables or disables searching for network platforms on automated startup. Specifies searching for network platforms. Searching is enabled if value is True, Network platforms are ignored otherwise.	Get/Set	*Boolean*
SeekNetworkPlatformsOnStartup	Enables or disables searching for network platforms on startup. Specifies searching for network platforms. Searching is enabled if value is True, Network platforms are ignored otherwise.	Get/Set	*Boolean*
SeekPlatformsOnAutomatedStartup	Enables or disables searching for platforms on automated startup. Specifies searching for platforms. Searching is enabled if value is True, Network platforms are ignored otherwise.	Get/Set	*Boolean*
SeekPlatformsOnStartup	Enable or disable searching for platforms on startup. Specifies searching for platforms. Searching is enabled if value is True, Network platforms are ignored otherwise.	Get/Set	*Boolean*
VariableObserverEnabled	Enables or disables Variable Observer on all platforms. If True, Variable Observer is enabled. Otherwise, Variable Observer is disabled.	Get/Set	*Boolean*
VariableObserverRate	Gets or sets the VariableObserver update rate.	Get/Set	VariableObserverRates (refer to VariableObserverRates <<Enumeration>> on page 159)

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
ClearSystem	This method clears the recent platforms collection, deletes all platforms and clears the device driver.	▪ *<Boolean>* `ForceDriverReset`: If this parameter is True, a driver reset is forced even if other applications are connected to the driver. If it is False, an exception is thrown when other applications are connected to the driver.	None
CreatePlatformRegistrationInfo	This method creates a platform registration information object.	▪ <PlatformType (refer to PlatformType <<Enumeration>> on page 107)> `PlatformType`: The parameter PlatformType.	A platform registration information object. ▪ IPmDS1006RegisterInfo (refer to DS1006RegisterInfo /

Name	Description	Parameter[1)]	Returns
			IPmDS1006RegisterInfo <<Interface>> on page 40) ▪ IPmDS1007RegistrationInfo (refer to DS1007RegistrationInfo / IPmDS1007RegistrationInfo <<Interface>> on page 44) ▪ IPmDS2301RegisterInfo (refer to DS2301RegisterInfo / IPmDS2301RegisterInfo <<Interface>> on page 55) ▪ IPmDS2302RegisterInfo (refer to DS2302RegisterInfo / IPmDS2302RegisterInfo <<Interface>> on page 56) ▪ IPmMABXRegisterInfo (refer to MABXRegisterInfo / IPmMABXRegisterInfo <<Interface>> on page 82) ▪ IPmMultiprocessorRegisterInfo (refer to MultiprocessorRegisterInfo / IPmMultiprocessorRegisterInfo <<Interface>> on page 84) ▪ IPmSCALEXIORegisterInfo (refer to SCALEXIORegisterInfo / IPmSCALEXIORegisterInfo <<Interface>> on page 149) ▪ IPmVEOSRegisterInfo (refer to VEOSRegisterInfo / IPmVEOSRegisterInfo <<Interface>> on page 165)
CreateSupportInfoFile	This method creates a support information file.	None	None
RefreshInterfaceConnections	This method deletes all platforms, clears the device driver and registers platforms again using the platform configuration.	▪ *<Boolean>* `ForceDriverReset`: If this parameter is True, a driver reset is forced even if other applications are connected to the driver. If it is False, an exception is thrown when other applications are connected to the driver.	None
RefreshPlatformConfiguration	This method initiates all platforms to check for hardware according to their current configuration parameters. Connected platforms can be disconnected,	None	None

Name	Description	Parameter[1)]	Returns
	disconnected platforms can be connected, or the connection state is unchanged.		
RegisterPlatform	This method registers a platform.	▪ <IPmDS1006RegisterInfo (refer to DS1006RegisterInfo / IPmDS1006RegisterInfo <<Interface>> on page 40)> `RegisterInfo`: A platform registration information object.	The registered platform. ▪ *dSPACE.PlatformManagement.Automation.IPmDS1006Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmDS1007Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmDS2301Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmDS2302Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmMABXPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmMultiprocessorPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmSCALEXIOPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmVEOSPlatform* ▪ IPmRegisteredDS1006Platform (refer to RegisteredDS1006Platform / IPmRegisteredDS1006Platform <<Interface>> on page 123) ▪ IPmRegisteredDS1104Platform (refer to RegisteredDS1104Platform / IPmRegisteredDS1104Platform <<Interface>> on page 128) ▪ IPmRegisteredMultiProcessorPlatform (refer to RegisteredMultiProcessorPlatform / IPmRegisteredMultiProcessorPlatform <<Interface>> on page 136) ▪ IPmRegisteredSCALEXIOPlatform (refer to RegisteredSCALEXIOPlatform / IPmRegisteredSCALEXIOPlatf

Name	Description	Parameter[1)]	Returns
			orm <<Interface>> on page 138) ▪ IPmRegisteredVEOSPlatform (refer to RegisteredVEOSPlatform / IPmRegisteredVEOSPlatform <<Interface>> on page 141)

1) <Type> `Name`: Description

Event Interfaces

The element provides the following event interfaces:

- IPmPlatformManagementEvents (refer to PlatformManagementEvents / IPmPlatformManagementEvents <<EventInterface>> on page 95)

PlatformManagementEvents / IPmPlatformManagementEvents <<EventInterface>>

Description

The platform management events.

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
InterfaceConnectionRefreshed	The interface connection has been refreshed.	None	None
InterfaceConnectionRefreshing	Refreshes the interface connection.	None	None
PlatformAdded	The specified platform was added.	▪ *<Object>* `Platform`: The platform which was added.	None
PlatformConnected	The specified platform has changed state to connected.	▪ *<Object>* `Platform`: The platform which has changed to connected state.	None
PlatformDisconnected	The specified platform has changed state to disconnected.	▪ *<Object>* `Platform`: The platform which has changed to disconnected state.	None
PlatformPlugged	The specified platform has changed state to plugged.	▪ *<Object>* `Platform`: The platform which has changed to plugged state.	None
PlatformRemoving	Removes the specified platform.	▪ *<Object>* `Platform`: The platform to be removed.	None
PlatformSearchFinished	The platform search has finished.	None	None

Name	Description	Parameter[1)]	Returns
PlatformUnplugged	The specified platform has changed state to unplugged.	▪ *<Object>* `Platform`: The platform which has changed to unplugged state.	None
RealTimeApplicationLoaded	The specified platform has loaded a real-time application.	▪ *<Object>* `Platform`: The platform which has loaded a real-time application.	None
RealTimeApplicationPaused	The specified platform paused the real-time application.	▪ *<Object>* `Platform`: Platform that paused the real-time application.	None
RealTimeApplicationStarted	The specified platform has started the real-time application.	▪ *<Object>* `Platform`: The platform which started the real-time application.	None
RealTimeApplicationStopped	The specified platform has stopped the real-time application.	▪ *<Object>* `Platform`: The platform which has stopped the real-time application.	None
RealTimeApplicationUnloaded	The specified platform unloaded a real-time application.	▪ *<Object>* `Platform`: Platform that unloaded a real-time application.	None

1) <Type> `Name`: Description

Provided by

The element is provided by following event sources:

- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)

PlatformNames / IPmPlatformNames <<Collection>>

Description

IPmSeekedPlatforms Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of platform names in the collection.	Get	*Signed 32 Bit Integer*

Methods The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Checks whether a platform name exists in the platform name collection. The index can be the zero-based integer index or the name of the platform.	▪ *<Object>* `Index`: Index or name of platform.	Returns True if the platform name exists. Otherwise, returns False. ▪ *Boolean*
Item	Returns the platform name from the collection specified by the index. The index is the zero-based integer index.	▪ *<Signed 32 Bit Integer>* `Index`: Index of platform name.	Returns the requested platform name, if it exists. Otherwise, returns null. ▪ *String*

1) <Type> `Name`: Description

Returned by The element is returned by properties or methods of the following elements:

- IPmSeekedPlatforms (refer to SeekedPlatforms / IPmSeekedPlatforms <<Collection>> on page 151)

PlatformProcessorNames / IPmPlatformProcessorNames <<Collection>>

Description IPmProcessorNamesInternal Interface

Properties The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of processorboards in the collection. [0..System.Int32 -1]	Get	*Signed 32 Bit Integer*

Methods The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Returns True if an element that is accessible by the specified string is in the collection.	▪ *<Object>* `Index`: Index to access a processorboard	True if processorboard name exists. ▪ *Boolean*
Item	Returns a processorboard from the collection.	▪ *<Object>* `Index`: System.Object; either integer index or string with processorboards name	Return value of the method. ▪ IPmProcessorNames (refer to ProcessorNames / IPmProcessorNames

Name	Description	Parameter[1)]	Returns
			<<Collection>> on page 112)

[1)] <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmRegisteredDS1007Platform (refer to RegisteredDS1007Platform / IPmRegisteredDS1007Platform <<Interface>> on page 126)

PlatformProposedCalibrationState <<Enumeration>>

Description

PlatformProposedCalibrationState Type Enumeration

Enumeration values

The enumeration has the following values:

Name	Description	Value
Stopped	Proposed calibration is stopped. Initial state.	0
Started	Proposed calibration is started. All read and write operations are performed to mirrored memory, until proposed calibration is applied or stopped.	1

Platforms / IPmPlatforms <<Collection>>

Description

Interface to access the platforms.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of platforms in the collection.	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Add	Creates a new platform of specified type and adds the	▪ <PlatformType (refer to PlatformType <<Enumeration>> on	The new platform object.

Name	Description	Parameter[1)]	Returns
	new platform to the platforms collection.	page 107)> `PlatformType`: The type of the new platform.	▪ *dSPACE.PlatformManagement.Automation.IPmCANDataOutputPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmCANMonitoringPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmCCPPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmDS1006Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmDS1104Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmECUDiagnostics2Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmECUDiagnosticsPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmFlexRayMonitoringPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmGSI2Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmGSIPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmIMCPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmIPEPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmLINMonitoringPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmMABXPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmMultiProcessorPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmRapidProPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmSCALEXIOPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmXCPonCANPlatform*

Name	Description	Parameter[1)]	Returns
			▪ *dSPACE.PlatformManagement.Automation.IPmXCPonEthernetPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmXCPonFlexRayPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmXCPonUSBPlatform*
AddExistingPlatform	Adds an existing platform to this platforms collection, e.g., a project-global platform, to the active experiment. For example, it adds a platform that was searched for to the active experiment. Index can be the zero-based integer index from platform management or the name of the platform.	▪ <*Object*> `Index`: Index or name of platform.	Returns True if adding an existing platform is successful. Otherwise, returns False. ▪ *Boolean*
AddExistingWithCustomName	Adds an existing platform to this platforms collection and sets the specified custom name, e.g. adds a found platform to the active experiment and give it a special name. Index can be the zero-based integer index from platform management or the name of the platform.	▪ <*Object*> `Index`: Index or name of platform. ▪ <*String*> `PlatformName`: The name of the platform.	Returns True if adding an existing platform is successful. Otherwise, returns False. ▪ *Boolean*
AddProjectPlatform	Adds an existing platform to this platforms collection, e.g., a project-global platform, to the active experiment. For example, it adds a platform that was searched for to the active experiment. Index can be the zero-based integer index from project or the name of the platform.	▪ <*Object*> `Index`: Index or name of platform.	Returns True if adding an existing platform is successful. Otherwise, returns False. ▪ *Boolean*
AddWithCustomName	Creates a new platform of specified type and adds the new platform to the platforms collection.	▪ <PlatformType (refer to PlatformType <<Enumeration>> on page 107)> `PlatformType`: The type of the new platform. ▪ <*String*> `PlatformName`: The name of the new platform.	The new platform object. ▪ *dSPACE.PlatformManagement.Automation.IPmCANDataOutputPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmCANMonitoringPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmCCPPlatform*

Name	Description	Parameter[1)]	Returns
			▪ *dSPACE.PlatformManagement.Automation.IPmDS1006Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmDS1104Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmECUDiagnostics2Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmECUDiagnosticsPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmFlexRayMonitoringPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmGMEPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmGSI2Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmGSIPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmIMCPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmIPEPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmLINMonitoringPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmMABXPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmMultiProcessorPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmRapidProPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmSCALEXIOPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmXCPonCANPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmXCPonEthernetPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmXCPonFlexRayPlatform*

Name	Description	Parameter[1)]	Returns
			▪ *dSPACE.PlatformManagement.Automation.IPmXCPonUSBPlatform*
Contains	Checks whether a platform exists in the platforms collection. The index can be the zero-based integer index or the name of the platform.	▪ <*Object*> `Index`: Index or name of platform.	Returns True if the platform exists. Otherwise, returns False. ▪ *Boolean*
Item	Returns the platform from the collection specified by the index. The index can be the zero-based integer index or the name of the platform.	▪ <*Object*> `Index`: Index or name of platform.	Returns the requested platform, if it exists. Otherwise, returns null. ▪ *dSPACE.PlatformManagement.Automation.IPmCANDataOutputPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmCANMonitoringPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmCCPPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmDS1006Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmDS1104Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmECUDiagnostics2Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmECUDiagnosticsPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmFlexRayMonitoringPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmGSI2Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmGSIPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmIMCPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmIPEPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmLINMonitoringPlatform*

Name	Description	Parameter[1)]	Returns
			▪ *dSPACE.PlatformManagement.Automation.IPmMABXPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmMultiProcessorPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmRapidProPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmSCALEXIOPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmXCPonCANPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmXCPonEthernetPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmXCPonFlexRayPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmXCPonUSBPlatform*
Remove	Removes a platform from the collection. The index can be the zero-based integer index or the name of the platform.	▪ *<Object>* `Index`: The index or the name of the platform to remove.	None
RemoveAll	Removes all platforms from the collection.	None	Returns True if successful. Otherwise, returns False. ▪ *Boolean*
Rename	Renames an existing platform/device. Index can be the zero-based integer index from platform management or the name of the platform.	▪ *<Object>* `Index`: Index or name of platform. ▪ *<String>* `NewPlatformName`: New name of platform.	None

[1)] <Type> `Name`: Description

PlatformsCollection / IPmPlatformsCollection <<Collection>>

Description IPmPlatforms Interface

Properties The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of platforms in the collection.	Get	*Signed 32 Bit Integer*

Methods The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Checks whether a platform exists in the platforms collection. The index can be the zero-based integer index or the name of the platform.	▪ <*Object*> `Index`: Index or name of platform.	Returns True if the platform exists. Otherwise, returns False. ▪ *Boolean*
Item	Returns the platform from the collection specified by the index. The index can be the zero-based integer index or the name of the platform.	▪ <*Object*> `Index`: Index or name of platform.	Returns the requested platform, if it exists. Otherwise, returns null. ▪ *dSPACE.PlatformManagement.Automation.IPmDS1006Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmDS1007Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmDS1104Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmMABXPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmMultiProcessorPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmSCALEXIOPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmVEOSPlatform* ▪ IPmRegisteredDS1006Platform (refer to RegisteredDS1006Platform / IPmRegisteredDS1006Platfor

Name	Description	Parameter[1)]	Returns
			m <<Interface>> on page 123) ▪ IPmRegisteredDS1104Platform (refer to RegisteredDS1104Platform / IPmRegisteredDS1104Platform <<Interface>> on page 128) ▪ IPmRegisteredMultiProcessorPlatform (refer to RegisteredMultiProcessorPlatform / IPmRegisteredMultiProcessorPlatform <<Interface>> on page 136) ▪ IPmRegisteredSCALEXIOPlatform (refer to RegisteredSCALEXIOPlatform / IPmRegisteredSCALEXIOPlatform <<Interface>> on page 138) ▪ IPmRegisteredVEOSPlatform (refer to RegisteredVEOSPlatform / IPmRegisteredVEOSPlatform <<Interface>> on page 141)

1) <Type> **Name**: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmRegisteredDS1007Platform (refer to RegisteredDS1007Platform / IPmRegisteredDS1007Platform <<Interface>> on page 126)
- IPmRegisteredDS1202Platform (refer to RegisteredDS1202Platform / IPmRegisteredDS1202Platform <<Interface>> on page 130)
- IPmRegisteredDS1403Platform (refer to RegisteredDS1403Platform / IPmRegisteredDS1403Platform <<Interface>> on page 132)
- IPmRegisteredMultiProcessorPlatform (refer to RegisteredMultiProcessorPlatform / IPmRegisteredMultiProcessorPlatform <<Interface>> on page 136)
- IPmRegisteredSCALEXIOPlatform (refer to RegisteredSCALEXIOPlatform / IPmRegisteredSCALEXIOPlatform <<Interface>> on page 138)
- IPmRegisteredVEOSPlatform (refer to RegisteredVEOSPlatform / IPmRegisteredVEOSPlatform <<Interface>> on page 141)

PlatformsCollectionExtension / IPmPlatformsCollectionExtension <<Interface>>

Description IPmPlatformsCollectionExtension interface.

Properties The element has no properties.

Methods The element has the following methods:

Name	Description	Parameter[1)]	Returns
Rename	This is the Rename method of the IPmPlatformsCollectionExtension interface.	▪ *<Object>* `Index`: The parameter Index. ▪ *<String>* `NewPlatformName`: The parameter NewPlatformName.	None

1) <Type> `Name`: Description

PlatformSeekers / IPmPlatformSeekers <<Interface>>

Description IPmPlatformSeekers Interface

Properties The element has the following properties:

Name	Description	Get/Set	Type
DS1005	Not implemented.	Get	*Object*
DS1006	Not implemented.	Get	*Object*
DS1104	Not implemented.	Get	*Object*
DS2510	Not implemented.	Get	*Object*
MABX	Not implemented.	Get	*Object*
MultiProcessor	Not implemented.	Get	*Object*
RapidPro	Not implemented.	Get	*Object*

Methods The element has no methods.

PlatformType <<Enumeration>>

Description

PlatformType enumeration type.

Enumeration values

The enumeration has the following values:

Name	Description	Value
MABX	MicroAutoBox platform.	0
DS1005	DS1005 platform.	3
XCPonCAN	XCPonCAN platform.	4
CCP	CCP platform.	6
GSI	GSI platform.	7
CANMonitoring	CANMonitoring platform.	10
Diagnostic	Diagnostics platform.	14
DS1103	DS1103 platform.	15
DS1104	DS1104 platform.	16
DS1006	DS1006 platform	17
MultiProcessor	MultiProcessor platform.	18
LINMonitoring	LINMonitoring platform.	19
XCPonEthernet	XCPonEthernet platform.	20
XCPonFlexRay	XCPonFlexRay platform.	21
SCALEXIO	SCALEXIO platform.	22
FlexRayMonitoring	FlexRayMonitoring Platform.	24
GSI2	GSI2 platform.	25
VEOS	VEOS platform.	26
Diagnostic2	Diagnostics2 platform.	27
VideoCapturing	Video Capturing platform.	28
DS1007	DS1007 platform.	29
DS1202	DS1202 platform.	30
XILAPIMAPort	XILAPIMAPort platform.	31
EthernetMonitoring	EthernetMonitoring platform.	32
DS1403	For future use.	33
GNSS	GPS Device.	34

Returned by

The element is returned by properties or methods of the following elements:

- IPmApplicationProcess (refer to ApplicationProcess / IPmApplicationProcess <<Interface>> on page 26)
- IPmDS1006RegisterInfo (refer to DS1006RegisterInfo / IPmDS1006RegisterInfo <<Interface>> on page 40)

- IPmDS1007ProcessingUnit (refer to DS1007ProcessingUnit / IPmDS1007ProcessingUnit <<Interface>> on page 43)
- IPmDS1007RegistrationInfo (refer to DS1007RegistrationInfo / IPmDS1007RegistrationInfo <<Interface>> on page 44)
- IPmDS1202ProcessingUnit (refer to DS1202ProcessingUnit / IPmDS1202ProcessingUnit <<Interface>> on page 48)
- IPmDS1202RegistrationInfo (refer to DS1202RegistrationInfo / IPmDS1202RegistrationInfo <<Interface>> on page 50)
- IPmDS1403ProcessingUnit (refer to DS1403ProcessingUnit / IPmDS1403ProcessingUnit <<Interface>> on page 52)
- IPmDS1403RegistrationInfo (refer to DS1403RegistrationInfo / IPmDS1403RegistrationInfo <<Interface>> on page 54)
- IPmDS2301RegisterInfo (refer to DS2301RegisterInfo / IPmDS2301RegisterInfo <<Interface>> on page 55)
- IPmDS2302RegisterInfo (refer to DS2302RegisterInfo / IPmDS2302RegisterInfo <<Interface>> on page 56)
- IPmMABXRegisterInfo (refer to MABXRegisterInfo / IPmMABXRegisterInfo <<Interface>> on page 82)
- IPmMultiprocessorRegisterInfo (refer to MultiprocessorRegisterInfo / IPmMultiprocessorRegisterInfo <<Interface>> on page 84)
- IPmProcessingUnitRegisterInfo (refer to ProcessingUnitRegisterInfo / IPmProcessingUnitRegisterInfo <<Interface>> on page 109)
- IPmRegisteredDS1006Platform (refer to RegisteredDS1006Platform / IPmRegisteredDS1006Platform <<Interface>> on page 123)
- IPmRegisteredDS1007Platform (refer to RegisteredDS1007Platform / IPmRegisteredDS1007Platform <<Interface>> on page 126)
- IPmRegisteredDS1104Platform (refer to RegisteredDS1104Platform / IPmRegisteredDS1104Platform <<Interface>> on page 128)
- IPmRegisteredDS1202Platform (refer to RegisteredDS1202Platform / IPmRegisteredDS1202Platform <<Interface>> on page 130)
- IPmRegisteredDS1403Platform (refer to RegisteredDS1403Platform / IPmRegisteredDS1403Platform <<Interface>> on page 132)
- IPmRegisteredMABXPlatform (refer to RegisteredMABXPlatform / IPmRegisteredMABXPlatform <<Interface>> on page 134)
- IPmRegisteredMultiProcessorPlatform (refer to RegisteredMultiProcessorPlatform / IPmRegisteredMultiProcessorPlatform <<Interface>> on page 136)
- IPmRegisteredSCALEXIOPlatform (refer to RegisteredSCALEXIOPlatform / IPmRegisteredSCALEXIOPlatform <<Interface>> on page 138)
- IPmRegisteredVEOSPlatform (refer to RegisteredVEOSPlatform / IPmRegisteredVEOSPlatform <<Interface>> on page 141)
- IPmRegisteredXILAPIMAPortPlatform (refer to RegisteredXILAPIMAPortPlatform / IPmRegisteredXILAPIMAPortPlatform <<Interface>> on page 143)
- IPmSCALEXIOProcessingUnit (refer to SCALEXIOProcessingUnit / IPmSCALEXIOProcessingUnit <<Interface>> on page 148)

- IPmSCALEXIORegisterInfo (refer to SCALEXIORegisterInfo / IPmSCALEXIORegisterInfo <<Interface>> on page 149)
- IPmSCALEXIORegistrationInfo (refer to SCALEXIORegistrationInfo / IPmSCALEXIORegistrationInfo <<Interface>> on page 150)
- IPmSubstitutePlatform (refer to SubstitutePlatform / IPmSubstitutePlatform <<Interface>> on page 153)
- IPmVEOSProcessingUnit (refer to VEOSProcessingUnit / IPmVEOSProcessingUnit <<Interface>> on page 164)
- IPmVEOSRegisterInfo (refer to VEOSRegisterInfo / IPmVEOSRegisterInfo <<Interface>> on page 165)
- IPmXILAPIMAPortRegisterInfo (refer to XILAPIMAPortRegisterInfo / IPmXILAPIMAPortRegisterInfo <<Interface>> on page 170)

PlugState <<Enumeration>>

Description

PlugState enumeration

Status: Proposed

Enumeration values

The enumeration has the following values:

Name	Description	Value
Unplugged	Platform is unplugged	0
Plugged	Platform is plugged	1

Returned by

The element is returned by properties or methods of the following elements:

- IPmRegisteredMABXPlatform (refer to RegisteredMABXPlatform / IPmRegisteredMABXPlatform <<Interface>> on page 134)

ProcessingUnitRegisterInfo / IPmProcessingUnitRegisterInfo <<Interface>>

Description

IPmProcessingUnitRegisterInfo Interface

Interface representing registration infos of the members (processing units) of smart multiprocessor platforms.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
AliasName	Property to specify the serial number that is used for assignment	Get/Set	*String*
BoardName	Property to specify the serial number that is used for assignment	Get/Set	*String*
IPAddress	Gets or sets the IP address.	Get/Set	*String*
MACAddress	Gets or sets the MAC address.	Get/Set	*String*
Name	Property to specify the the custom platform, processing unit or processor board name.	Get/Set	*String*
Type	Returns the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmProcessingUnitRegisterInfos (refer to ProcessingUnitRegisterInfos / IPmProcessingUnitRegisterInfos <<Collection>> on page 110)

ProcessingUnitRegisterInfos / IPmProcessingUnitRegisterInfos <<Collection>>

Description

IPmProcessingUnitRegisterInfos Interface

Interface representing a collection of registration infos containing the members of smart multiprocessor platforms.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of members in the collection.	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Add	Creates a new registration info, adds it to the collection and returns it.	None	The new registration info. ▪ IPmProcessingUnitRegisterInfo (refer to ProcessingUnitRegisterInfo / IPmProcessingUnitRegisterInfo <<Interface>> on page 109)
Item	Returns the registration info from the collection specified by the index. The index can be the zero-based integer index.	▪ *<Signed 32 Bit Integer>* `Index`: Index of registration info in collection.	Returns the requested registration info if it exists. Otherwise, returns null. ▪ IPmProcessingUnitRegisterInfo (refer to ProcessingUnitRegisterInfo / IPmProcessingUnitRegisterInfo <<Interface>> on page 109)

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1007RegistrationInfo (refer to DS1007RegistrationInfo / IPmDS1007RegistrationInfo <<Interface>> on page 44)
- IPmDS1202RegistrationInfo (refer to DS1202RegistrationInfo / IPmDS1202RegistrationInfo <<Interface>> on page 50)
- IPmDS1403RegistrationInfo (refer to DS1403RegistrationInfo / IPmDS1403RegistrationInfo <<Interface>> on page 54)
- IPmSCALEXIORegistrationInfo (refer to SCALEXIORegistrationInfo / IPmSCALEXIORegistrationInfo <<Interface>> on page 150)

ProcessorName / IPmProcessorName <<Interface>>

Description

IPmProcessorName Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Name	Property to specify the processor name	Get/Set	*String*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmProcessorNames (refer to ProcessorNames / IPmProcessorNames <<Collection>> on page 112)

ProcessorNames / IPmProcessorNames <<Collection>>

Description

IPmPlatformProcessorNamesInternal Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of processorboards in the collection. [0..System.Int32 -1]	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Returns True if an element that is accessible by the specified string is in the collection.	▪ *<Object>* `Index`: Index to access a processorname	True if processorboard name exists. ▪ *Boolean*
Item	Returns a processorboard from the collection.	▪ *<Object>* `Index`: System.Object; either integer index or string with processorboards name	Return value of the method. ▪ IPmProcessorName (refer to ProcessorName / IPmProcessorName <<Interface>> on page 111)

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmPlatformProcessorNames (refer to PlatformProcessorNames / IPmPlatformProcessorNames <<Collection>> on page 97)

ProcessorState <<Enumeration>>

Description

Enumeration for ProcessorState

Enumeration values

The enumeration has the following values:

Name	Description	Value
Undefined	Processor state is undefined	0
Running	Processor state is running	1
Reset	Processor state is reset	2
DetectionLoop	Hardware checkloop is running	3
RunningFromFlash	Application is running from flash	4

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS1006BoardDetails (refer to DS1006BoardDetails / IPmDS1006BoardDetails <<Interface>> on page 39)
- IPmDS1104BoardDetails (refer to DS1104BoardDetails / IPmDS1104BoardDetails <<Interface>> on page 45)
- IPmMABXBoardDetails (refer to MABXBoardDetails / IPmMABXBoardDetails <<Interface>> on page 80)

Properties / IPmProperties <<Collection>>

Description

Interface providing access to a collection of properties.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of the properties in the collection.	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Indicates whether the collection contains a property with the specified index or name.	▪ *<String>* `Index`: Name of the property or a numeric index value to check whether the collection	True if the collection contains a property with the specified index or name, otherwise false. ▪ *Boolean*

Name	Description	Parameter[1)]	Returns
		contains a matching property.	
Item	Returns a property according to the specified index or name.	▪ *<String>* `Index`: Name of the property or a numeric index value.	The property object matching to the specified index or name. ▪ IPmProperty (refer to Property / IPmProperty <<Interface>> on page 115)

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmApplicationPart (refer to ApplicationPart / IPmApplicationPart <<Interface>> on page 24)
- IPmControllableRealTimeApplication (refer to ControllableRealTimeApplication / IPmControllableRealTimeApplication <<Interface>> on page 35)
- IPmDS1007ProcessingUnit (refer to DS1007ProcessingUnit / IPmDS1007ProcessingUnit <<Interface>> on page 43)
- IPmDS1202ProcessingUnit (refer to DS1202ProcessingUnit / IPmDS1202ProcessingUnit <<Interface>> on page 48)
- IPmDS1403ProcessingUnit (refer to DS1403ProcessingUnit / IPmDS1403ProcessingUnit <<Interface>> on page 52)
- IPmDS230xIOPlatform (refer to DS230xIOPlatform / IPmDS230xIOPlatform <<Interface>> on page 57)
- IPmDS4505IOPlatform (refer to DS4505IOPlatform / IPmDS4505IOPlatform <<Interface>> on page 58)
- IPmEmbeddedIOPlatform (refer to EmbeddedIOPlatform / IPmEmbeddedIOPlatform <<Interface>> on page 60)
- IPmEthernetAdapter (refer to EthernetAdapter / IPmEthernetAdapter <<Interface>> on page 61)
- IPmIOModule (refer to IOModule / IPmIOModule <<Interface>> on page 70)
- IPmIOPlatform (refer to IOPlatform / IPmIOPlatform <<Interface>> on page 72)
- IPmIOUnit (refer to IOUnit / IPmIOUnit <<Interface>> on page 78)
- IPmPHSIOPlatform (refer to PHSIOPlatform / IPmPHSIOPlatform <<Interface>> on page 89)
- IPmRealTimeApplication (refer to RealTimeApplication / IPmRealTimeApplication <<Interface>> on page 118)
- IPmRegisteredDS1006Platform (refer to RegisteredDS1006Platform / IPmRegisteredDS1006Platform <<Interface>> on page 123)
- IPmRegisteredDS1007Platform (refer to RegisteredDS1007Platform / IPmRegisteredDS1007Platform <<Interface>> on page 126)
- IPmRegisteredDS1104Platform (refer to RegisteredDS1104Platform / IPmRegisteredDS1104Platform <<Interface>> on page 128)
- IPmRegisteredDS1202Platform (refer to RegisteredDS1202Platform / IPmRegisteredDS1202Platform <<Interface>> on page 130)

- IPmRegisteredDS1403Platform (refer to RegisteredDS1403Platform / IPmRegisteredDS1403Platform <<Interface>> on page 132)
- IPmRegisteredMABXPlatform (refer to RegisteredMABXPlatform / IPmRegisteredMABXPlatform <<Interface>> on page 134)
- IPmRegisteredMultiProcessorPlatform (refer to RegisteredMultiProcessorPlatform / IPmRegisteredMultiProcessorPlatform <<Interface>> on page 136)
- IPmRegisteredSCALEXIOPlatform (refer to RegisteredSCALEXIOPlatform / IPmRegisteredSCALEXIOPlatform <<Interface>> on page 138)
- IPmRegisteredVEOSPlatform (refer to RegisteredVEOSPlatform / IPmRegisteredVEOSPlatform <<Interface>> on page 141)
- IPmSCALEXIOProcessingUnit (refer to SCALEXIOProcessingUnit / IPmSCALEXIOProcessingUnit <<Interface>> on page 148)
- IPmUnit (refer to Unit / IPmUnit <<Interface>> on page 157)
- IPmVEOSApplication (refer to VEOSApplication / IPmVEOSApplication <<Interface>> on page 160)
- IPmVEOSProcessingUnit (refer to VEOSProcessingUnit / IPmVEOSProcessingUnit <<Interface>> on page 164)

Property / IPmProperty <<Interface>>

Description

Interface providing methods to get or set the primitive data type value of a property.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
IsReadOnly	Indicates if the property value is read-only.	Get	*Boolean*
Name	Returns the property name.	Get	*String*
Type	Returns the type of the property value.	Get	*String*
Value	Returns or sets the property value.	Get/Set	*Object*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)

ProtocolCommunicationLogging <<Enumeration>>

Description

Protocol communication logging enumeration.

Enumeration values

The enumeration has the following values:

Name	Description	Value
Off	Communication logging is off.	0
All	Enumeration value for communication logging is set to all.	1
ExcludeDAQData	Communication logging is set to exclude DAQ data.	2

Returned by

The element is returned by properties or methods of the following elements:

- IPmCalibrationPlatformProtocolLogging (refer to CalibrationPlatformProtocolLogging / IPmCalibrationPlatformProtocolLogging <<Interface>> on page 32)

ProtocolConfigurationLogging <<Enumeration>>

Description

Protocol configuration logging enumeration.

Enumeration values

The enumeration has the following values:

Name	Description	Value
Off	Configuration logging is off	0
All	Enumeration value for configuration logging is set to all.	1

Returned by

The element is returned by properties or methods of the following elements:

- IPmCalibrationPlatformProtocolLogging (refer to CalibrationPlatformProtocolLogging / IPmCalibrationPlatformProtocolLogging <<Interface>> on page 32)

R

Where to go from here

Information in this section

RealTimeApplication / IPmRealTimeApplication <<Interface>>

Description

IPmRealTimeApplication Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BuildDateTime	Returns the time when the real-time application was built.	Get	*Date Time*
FullPath	Returns the path of the loaded real-time application The returned path is the original file path to the loaded application, which might not be suitable for the current file system.	Get	*String*
Name	Returns the name of the real-time application.	Get	*String*
Properties	Provides access to the supported property collection of the realtimeapplication properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmDS230xIOPlatform (refer to DS230xIOPlatform / IPmDS230xIOPlatform <<Interface>> on page 57)

- IPmRegisteredDS1006Platform (refer to RegisteredDS1006Platform / IPmRegisteredDS1006Platform <<Interface>> on page 123)
- IPmRegisteredDS1104Platform (refer to RegisteredDS1104Platform / IPmRegisteredDS1104Platform <<Interface>> on page 128)
- IPmRegisteredMABXPlatform (refer to RegisteredMABXPlatform / IPmRegisteredMABXPlatform <<Interface>> on page 134)
- IPmRegisteredMultiProcessorPlatform (refer to RegisteredMultiProcessorPlatform / IPmRegisteredMultiProcessorPlatform <<Interface>> on page 136)

RealTimeApplications / IPmRealTimeApplications <<Collection>>

Description

IPmRealTimeApplication Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of real-time applications in the collection. [0..System.Int32 -1]	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Returns True if an element that is accessible by the specified string is in the collection.	▪ *<String>* `RTAppName`: Name to access a real-time application	True if the real-time application exists. ▪ *Boolean*
Item	Returns a real-time application from the collection.	▪ *<Object>* `Index`: System.Object; either integer index or string with I/O platform name	Return value of the method. ▪ IPmControllableRealTimeApplication (refer to ControllableRealTimeApplication / IPmControllableRealTimeApplication <<Interface>> on page 35)

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmRegisteredDS1007Platform (refer to RegisteredDS1007Platform / IPmRegisteredDS1007Platform <<Interface>> on page 126)

- IPmRegisteredDS1202Platform (refer to RegisteredDS1202Platform / IPmRegisteredDS1202Platform <<Interface>> on page 130)
- IPmRegisteredDS1403Platform (refer to RegisteredDS1403Platform / IPmRegisteredDS1403Platform <<Interface>> on page 132)
- IPmRegisteredSCALEXIOPlatform (refer to RegisteredSCALEXIOPlatform / IPmRegisteredSCALEXIOPlatform <<Interface>> on page 138)

RecentHardwareItem / IPmRecentHardwareItem <<Interface>>

Description

This interface is to access a recent hardware item.

Contains hardware information.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Active	Returns the active flag.	Get	*Boolean*
Children	Returns the children collection.	Get	IPmRecentHardwareItemCollection (refer to RecentHardwareItemCollection / IPmRecentHardwareItemCollection <<Collection>> on page 121)
ConnectionType	Returns the connection type.	Get	InterfaceConnectionType (refer to InterfaceConnectionType <<Enumeration>> on page 69)
IPAddress	Returns the IPAddress.	Get	*String*
MACAddress	Returns the MACAddress.	Get	*String*
PlatformType	Returns the type of the platform.	Get	*String*
SerialNumber	Returns the serial number.	Get	*String*
UniqueName	Returns the platform's unique name.	Get	*String*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmRecentHardwareItemCollection (refer to RecentHardwareItemCollection / IPmRecentHardwareItemCollection <<Collection>> on page 121)

RecentHardwareItemCollection / IPmRecentHardwareItemCollection <<Collection>>

Description

This interface is to access the recent hardware collection.

Collection of recent hardware items.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of members in the collection.	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Checks whether register information exists in the register information collection. Index can be the zero-based integer index of the register information.	▪ *<Object>* `Index`: The index of the register information.	Returns True if register information exists. Otherwise, returns False. ▪ *Boolean*
Item	Returns the register information from the collection specified by the index. Index can be the zero-based integer index.	▪ *<Object>* `Index`: Index or name of platform.	Returns the requested register information, if available. Otherwise, returns null. ▪ IPmRecentHardwareItem (refer to RecentHardwareItem / IPmRecentHardwareItem <<Interface>> on page 120)

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmRecentHardwareItem (refer to RecentHardwareItem / IPmRecentHardwareItem <<Interface>> on page 120)

RecentPlatformConfiguration / IPmRecentPlatformConfiguration <<Collection>>

Description

This interface is to access the recent platform configuration.

Properties The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of members in the collection.	Get	*Signed 32 Bit Integer*

Methods The element has the following methods:

Name	Description	Parameter[1)]	Returns
Activate	Activates the specified entry in the recent platform configuration collection.	▪ <*Object*> `Item`: The object, platform name or index to activate.	None
Contains	Checks whether register information exists in the register information collection. Index can be the zero-based integer index of the register information.	▪ <*Object*> `Index`: Index of the register information.	Returns True if register information exists. Otherwise, returns False. ▪ *Boolean*
Deactivate	Deactivates the specified entry in the recent platform configuration collection.	▪ <*Object*> `Item`: The object, platform name or index to deactivate.	None
Export	Exports the recent hardware configuration to the passed file.	▪ <*String*> `FileFullPath`: Full path of the file to export. ▪ <*Boolean*> `OverwriteExistingFile`: Indicates if the specified file should be overwritten if it already exists.	None
Import	Imports the comprised platform entries from the passed file to the recent platform configuration and registers them.	▪ <*String*> `FileFullPath`: The parameter FileFullPath.	None
Item	Returns the register information from the collection specified by the index. Index can be the zero-based integer index.	▪ <*Object*> `Index`: Index or name of platform.	None
Remove	Removes the specified entry from the recent platform configuration collection.	▪ <*Object*> `Item`: The object, platform name or index to remove.	None
RemoveAll	Removes all entries from the recent platform configuration collection.	None	None

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)

ReducedCompatibilityBehavior <<Enumeration>>

Description

ReducedCompatibilityBehavior type enumeration.

The enumeration describes the behavior if compatibility is reduced.

Enumeration values

The enumeration has the following values:

Name	Description	Value
Abort	Abort because of reduced compatibility.	0
FixReducedIncompatibility	Ignores conflicts because of reduced compatibility.	1
IgnoreReducedIncompatibility	Ignores conflicts because of resources.	2

RegisteredDS1006Platform / IPmRegisteredDS1006Platform <<Interface>>

Description

IPmRegisteredDS1006Platform Interface

///

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BoardDetails	Returns the BoardDetails object	Get	IPmDS1006BoardDetails (refer to DS1006BoardDetails / IPmDS1006BoardDetails <<Interface>> on page 39)
ConnectionState	Gets the current connection state to indicate whether the device is connected. Not visible in property grid, only visualized by different icons.	Get	ConnectionState (refer to ConnectionState <<Enumeration>> on page 34)
ConnectionType	Gets or sets the connection type (BUS/Net).	Get	InterfaceConnectionType (refer to InterfaceConnectionType <<Enumeration>> on page 69)

Name	Description	Get/Set	Type
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*
InventoryInformation	Gets the inventory information as an XML-formatted string. This might be only the DLL version if no more details are available.	Get	IPmInventoryInformation (refer to InventoryInformation / IPmInventoryInformation <<Interface>> on page 70)
IOPlatforms	Gets the connected I/O platforms.	Get	IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)
IsAssignable	Gets a value indicating whether the platform is assignable. To avoid exceptions, use the property to check whether the platform is assignable.	Get	*Boolean*
MemoryInfo	Returns the memory information object	Get	IPmDS1006MemoryInfo (refer to DS1006MemoryInfo / IPmDS1006MemoryInfo <<Interface>> on page 40)
NetClient	To specify a net connection by IP address or alias name. Always returns the IP address.	Get	*String*
Properties	Provides access to the supported property collection of the platform properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RealTimeApplication	Returns the currently loaded real-time application.	Get	IPmRealTimeApplication (refer to RealTimeApplication / IPmRealTimeApplication <<Interface>> on page 118)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
ClearCompleteFlash	Clears the complete flash memory.	None	None
LoadRealtimeApplication	Loads the real-time application specified by ApplicationFullPath. If the real-time application is already loaded, this method reloads it. The real-time application is also loaded if another real-time application is currently running.	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None
LoadRealTimeApplicationToFlash	Loads the real-time application to the flash memory. If the real-time application is already loaded to flash, this method reloads it. The real-time application is also loaded if another real-time application is currently running (from flash).	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None
StopRTP	Stops the real-time processor. If the real-time processor was already stopped, the method just returns without any exception.	None	None

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmExperimentPlatformsCollection (refer to ExperimentPlatformsCollection / IPmExperimentPlatformsCollection <<Interface>> on page 64)
- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)
- IPmPlatformsCollection (refer to PlatformsCollection / IPmPlatformsCollection <<Collection>> on page 104)
- IPmSeekedPlatforms (refer to SeekedPlatforms / IPmSeekedPlatforms <<Collection>> on page 151)

RegisteredDS1007Platform / IPmRegisteredDS1007Platform <<Interface>>

Description

Interface to access a registered DS1007 Platform.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*
IOPlatforms	Gets the connected I/O platforms.	Get	IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)
IsAssignable	Gets a value indicating whether the platform is assignable. To avoid exceptions, use the property to check whether the platform is assignable.	Get	*Boolean*
ProcessingUnits	Gets the Substitute collection.	Get	IPmPlatformsCollection (refer to PlatformsCollection / IPmPlatformsCollection <<Collection>> on page 104)
Properties	Provides access to the supported property collection of the platform properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RealTimeApplications	Returns the currently loaded real-time application collection.	Get	IPmRealTimeApplications (refer to RealTimeApplications / IPmRealTimeApplications <<Collection>> on page 119)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
ApplyProcessorNames	Set the processornames on this platform and write it to recenthardware.xml	▪ <IPmPlatformProcessorNames (refer to PlatformProcessorNames / IPmPlatformProcessorNames	true if all names are unique and valid, otherwise false ▪ *Boolean*

Name	Description	Parameter[1)]	Returns
		<<Collection>> on page 97)> **ProcessorNamesCollection**: Collection of special list with ppc, there pus and cpu names	
ClearCompleteFlash	Clears the complete flash memory.	None	None
GetProcessorNames	Get the collection with all processorboards on this platform with all there cpu names.	None	Return value of the method. ▪ IPmPlatformProcessorNames (refer to PlatformProcessorNames / IPmPlatformProcessorNames <<Collection>> on page 97)
LoadRealtimeApplication	Loads the real-time application specified by ApplicationFullPath. If the real-time application is already loaded, this method reloads it.	▪ *<String>* **ApplicationFullPath**: Specifies the file path to the real-time application to be loaded.	None
LoadRealtimeApplicationControlled	Loads the real-time application specified by ApplicationFullPath.	▪ *<String>* **ApplicationFullPath**: Specifies the file path to the real-time application to be loaded. ▪ *<Boolean>* **UnloadConflictingApplication**: Unloads a currently running conflicting real-time application if UnloadConflictingApplication is True. Otherwise, loading is aborted if another conflicting real-time application is running. ▪ *<Boolean>* **StartAfterLoading**: Starts the real-time application if StartAfterLoading is True. Otherwise, the application is stopped after loading. ▪ <ReducedCompatibilityBehavior (refer to ReducedCompatibilityBehavior <<Enumeration>> on page 123)> **ReducedCompatibilityBehavior**: Specifies the behavior if compatibility is reduced.	None

Name	Description	Parameter[1)]	Returns
LoadRealTimeApplicationToFlash	Loads the real-time application to the flash memory. If the real-time application is already loaded to flash, this method reloads it. The real-time application is also loaded if another real-time application is currently running (from flash).	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None

[1)] <Type> `Name`: Description

RegisteredDS1104Platform / IPmRegisteredDS1104Platform <<Interface>>

Description

IPmRegisteredDS1104Platform Interface

///

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BoardDetails	Returns the BoardDetails object	Get	IPmDS1104BoardDetails (refer to DS1104BoardDetails / IPmDS1104BoardDetails <<Interface>> on page 45)
ConnectionState	Gets the current connection state to indicate whether the device is connected. Not visible in property grid, only visualized by different icons.	Get	ConnectionState (refer to ConnectionState <<Enumeration>> on page 34)
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*
InventoryInformation	Gets the inventory information as an XML-formatted string. This might be only the DLL version if no more details are available.	Get	IPmInventoryInformation (refer to InventoryInformation / IPmInventoryInformation <<Interface>> on page 70)
IsAssignable	Gets a value indicating whether the platform is assignable. To avoid exceptions, use the property to check whether the platform is assignable.	Get	*Boolean*
MemoryInfo	Returns the memory information object	Get	IPmDS1104MemoryInfo (refer to DS1104MemoryInfo /

Name	Description	Get/Set	Type
			IPmDS1104MemoryInfo <<Interface>> on page 46)
Properties	Provides access to the supported property collection of the pu properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RealTimeApplication	Returns the currently loaded real-time application.	Get	IPmRealTimeApplication (refer to RealTimeApplication / IPmRealTimeApplication <<Interface>> on page 118)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
ClearCompleteFlash	Clears the complete flash memory.	None	None
LoadRealtimeApplication	Loads the real-time application specified by ApplicationFullPath. If the real-time application is already loaded, this method reloads it. The real-time application is also loaded if another real-time application is currently running.	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None
LoadRealTimeApplicationToFlash	Loads the real-time application to the flash memory. If the real-time application is already loaded to flash, this method reloads it. The real-time application is also loaded if another real-time application is currently running (from flash).	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None
StopRTP	Stops the real-time processor. If the real-time processor was already stopped, the method just returns without any exception.	None	None

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmExperimentPlatformsCollection (refer to ExperimentPlatformsCollection / IPmExperimentPlatformsCollection <<Interface>> on page 64)
- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)
- IPmPlatformsCollection (refer to PlatformsCollection / IPmPlatformsCollection <<Collection>> on page 104)
- IPmSeekedPlatforms (refer to SeekedPlatforms / IPmSeekedPlatforms <<Collection>> on page 151)

RegisteredDS1202Platform / IPmRegisteredDS1202Platform <<Interface>>

Description

Interface to access a DS1202 platform.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*
IOPlatforms	Gets the connected I/O platforms.	Get	IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)
IsAssignable	Gets a value indicating whether the platform is assignable. To avoid exceptions, use the property to check whether the platform is assignable.	Get	*Boolean*
ProcessingUnits	Gets the substitute collection.	Get	IPmPlatformsCollection (refer to PlatformsCollection / IPmPlatformsCollection <<Collection>> on page 104)
Properties	Provides access to the supported property collection of the platform properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RealTimeApplications	Returns the currently loaded real-time application collection.	Get	IPmRealTimeApplications (refer to RealTimeApplications / IPmRealTimeApplications <<Collection>> on page 119)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Name	Description	Get/Set	Type
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
ClearCompleteFlash	Clears the complete flash memory.	None	None
LoadRealtimeApplication	Loads the real-time application specified by ApplicationFullPath. If the real-time application is already loaded, this method reloads it.	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None
LoadRealtimeApplicationControlled	Loads the real-time application specified by ApplicationFullPath.	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded. ▪ *<Boolean>* `UnloadConflictingApplication`: Unloads a currently running conflicting real-time application if UnloadConflictingApplication is True. Otherwise, loading is aborted if another conflicting real-time application is running. ▪ *<Boolean>* `StartAfterLoading`: Starts the real-time application if StartAfterLoading is True. Otherwise, the application is stopped after loading. ▪ <ReducedCompatibilityBehavior (refer to ReducedCompatibilityBehavior <<Enumeration>> on page 123)> `ReducedCompatibilityBehavior`: Specifies the behavior if compatibility is reduced.	None

Name	Description	Parameter[1)]	Returns
LoadRealTimeApplicationToFlash	Loads the real-time application to the flash memory. If the real-time application is already loaded to flash, this method reloads it. The real-time application is also loaded if another real-time application is currently running (from flash).	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None

[1)] <Type> `Name`: Description

RegisteredDS1403Platform / IPmRegisteredDS1403Platform <<Interface>>

Description

IPmRegisteredDS1403Platform Interface (RCP Device)

///

Properties

The element has the following properties:

Name	Description	Get/Set	Type
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*
IOPlatforms	Gets the connected I/O platforms.	Get	IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)
IOUnits	Gets the units of the platform.	Get	IPmUnits (refer to Units / IPmUnits <<Collection>> on page 158)
IsAssignable	Gets a value indicating whether the platform is assignable. To avoid exceptions, use the property to check whether the platform is assignable.	Get	*Boolean*
ProcessingUnits	Gets the Substitute collection.	Get	IPmPlatformsCollection (refer to PlatformsCollection / IPmPlatformsCollection <<Collection>> on page 104)
Properties	Provides access to the supported property collection of the platform properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RealTimeApplications	Gets the currently loaded real-time application collection.	Get	IPmRealTimeApplications (refer to RealTimeApplications /

Name	Description	Get/Set	Type
			IPmRealTimeApplications <<Collection>> on page 119)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
ClearCompleteFlash	Clears the complete flash memory.	None	None
InitializeHardwareConfiguration	Writes a hardware configuration provided by the HTFX file to the platform.	▪ <*String*> `HtfxFilePath`: Specifies the HTFX file path.	None
LoadRealtimeApplication	Loads the real-time application specified by ApplicationFullPath. If the real-time application is already loaded, this method reloads it.	▪ <*String*> `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None
LoadRealtimeApplicationControlled	Loads the real-time application specified by ApplicationFullPath.	▪ <*String*> `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded. ▪ <*Boolean*> `UnloadConflictingApplication`: Unloads a currently running conflicting real-time application if UnloadConflictingApplication is True. Otherwise, loading is aborted if another conflicting real-time application is running. ▪ <*Boolean*> `StartAfterLoading`: Starts the real-time application if StartAfterLoading is True. Otherwise, the application is stopped after loading. ▪ <ReducedCompatibilityBehavior (refer to	None

Name	Description	Parameter[1)]	Returns
		ReducedCompatibilityBehavior <<Enumeration>> on page 123)> `ReducedCompatibilityBehavior`: Specifies the behavior if compatibility is reduced.	
LoadRealTimeApplicationToFlash	Loads the real-time application to the flash memory. If the real-time application is already loaded to flash, this method reloads it. The real-time application is also loaded if another real-time application is currently running (from flash).	▪ <*String*> `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None

1) <Type> `Name`: Description

RegisteredMABXPlatform / IPmRegisteredMABXPlatform <<Interface>>

Description

IPmRegisteredMABXPlatform Interface

///

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BoardDetails	Returns the BoardDetails object	Get	IPmMABXBoardDetails (refer to MABXBoardDetails / IPmMABXBoardDetails <<Interface>> on page 80)
ConnectionState	Gets the current connection state to indicate whether the device is connected. Not visible in property grid, only visualized by different icons.	Get	ConnectionState (refer to ConnectionState <<Enumeration>> on page 34)
ConnectionType	Gets or sets the connection type (BUS/Net).	Get	InterfaceConnectionType (refer to InterfaceConnectionType <<Enumeration>> on page 69)
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*

Name	Description	Get/Set	Type
InventoryInformation	Gets the inventory information as an XML-formatted string. This might be only the DLL version if no more details are available.	Get	IPmInventoryInformation (refer to InventoryInformation / IPmInventoryInformation <<Interface>> on page 70)
IOPlatforms	Gets the connected I/O platforms.	Get	IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)
IsAssignable	Gets a value indicating whether the platform is assignable. To avoid exceptions, use the property to check whether the platform is assignable.	Get	*Boolean*
MemoryInfo	Returns the memory information object	Get	IPmMABXMemoryInfo (refer to MABXMemoryInfo / IPmMABXMemoryInfo <<Interface>> on page 81)
NetClient	To specify a net connection by IP address or alias name. Always returns the IP address.	Get	*String*
PlugState	Returns the platform's plug state.	Get	PlugState (refer to PlugState <<Enumeration>> on page 109)
Properties	Provides access to the supported property collection of the platform properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RealTimeApplication	Returns the currently loaded real-time application.	Get	IPmRealTimeApplication (refer to RealTimeApplication / IPmRealTimeApplication <<Interface>> on page 118)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
ClearCompleteFlash	Clears the complete flash memory.	None	None
ClearFlashApplicationMemory	Clears the real-time application from the flash memory.	None	None
ClearFlightRecorderMemory	Clears the flight recorder from the flash memory.	None	None
ClearNonvolatileData	Clears the nonvolatile data from the flash memory.	None	None

Name	Description	Parameter[1)]	Returns
LoadRealtimeApplication	Loads the real-time application specified by ApplicationFullPath. If the real-time application is already loaded, this method reloads it. The real-time application is also loaded if another real-time application is currently running.	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None
LoadRealTimeApplicationToFlash	Loads the real-time application to the flash memory. If the real-time application is already loaded to flash, this method reloads it. The real-time application is also loaded if another real-time application is currently running (from flash).	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None
StopRTP	Stops the real-time processor. If the real-time processor was already stopped, the method just returns without any exception.	None	None

1) <Type> `Name`: Description

RegisteredMultiProcessorPlatform / IPmRegisteredMultiProcessorPlatform <<Interface>>

Description

IPmRegisteredMultiProcessorPlatform Interface

///

Properties

The element has the following properties:

Name	Description	Get/Set	Type
ConnectionState	Gets the current connection state to indicate whether the device is connected. Not visible in property grid, only visualized by different icons.	Get	ConnectionState (refer to ConnectionState <<Enumeration>> on page 34)
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*

Name	Description	Get/Set	Type
IsAssignable	Gets a value indicating whether the platform is assignable. To avoid exceptions, use the property to check whether the platform is assignable.	Get	*Boolean*
MultiprocessorType	Gets the processor board type.	Get	*Signed 32 Bit Integer*
Platforms	Gets the platforms of the multiprocessor system.	Get	IPmPlatformsCollection (refer to PlatformsCollection / IPmPlatformsCollection <<Collection>> on page 104)
Properties	Provides access to the supported property collection of the platform properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RealTimeApplication	Returns the currently loaded real-time application.	Get	IPmRealTimeApplication (refer to RealTimeApplication / IPmRealTimeApplication <<Interface>> on page 118)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
ClearCompleteFlash	Clears the complete flash memory.	None	None
GetProcessorName	Gets the processor name by index.	▪ *<Object>* `Index`: The parameter Index.	The processor name by index. ▪ *String*
IsTopologyValid	Checks if the MP topology is valid.	None	True if the topology is valid. ▪ *Boolean*
LoadRealtimeApplication	Loads the real-time application specified by ApplicationFullPath. If the real-time application is already loaded, this method reloads it. The real-time application is also loaded if another real-time application is currently running.	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None
LoadRealTimeApplicationToFlash	Loads the real-time application to the flash memory. If the real-time application is already loaded to flash, this	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the	None

Name	Description	Parameter[1)]	Returns
	method reloads it. The real-time application is also loaded if another real-time application is currently running (from flash).	real-time application to be loaded.	
SetProcessorName	Sets the processor name by index.	▪ *<Object>* `Index`: The parameter Index. ▪ *<String>* `ProcessorName`: The parameter ProcessorName.	None
StopRTP	Stops the real-time processor. If the real-time processor was already stopped, the method just returns without any exception.	None	None

[1)] <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmExperimentPlatformsCollection (refer to ExperimentPlatformsCollection / IPmExperimentPlatformsCollection <<Interface>> on page 64)
- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)
- IPmPlatformsCollection (refer to PlatformsCollection / IPmPlatformsCollection <<Collection>> on page 104)
- IPmSeekedPlatforms (refer to SeekedPlatforms / IPmSeekedPlatforms <<Collection>> on page 151)

RegisteredSCALEXIOPlatform / IPmRegisteredSCALEXIOPlatform <<Interface>>

Description

IPmRegisteredSCALEXIOPlatform Interface (RCP Device)

///

Properties

The element has the following properties:

Name	Description	Get/Set	Type
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*

Name	Description	Get/Set	Type
IOPlatforms	Gets the connected I/O platforms.	Get	IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)
IOUnits	Gets the units of the platform.	Get	IPmUnits (refer to Units / IPmUnits <<Collection>> on page 158)
IsAssignable	Gets a value indicating whether the platform is assignable. To avoid exceptions, use the property to check whether the platform is assignable.	Get	*Boolean*
ProcessingUnits	Gets the Substitute collection.	Get	IPmPlatformsCollection (refer to PlatformsCollection / IPmPlatformsCollection <<Collection>> on page 104)
Properties	Provides access to the supported property collection of the platform properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RealTimeApplications	Gets the currently loaded real-time application collection.	Get	IPmRealTimeApplications (refer to RealTimeApplications / IPmRealTimeApplications <<Collection>> on page 119)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
ClearCompleteFlash	Clears the complete flash memory.	None	None
InitializeHardwareConfiguration	Writes a hardware configuration provided by the HTFX file to the platform.	▪ <*String*> `HtfxFilePath`: Specifies the HTFX file path.	None
LoadRealtimeApplication	Loads the real-time application specified by ApplicationFullPath. If the real-time application is already loaded, this method reloads it.	▪ <*String*> `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None
LoadRealtimeApplicationControlled	Loads the real-time application specified by ApplicationFullPath.	▪ <*String*> `ApplicationFullPath`: Specifies the file path to the	None

Name	Description	Parameter[1)]	Returns
		real-time application to be loaded. ▪ *<Boolean>* `UnloadConflictingApplication`: Unloads a currently running conflicting real-time application if UnloadConflictingApplication is True. Otherwise, loading is aborted if another conflicting real-time application is running. ▪ *<Boolean>* `StartAfterLoading`: Starts the real-time application if StartAfterLoading is True. Otherwise, the application is stopped after loading. ▪ <ReducedCompatibilityBehavior (refer to ReducedCompatibilityBehavior <<Enumeration>> on page 123)> `ReducedCompatibilityBehavior`: Specifies the behavior if compatibility is reduced.	
LoadRealTimeApplicationToFlash	Loads the real-time application to the flash memory. If the real-time application is already loaded to flash, this method reloads it. The real-time application is also loaded if another real-time application is currently running (from flash).	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmExperimentPlatformsCollection (refer to ExperimentPlatformsCollection / IPmExperimentPlatformsCollection <<Interface>> on page 64)
- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)
- IPmPlatformsCollection (refer to PlatformsCollection / IPmPlatformsCollection <<Collection>> on page 104)
- IPmSeekedPlatforms (refer to SeekedPlatforms / IPmSeekedPlatforms <<Collection>> on page 151)

RegisteredVEOSPlatform / IPmRegisteredVEOSPlatform <<Interface>>

Description Interface to access a registered VEOS platform.

Properties The element has the following properties:

Name	Description	Get/Set	Type
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*
IsAssignable	Gets a value indicating whether the platform is assignable. To avoid exceptions, use the property to check whether the platform is assignable.	Get	*Boolean*
ProcessingUnits	Gets the Substitute collection.	Get	IPmPlatformsCollection (refer to PlatformsCollection / IPmPlatformsCollection <<Collection>> on page 104)
Properties	Provides access to the supported property collection of the platform properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RealTimeApplications	Returns the currently loaded real-time application collection.	Get	IPmVEOSApplications (refer to VEOSApplications / IPmVEOSApplications <<Collection>> on page 162)
SimulationTimeOptions	Gets the simulation time options object.	Get	IPmVEOSSimulationTimeOptions (refer to VEOSSimulationTimeOptions / IPmVEOSSimulationTimeOptions <<Interface>> on page 165)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
LoadRealtimeApplication	Loads the real-time application specified by ApplicationFullPath. If the real-time application is already loaded, this method reloads it.	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded.	None
LoadRealtimeApplicationControlled	Loads the real-time application specified by ApplicationFullPath.	▪ *<String>* `ApplicationFullPath`: Specifies the file path to the real-time application to be loaded. ▪ *<Boolean>* `UnloadConflictingApplication`: Unloads a currently running conflicting real-time application if UnloadConflictingApplication is True. Otherwise, loading is aborted if another conflicting real-time application is running. ▪ *<Boolean>* `StartAfterLoading`: Starts the real-time application if StartAfterLoading is True. Otherwise, the application is stopped after loading. ▪ <ReducedCompatibilityBehavior (refer to ReducedCompatibilityBehavior <<Enumeration>> on page 123)> `ReducedCompatibilityBehavior`: Specifies the behavior if compatibility is reduced.	None

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmExperimentPlatformsCollection (refer to ExperimentPlatformsCollection / IPmExperimentPlatformsCollection <<Interface>> on page 64)
- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)
- IPmPlatformsCollection (refer to PlatformsCollection / IPmPlatformsCollection <<Collection>> on page 104)

- IPmSeekedPlatforms (refer to SeekedPlatforms / IPmSeekedPlatforms <<Collection>> on page 151)

RegisteredXILAPIMAPortPlatform / IPmRegisteredXILAPIMAPortPlatform <<Interface>>

Description

IPmRegistreredXILAPIMAPortPlatformInternal Interface

///

Properties

The element has the following properties:

Name	Description	Get/Set	Type
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*
IsAssignable	Gets a value indicating whether the platform is assignable. To avoid exceptions, use the property to check whether the platform is assignable.	Get	*Boolean*
MAPort	Returns the access to model access port.	Get	IPmXILAPIMAPort (refer to XILAPIMAPort / IPmXILAPIMAPort <<Interface>> on page 167)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods

The element has no methods.

RegisterInfos / IPmRegisterInfos <<Collection>>

Description IPmRegisterInfos Interface

Properties The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of members in the collection.	Get	*Signed 32 Bit Integer*

Methods The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Checks whether register information exists in the register information collection. Index can be the zero-based integer index of the register information.	▪ <*Object*> `Index`: Index of the register information.	Returns True if register information exists. Otherwise, returns False. ▪ *Boolean*
Item	Returns the register information from the collection specified by the index. Index can be the zero-based integer index.	▪ <*Object*> `Index`: Index or name of platform.	None
Remove	Removes register information from the collection. Index can be the zero-based integer index of the register information.	▪ <*Object*> `Index`: The index of the register information to remove.	None
RemoveAll	Removes all register information from the collection.	None	Returns True if successful. Otherwise, returns False. ▪ *Boolean*

[1)] <Type> `Name`: Description

S

Where to go from here

Information in this section

SCALEXIOHardwareInformation / IPmSCALEXIOHardwareInformation <<Interface>>

Description

IPmSCALEXIOHardwareInformation interface (RCP/HIL device).

Properties

The element has the following properties:

Name	Description	Get/Set	Type
CPU	Gets the CPU type of the SCALEXIO hardware.	Get	*String*
FlashDriveSize	Gets the flash drive size of the SCALEXIO hardware.	Get	*Signed 64 Bit Integer*
Frequency	Gets the frequency of the SCALEXIO hardware.	Get	*Double*
ProductVersion	Gets the product version of the SCALEXIO hardware.	Get	*String*
RAMSize	Gets the RAM size of the SCALEXIO hardware.	Get	*Signed 32 Bit Integer*
RAMSize64	returns the RAM size of the SCALEXIO hardware	Get	*Signed 64 Bit Integer*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmSCALEXIOProcessingUnit (refer to SCALEXIOProcessingUnit / IPmSCALEXIOProcessingUnit <<Interface>> on page 148)

SCALEXIOIdentificationInformation / IPmSCALEXIOIdentificationInformation <<Interface>>

Description

IPmSCALEXIOIdentificationInformation interface (RCP/HIL device).

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BoardName	Gets the board name of the SCALEXIO hardware.	Get	*String*
BoardType	Gets the board type of the SCALEXIO hardware.	Get	*String*
Identifier	Gets the serial number of the SCALEXIO hardware.	Get	*Signed 32 Bit Integer*

Name	Description	Get/Set	Type
IPAddress	Gets the IP address of the SCALEXIO hardware.	Get	*String*
MACAddress	Gets the MAC address of the SCALEXIO hardware.	Get	*String*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmSCALEXIOProcessingUnit (refer to SCALEXIOProcessingUnit / IPmSCALEXIOProcessingUnit <<Interface>> on page 148)

SCALEXIOPlatformConnectionSettings / IPmSCALEXIOPlatformConnectionSettings <<Interface>>

Description

IPmSCALEXIOPlatformConnectionSettings interface (RCP/HIL device).

Properties

The element has the following properties:

Name	Description	Get/Set	Type
AliasName	Gets or sets the alias name as the filter criteria for a scan.	Get	*String*
BoardName	Gets or sets the board name as the filter criterion for a scan.	Get	*String*
Identifier	Gets or sets the serial number as the filter criterion for a scan.	Get	*Signed 32 Bit Integer*
IPAddress	Gets or sets the IP address as the filter criterion for a scan.	Get	*String*

Methods

The element has no methods.

SCALEXIOProcessingUnit / IPmSCALEXIOProcessingUnit <<Interface>>

Description IPmSCALEXIOProcessingUnit Interface.

Properties The element has the following properties:

Name	Description	Get/Set	Type
ApplicationParts	Returns the currently loaded real-time application parts collection.	Get	IPmApplicationParts (refer to ApplicationParts / IPmApplicationParts <<Collection>> on page 25)
BoardHardware	Gets the hardware information object.	Get	IPmSCALEXIOHardwareInformation (refer to SCALEXIOHardwareInformation / IPmSCALEXIOHardwareInformation <<Interface>> on page 146)
BoardName	Gets the board name.	Get	*String*
ConnectionState	Gets the current connection state to indicate whether the device is connected. Not visible in the property grid, visualized only by different icons.	Get	ConnectionState (refer to ConnectionState <<Enumeration>> on page 34)
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*
EthernetAdapters	Returns the ethernet adapter collection of the ethernet adapters.	Get	IPmEthernetAdapters (refer to EthernetAdapters / IPmEthernetAdapters <<Collection>> on page 61)
EthernetSwitches	Returns a collection including the ethernet switches of processing unit's IO platforms.	Get	IPmEthernetSwitches (refer to EthernetSwitches / IPmEthernetSwitches <<Collection>> on page 63)
Identification	Gets the identification information object.	Get	IPmSCALEXIOIdentificationInformation (refer to SCALEXIOIdentificationInformation / IPmSCALEXIOIdentificationInformation <<Interface>> on page 146)
IOPlatforms	Gets the connected I/O platforms.	Get	IPmIOPlatforms (refer to IOPlatforms / IPmIOPlatforms <<Collection>> on page 73)

Name	Description	Get/Set	Type
Properties	Provides access to the supported property collection of the pu properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RackName	Gets the name of the rack the hardware element belongs to.	Get	*String*
Software	Gets the software information object.	Get	IPmSCALEXIOSoftwareInformation (refer to SCALEXIOSoftwareInformation / IPmSCALEXIOSoftwareInformation <<Interface>> on page 151)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
SetBoardName	Sets the board name.	▪ <*String*> `Name`: Board name to set.	None
SetRackName	Sets the name of the rack the hardware element belongs to.	▪ <*String*> `Name`: Rack name to set.	None

1) <Type> `Name`: Description

SCALEXIORegisterInfo / IPmSCALEXIORegisterInfo <<Interface>>

Description

IPmSCALEXIORegisterInfo Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
AliasName	Property to specify the serial number that is used for assignment	Get/Set	*String*
BoardName	Property to specify the serial number that is used for assignment	Get/Set	*String*

Name	Description	Get/Set	Type
MACAddress	Gets or sets the MAC address.	Get/Set	*String*
NetClient	Property to specify the net client that is used for assignment	Get/Set	*String*
Type	Returns the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)

SCALEXIORegistrationInfo / IPmSCALEXIORegistrationInfo <<Interface>>

Description

IPmSCALEXIORegistrationInfo Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Name	Property to specify the the custom platform, processing unit or processor board name.	Get/Set	*String*
RegistrationInfos	Registration infos of member processing units of platform.	Get	IPmProcessingUnitRegisterInfos (refer to ProcessingUnitRegisterInfos / IPmProcessingUnitRegisterInfos <<Collection>> on page 110)
Type	Returns the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Methods

The element has no methods.

SCALEXIOSoftwareInformation / IPmSCALEXIOSoftwareInformation <<Interface>>

Description

IPmSCALEXIOSoftwareInformation interface (RCP/HIL device).

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BIOSVersion	Gets the version of the BIOS	Get	*String*
FirmwareVersion	Gets the version of the firmware.	Get	*String*
FPGACoreVersion	Gets the version of the core FPGA.	Get	*String*
FPGAVersion	Gets the version of the FPGA.	Get	*String*
LastFirmwareUpdate	Gets the date of the last firmware update as a string.	Get	*String*
LastFPGAUpdate	Gets the date of the last FPGA update as a string.	Get	*String*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmSCALEXIOProcessingUnit (refer to SCALEXIOProcessingUnit / IPmSCALEXIOProcessingUnit <<Interface>> on page 148)

SeekedPlatforms / IPmSeekedPlatforms <<Collection>>

Description

This interface is to access the sought platforms.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of platforms in the collection.	Get	*Signed 32 Bit Integer*
UniqueNames	Returns the collection of platform unique names.	Get	IPmPlatformNames (refer to PlatformNames / IPmPlatformNames <<Collection>> on page 96)

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Checks whether a platform exists in the platforms collection. The index can be the zero-based integer index or the name of the platform.	▪ *<Object>* `Index`: Index or name of platform.	Returns True if the platform exists. Otherwise, returns False. ▪ *Boolean*
Item	Returns the platform from the collection specified by the index. The index can be the zero-based integer index or the name of the platform.	▪ *<Object>* `Index`: Index or name of platform.	Returns the requested platform, if it exists. Otherwise, returns null. ▪ *dSPACE.PlatformManagement.Automation.IPmDS1006Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmDS1007Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmDS1104Platform* ▪ *dSPACE.PlatformManagement.Automation.IPmMABXPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmMultiProcessorPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmSCALEXIOPlatform* ▪ *dSPACE.PlatformManagement.Automation.IPmVEOSPlatform* ▪ IPmRegisteredDS1006Platform (refer to RegisteredDS1006Platform / IPmRegisteredDS1006Platform <<Interface>> on page 123) ▪ IPmRegisteredDS1104Platform (refer to RegisteredDS1104Platform / IPmRegisteredDS1104Platform <<Interface>> on page 128) ▪ IPmRegisteredMultiProcessorPlatform (refer to RegisteredMultiProcessorPlatform / IPmRegisteredMultiProcessor

Name	Description	Parameter[1)]	Returns
			Platform <<Interface>> on page 136) ▪ IPmRegisteredSCALEXIOPlatform (refer to RegisteredSCALEXIOPlatform / IPmRegisteredSCALEXIOPlatform <<Interface>> on page 138) ▪ IPmRegisteredVEOSPlatform (refer to RegisteredVEOSPlatform / IPmRegisteredVEOSPlatform <<Interface>> on page 141)

[1)] <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)

SubstitutePlatform / IPmSubstitutePlatform <<Interface>>

Description

interface IPmSubstitutePlatform

Properties

The element has the following properties:

Name	Description	Get/Set	Type
ActiveVariableDescription	Gets the platform's active variable description.	Get	*dSPACE.ToolAutomation.ControlDeskNG.IXaActiveVariableDescription*
Capabilities	Gets the capabilities of the platform. The Capabilities property can be used to check whether a platform can calibrate or measure.	Get	Capabilities (refer to Capabilities <<Enumeration>> on page 33)
ConnectionState	Gets the current connection state to indicate whether the device is connected. Not visible in the property grid, visualized only by different icons.	Get	ConnectionState (refer to ConnectionState <<Enumeration>> on page 34)
DisplayName	Gets the name of the device, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*

Name	Description	Get/Set	Type
IsAssignable	Gets a value indicating whether the platform is assignable. To avoid exceptions, use the property to check whether the platform is assignable.	Get	*Boolean*
Name	Gets the name of the device used in an experiment.	Get	*String*
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*
VariableDescriptions	Gets the list of variable descriptions available for the platform.	Get	*dSPACE.ToolAutomation.ControlDeskNG.IXaVariableDescriptions*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Connect	Connects the platform to the specified (virtual) hardware according to the current configuration settings. If the platform is already connected, the method returns without an exception.	None	None
Disconnect	Disconnects the platform from the (virtual) hardware. If the platform is already disconnected, the method returns without an exception.	None	None

1) <Type> `Name`: Description

T

Where to go from here

Information in this section

TransportLayerCommunicationLogging <<Enumeration>>

Description

Transport layer communication logging enumeration.

Enumeration values

The enumeration has the following values:

Name	Description	Value
Off	Enumeration value for transport layer communication logging is off.	0
All	Enumeration value for transport layer communication logging is all.	1

Returned by

The element is returned by properties or methods of the following elements:

- IPmCalibrationPlatformTransportLayerLogging (refer to CalibrationPlatformTransportLayerLogging / IPmCalibrationPlatformTransportLayerLogging <<Interface>> on page 32)

TransportLayerConfigurationLogging <<Enumeration>>

Description

Transport layer configuration logging enumeration.

Enumeration values

The enumeration has the following values:

Name	Description	Value
Off	Enumeration value for transport layer configuration logging is off.	0
All	Enumeration value for transport layer configuration logging is all.	1

Returned by

The element is returned by properties or methods of the following elements:

- IPmCalibrationPlatformTransportLayerLogging (refer to CalibrationPlatformTransportLayerLogging / IPmCalibrationPlatformTransportLayerLogging <<Interface>> on page 32)

U

Where to go from here

Information in this section

Unit / IPmUnit <<Interface>>

Description

IPmUnit Interface

Interface representing units of smart platforms.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Name	Gets the name of the unit.	Get	*String*
Properties	Provides access to the supported property collection of the platform properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
RackName	Gets the name of the rack the hardware element belongs to.	Get	*String*
Type	Gets the type of the unit.	Get	*String*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
SetName	Sets the name of the unit.	▪ <*String*> `Name`: Unit name to set.	None
SetRackName	Sets the name of the rack the hardware element belongs to.	▪ <*String*> `Name`: Rack name to set.	None

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmUnits (refer to Units / IPmUnits <<Collection>> on page 158)

Units / IPmUnits <<Collection>>

Description

IPmUnits Interface

Interface representing a collection of smart units.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of members in the collection.	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Item	Returns the unit from the collection specified by the index. The index can be the zero-based integer index.	▪ *<Signed 32 Bit Integer>* `Index`: Index of unit in collection.	Returns the requested unit if it exists. Otherwise, returns null. ▪ IPmUnit (refer to Unit / IPmUnit <<Interface>> on page 157)

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmRegisteredDS1403Platform (refer to RegisteredDS1403Platform / IPmRegisteredDS1403Platform <<Interface>> on page 132)
- IPmRegisteredSCALEXIOPlatform (refer to RegisteredSCALEXIOPlatform / IPmRegisteredSCALEXIOPlatform <<Interface>> on page 138)

V

Where to go from here

Information in this section

VariableObserverRates <<Enumeration>>

Description

VariableObserverRates enumeration type.

Enumeration values

The enumeration has the following values:

Name	Description	Value
VariableObserverRate25ms	VariableObserver runs at 25ms. VariableObserverRate25ms enumeration value of enumuration type VariableObserverRates.	25
VariableObserverRate50ms	VariableObserver runs at 50ms. VariableObserverRate50ms enumeration value of enumuration type VariableObserverRates.	50
VariableObserverRate100ms	VariableObserver runs at 100ms. VariableObserverRate100ms enumeration value of enumuration type VariableObserverRates.	100

Name	Description	Value
VariableObserverRate250ms	VariableObserver runs at 250ms. VariableObserverRate250ms enumeration value of enumuration type VariableObserverRates.	250
VariableObserverRate500ms	VariableObserver runs at 500ms. VariableObserverRate500ms enumeration value of enumuration type VariableObserverRates.	500
VariableObserverRate1000ms	VariableObserver runs at 1000ms. VariableObserverRate1000ms enumeration value of enumuration type VariableObserverRates.	1000
VariableObserverRate2000ms	VariableObserver runs at 2000ms. VariableObserverRate2000ms enumeration value of enumuration type VariableObserverRates.	2000
VariableObserverRate5000ms	VariableObserver runs at 5000ms. VariableObserverRate5000ms enumeration value of enumuration type VariableObserverRates.	5000

Returned by

The element is returned by properties or methods of the following elements:

- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)

VEOSApplication / IPmVEOSApplication <<Interface>>

Description

IPmVEOSApplication Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
ApplicationParts	Returns the application parts of the current application.	Get	IPmApplicationParts (refer to ApplicationParts / IPmApplicationParts <<Collection>> on page 25)
BuildDateTime	Returns the time when the real-time application was built.	Get	*Date Time*
FullPath	Returns the path of the loaded real-time application The returned path is the original file path to the loaded application, which might not be suitable for the current file system.	Get	*String*
Name	Returns the name of the real-time application.	Get	*String*
Properties	Provides access to the supported property collection of the realtimeapplication properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
State	Gets the state of the offline simulation application.	Get	ApplicationState (refer to ApplicationState <<Enumeration>> on page 27)

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Pause	Pauses the offline simulation application. To pause the offline simulation application, you need exclusive access to it.	None	None
SingleStep	Steps through the offline simulation application. To step through the offline simulation application, you need exclusive access to it.	None	None
Start	Starts the offline simulation application. The method returns whether the offline simulation application started without an exception. To start the real-time application, you need shared access to it.	None	None
Stop	Stops the offline simulation application The method returns whether the offline simulation application stopped without an exception. To stop the real.time application, you need shared access to it.	None	None
Unload	Unloads the offline simulation application. To unload the offline simulation application, you need exclusive access to it.	None	None

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmVEOSApplications (refer to VEOSApplications / IPmVEOSApplications <<Collection>> on page 162)

VEOSApplications / IPmVEOSApplications <<Collection>>

Description

IPmVEOSApplications Interface

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of real-time applications in the collection. [0..System.Int32 -1]	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Contains	Returns True if an element that is accessible by the specified string is in the collection.	▪ *<String>* `RTAppName`: Name to access a real-time application	True if the real-time application exists. ▪ *Boolean*
Item	Returns a VEOS application from the collection.	▪ *<Object>* `Index`: System.Object	Return value of the method. ▪ IPmVEOSApplication (refer to VEOSApplication / IPmVEOSApplication <<Interface>> on page 160)

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmRegisteredVEOSPlatform (refer to RegisteredVEOSPlatform / IPmRegisteredVEOSPlatform <<Interface>> on page 141)

VEOSIdentificationInformation / IPmVEOSIdentificationInformation <<Interface>>

Description

IPmVEOSIdentificationInformation interface.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BoardName	Gets the board name of VEOS.	Get	*String*
BoardType	Gets the board type of VEOS.	Get	*String*
IPAddress	Gets the IP address of VEOS.	Get	*String*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmVEOSProcessingUnit (refer to VEOSProcessingUnit / IPmVEOSProcessingUnit <<Interface>> on page 164)

VEOSPlatformConnectionSettings / IPmVEOSPlatformConnectionSettings <<Interface>>

Description

IPmVEOSPlatformConnectionSettings interface.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
BoardName	Gets or sets the board name as the filter criterion for a scan.	Get	*String*
IPAddress	Gets or sets the IP address as the filter criterion for a scan.	Get	*String*

Methods

The element has no methods.

VEOSProcessingUnit / IPmVEOSProcessingUnit <<Interface>>

Description IPmVEOSProcessingUnit Interface.

Properties The element has the following properties:

Name	Description	Get/Set	Type
ApplicationParts	Returns the currently loaded real-time application parts collection.	Get	IPmApplicationParts (refer to ApplicationParts / IPmApplicationParts <<Collection>> on page 25)
ConnectionState	Gets the current connection state to indicate whether the device is connected. Not visible in the property grid, visualized only by different icons.	Get	ConnectionState (refer to ConnectionState <<Enumeration>> on page 34)
DisplayName	Gets the name of the platform, including type information, as shown in the navigator. Do not use the display name as an identifier in automation. Use a specified or unique name instead.	Get	*String*
Identification	Gets the identification information object.	Get	IPmVEOSIdentificationInformation (refer to VEOSIdentificationInformation / IPmVEOSIdentificationInformation <<Interface>> on page 163)
Properties	Provides access to the supported property collection of the pu properties.	Get	IPmProperties (refer to Properties / IPmProperties <<Collection>> on page 113)
Type	Gets the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)
UniqueName	Gets the name of the platform used in the Platform Manager. Do not use the display name as an identifier in automation. Use the specified name instead.	Get	*String*

Methods The element has no methods.

VEOSRegisterInfo / IPmVEOSRegisterInfo <<Interface>>

Description

IPmVEOSRegisterInfo interface.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Name	Property to specify the the custom platform, processing unit or processor board name.	Get/Set	*String*
NetClient	Property to specify the net client that is used for assignment	Get/Set	*String*
Type	Returns the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmPlatformManagement (refer to PlatformManagement / IPmPlatformManagement <<Interface>> on page 91)

VEOSSimulationTimeOptions / IPmVEOSSimulationTimeOptions <<Interface>>

Description

IPmVEOSSimulationTimeOptions interface.

Properties

The element has the following properties:

Name	Description	Get/Set	Type
AccelerationFactor	Sets or gets the AccelerationFactor of the offline simulation application. To set or get the AccelerationFactor of the offline simulation application, you need exclusive access to it.	Get/Set	*String*

Name	Description	Get/Set	Type
BreakAtTime	Sets or gets the break-at time of the offline simulation application. To set or get the break-at time of the offline simulation application, you need exclusive access to it.	Get/Set	*String*
SingleStepTime	Sets or gets the step time of the offline simulation application. To set or get the step time of the offline simulation application, you need exclusive access to it.	Get/Set	*String*
StopTime	Sets or gets the stop time of the offline simulation application. To set or get the stop time of the offline simulation application, you need exclusive access to it.	Get/Set	*String*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmRegisteredVEOSPlatform (refer to RegisteredVEOSPlatform / IPmRegisteredVEOSPlatform <<Interface>> on page 141)

X

Where to go from here

Information in this section

XILAPIMAPort / IPmXILAPIMAPort <<Interface>>

Description

IPmRegistreredXILAPIMAPortPlatformInternal Interface

///

Properties

The element has the following properties:

Name	Description	Get/Set	Type
ConfigurationFile	Returns the full path to the XIL API model access port configuration file currently configured.	Get	*String*
Implementation	Property for getting the XIL API model access port implementation.	Get	IPmXILAPIMAPortImplementation (refer to XILAPIMAPortImplementation / IPmXILAPIMAPortImplementation <<Interface>> on page 168)
State	Model access port state.	Get	XILAPIMAPortState (refer to XILAPIMAPortState <<Enumeration>> on page 170)

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Disconnect	Disconnects the model access port.	None	None
LoadAndConfigure	Load mode access port configuration and configure the model access port.	▪ *<String>* `ConfigurationPath`: Specifies the file path to the model access port configuration to be configured. ▪ *<Boolean>* `ForceConfiguration`: Specifies to force configuration of the model access port.	None
StartSimulation	Starts the simulation.	None	None
StopSimulation	Stops the simulation.	None	None

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmRegisteredXILAPIMAPortPlatform (refer to RegisteredXILAPIMAPortPlatform / IPmRegisteredXILAPIMAPortPlatform <<Interface>> on page 143)

XILAPIMAPortImplementation / IPmXILAPIMAPortImplementation <<Interface>>

Description

IPmXILAPIMAPortImplementation interface (XIL API device).

Properties

The element has the following properties:

Name	Description	Get/Set	Type
ProductName	Returns the product name of the XIL API model access port implementation.	Get	*String*
ProductVersion	Returns the product version of the XIL API model access port implementation.	Get	*String*
VendorName	Returns the vendor name of the XIL API model access port implementation.	Get	*String*
Version	Returns the XIL API version of the XIL API model access port implementation.	Get	*String*

Methods

The element has no methods.

Returned by

The element is returned by properties or methods of the following elements:

- IPmXILAPIMAPort (refer to XILAPIMAPort / IPmXILAPIMAPort <<Interface>> on page 167)
- IPmXILAPIMAPortImplementations (refer to XILAPIMAPortImplementations / IPmXILAPIMAPortImplementations <<Collection>> on page 169)
- IPmXILAPIMAPortRegisterInfo (refer to XILAPIMAPortRegisterInfo / IPmXILAPIMAPortRegisterInfo <<Interface>> on page 170)

XILAPIMAPortImplementations / IPmXILAPIMAPortImplementations <<Collection>>

Description

IPmXILAPIMAPortImplementations interface (XIL API device).

Properties

The element has the following properties:

Name	Description	Get/Set	Type
Count	Returns the number of XIL model access port implementations in the collection.	Get	*Signed 32 Bit Integer*

Methods

The element has the following methods:

Name	Description	Parameter[1)]	Returns
Item	Returns the XIL model access port implementation from the collection specified by the index.	▪ *<Signed 32 Bit Integer>* `Index`: Index of XIL model access port implemenation.	Returns the requested XIL model access port, if it exists. Otherwise, returns null. ▪ IPmXILAPIMAPortImplementation (refer to XILAPIMAPortImplementation / IPmXILAPIMAPortImplementation <<Interface>> on page 168)

1) <Type> `Name`: Description

Returned by

The element is returned by properties or methods of the following elements:

- IPmXILAPIMAPortRegisterInfo (refer to XILAPIMAPortRegisterInfo / IPmXILAPIMAPortRegisterInfo <<Interface>> on page 170)

XILAPIMAPortRegisterInfo / IPmXILAPIMAPortRegisterInfo <<Interface>>

Description

IPmXILAPIMAPortRegisterInfo interface (XIL API device).

Properties

The element has the following properties:

Name	Description	Get/Set	Type
AvailableMAPortImplementations	Returns the available XIL model access port implementations.	Get	IPmXILAPIMAPortImplementations (refer to XILAPIMAPortImplementations / IPmXILAPIMAPortImplementations <<Collection>> on page 169)
MAPortImplementation	Property to specify the XIL model access port implementation that is used for assignment.	Get/Set	IPmXILAPIMAPortImplementation (refer to XILAPIMAPortImplementation / IPmXILAPIMAPortImplementation <<Interface>> on page 168)
Type	Returns the type of the platform.	Get	PlatformType (refer to PlatformType <<Enumeration>> on page 107)

Methods

The element has no methods.

XILAPIMAPortState <<Enumeration>>

Description

XIL API State Enumeration

Describes the states of the XIL API.

Enumeration values

The enumeration has the following values:

Name	Description	Value
Disconnected	XIL API is in the following state: disconnected.	0
SimulationStarted	XIL API is in the following state: simulation started.	1
SimulationStopped	XIL API is in the following state: simulation stopped.	2

Returned by

The element is returned by properties or methods of the following elements:

- IPmXILAPIMAPort (refer to XILAPIMAPort / IPmXILAPIMAPort <<Interface>> on page 167)