

dSPACE Installation Manager

API Reference

Release 2020-B – November 2020

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

This publication contains proprietary information that is protected by copyright. All rights are reserved. The publication may be printed for personal or internal use provided all the proprietary markings are retained on all printed copies. In all other cases, the publication must not be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of dSPACE GmbH.

© 2015 - 2020 by:
dSPACE GmbH
Rathenaustraße 26
33102 Paderborn
Germany

This publication and the contents hereof are subject to change without notice.

AUTERA, ConfigurationDesk, ControlDesk, MicroAutoBox, MicroLabBox, SCALEXIO, SIMPHERA, SYNECT, SystemDesk, TargetLink and VEOS are registered trademarks of dSPACE GmbH in the United States or other countries, or both. Other brand names or product names are trademarks or registered trademarks of their respective companies or organizations.

Contents

About This Reference	5
Introduction	7
Overview of the API.....	7
Features of the API and Important Notes.....	9
Related Documents.....	11
Examples for Automating dSPACE Installation Manager	13
Examples for Using Windows PowerShell.....	13
Examples for Using Python.....	19
Namespaces	21
dSPACE.InstallationManager.API Namespace.....	21
dSPACE.InstallationManager.API.Internal Namespace.....	22
Classes	25
ConfigurationChange Class.....	26
ConfigurationInfo Class.....	27
Configuration Class.....	29
Diagnostics Class.....	29
Extension Class.....	35
Installation Class.....	36
InstallationConfig Class.....	40
InstallationManagerException Class.....	46
MaintenanceExtension Class.....	46
Matlab Class.....	48
MatlabIntegration Class.....	50
MatlabIntegrationConfig Class.....	53
MatlabIntegrationsConfig Class.....	55
Product Class.....	66
Enumerations	69
InstallationType Enumeration.....	69

MaintenanceExtensionKind Enumeration.....	70
MatlabVersionSupport Enumeration.....	71
ReportContent Enumeration.....	72

Index	73
-------	----









About This Reference

Contents

This Reference introduces you to the API of dSPACE Installation Manager.

Symbols

dSPACE user documentation uses the following symbols:

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

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.

Introduction

Where to go from here

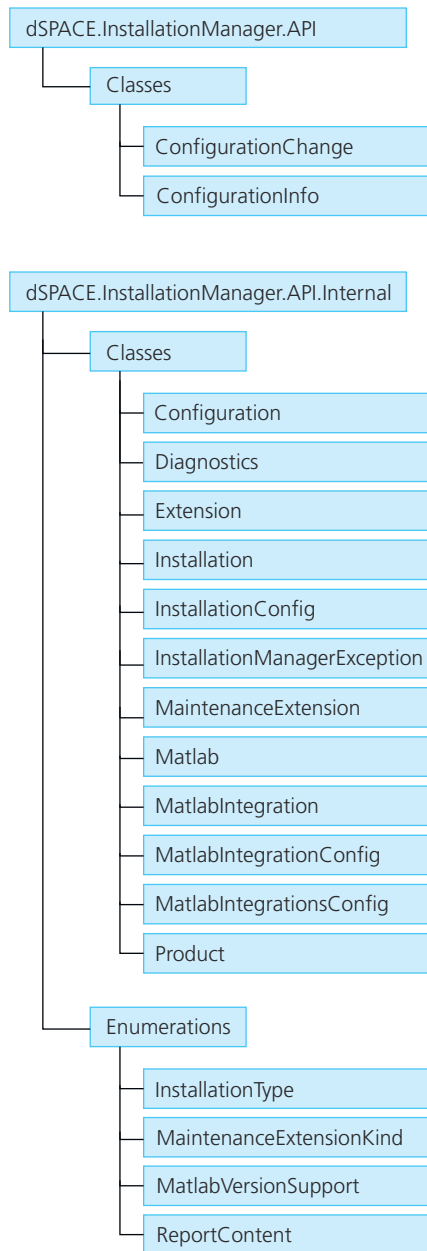
Information in this section

Overview of the API.....	7
Features of the API and Important Notes.....	9
Related Documents.....	11

Overview of the API

API objects dependency

The illustration shows the dependencies of the objects that are used in the API to handle automation tasks.



Features of the API and Important Notes

Features of the API

The API Version 2.0.0.0 of the dSPACE Installation Manager provides the following features:

- Getting information on the dSPACE installations on your host PC.
- Getting information on MATLAB installations (on your host PC and on a connected network) and their integration into dSPACE software.
- Activation and deactivation of dSPACE installations
- Decryption of software parts that are installed in encrypted archives.

Note

This feature is only supported when the API runs in a 64-bit process.

- Adding and removing a MATLAB integration (integration of a specific MATLAB installation into dSPACE software) to/from the dSPACE Installation Manager.
- Modifying an existing MATLAB integration, for example, setting a specific MATLAB integration as the preferred one.
- Collects information about the dSPACE software environment on your PC and the problem you may have and saves it to a ZIP file. This you can send to dSPACE Support for diagnostic purposes.

Software environment

You can use the API of dSPACE Installation Manager in combination with 32-bit and 64-bit software programs.

Note

- As of dSPACE Release 2017-B, new features of the API of dSPACE Installation Manager are only supported when the API runs in a 64-bit process.
- As of dSPACE Release 2021-B, the API of dSPACE Installation Manager will only run in 64-bit processes.

Accessing the API

You can access the API Version 2.0.0.0 of dSPACE Installation Manager by using the following scripting environments and programming languages:

- Windows PowerShell 4.0
Windows PowerShell 4.0 is part of the Windows Management Framework 4.0. This software package is free of charge and can be downloaded from <http://www.microsoft.com>.
- Python (Python scripts running in an external interpreter, for example, PythonWin)

You can find PythonWin in the Windows Start menu (Start — Python 3.6 — PythonWin) after installing of the dSPACE software.

Note

Python does not support modifying an existing MATLAB integration. As a workaround, you can use IronPython. IronPython is an open-source implementation of the Python programming language which is tightly integrated with the .NET Framework

In the IronPython console you can, for example, start a Python script as follows:

```
>>> execfile(r"c:\MyScript.py")
```

- Batch file (starting a PowerShell script)

Enabling scripting with Windows PowerShell By default, the Windows operating system does not support scripting with Windows PowerShell. You can enable scripting by executing the following steps:

1. From Start — Windows PowerShell, select Windows PowerShell.
2. On the Windows PowerShell entry, right-click Run as administrator.
3. In the PowerShell window enter the following command: **Set-ExecutionPolicy RemoteSigned**.
4. Confirm the Execution Policy Change question with Y (= yes).
Scripting with Windows PowerShell is enabled.

Activating/deactivating dSPACE installations and decrypting via Python scripts

As of dSPACE Release 2019-A, the API supports activating/deactivating dSPACE installations and decrypting archives of the installed dSPACE software via Python scripts.




Note

You have to restart all running processes that use **python.exe**, **PythonWin.exe**, or **pythonw.exe** after the action performed with dSPACE Installation Manager has been finished. If you do not restart the running processes, the previous dSPACE Python enhancements still remain active.

Related Documents

Overview

Below is an overview of documents that you are also recommended to read when working with the API of dSPACE Installation Manager.

Document	Contents
 Managing dSPACE Software Installations	Shows you how to manage all the dSPACE installations on your host PC with dSPACE Installation Manager. Managing means, for example, activating or deactivating installations, decrypting the encrypted archives of the installed software and connecting dSPACE software installations to MATLAB software.
 Providing Diagnostic Information	Shows how to collect diagnostic information via dSPACE Installation Manager and to provide this information to dSPACE Support.
 dSPACE Installation Manager Glossary	Briefly explains the most important expressions and naming conventions used in the dSPACE Installation Manager documentation.

Examples for Automating dSPACE Installation Manager

Where to go from here

Information in this section

Examples for Using Windows PowerShell.....	13
Examples for Using Python.....	19

Examples for Using Windows PowerShell

Getting and listing all dSPACE installations on your host PC

```
#Load assembly dSPACE.InstallationManager.API
$assembly=[System.Reflection.Assembly]::LoadWithPartialName("dSPACE.InstallationManager.API")

#Create ConfigurationInfo object for later use
$Info=New-Object dSPACE.InstallationManager.API.ConfigurationInfo

$Info.Installations
```

Getting and listing all ControlDesk installations on your host PC

```
#Load assembly dSPACE.InstallationManager.API
$assembly=[System.Reflection.Assembly]::LoadWithPartialName("dSPACE.InstallationManager.API")

#Create InstallationInfo object for later use
$installationInfo=New-Object dSPACE.InstallationManager.API.ConfigurationInfo

$installationInfo.Installations | Where-Object {$_.TypeName -eq
[dSPACE.InstallationManager.API.Internal.InstallationType]::ControlDesk}
```

Getting and listing all MATLAB installations on your host PC and on the network that are added manually

```
#This demo script must be used with version 4.0 of Windows PowerShell.

#Load assembly dSPACE.InstallationManager.API

[System.Reflection.Assembly]::LoadWithPartialName("dSPACE.InstallationManager.API")

#Create ConfigurationInfo object for later use
$Info=New-Object dSPACE.InstallationManager.API.ConfigurationInfo

$Info.MatlabInstallations
```

Getting and listing all MATLAB installations that are integrated in a specific dSPACE installation

```
#Load assembly dSPACE.InstallationManager.API
$assembly=[System.Reflection.Assembly]::LoadWithPartialName("dSPACE.InstallationManager.API")

#Create ConfigurationInfo object for later use
$Info=New-Object dSPACE.InstallationManager.API.ConfigurationInfo

$Info.MatlabIntegrations
```

Getting and listing the preferred MATLAB installation

```
#Load assembly dSPACE.InstallationManager.API
[System.Reflection.Assembly]::LoadWithPartialName("dSPACE.InstallationManager.API")

#Create ConfigurationInfo object for later use
$Info=New-Object dSPACE.InstallationManager.API.ConfigurationInfo

$Info.MatlabIntegrations | Where-Object {$_.IsPreferred}
```

Getting and listing all patches and hotfixes for a ControlDesk installation

```
#Load assembly dSPACE.InstallationManager.API
[System.Reflection.Assembly]::LoadWithPartialName("dSPACE.InstallationManager.API")

#Create InstallationInfo object for later use
$installationInfo=New-Object dSPACE.InstallationManager.API.ConfigurationInfo

#Get ControlDesk installations
$ControlDeskInstallations = $installationInfo.Installations | Where-Object {$_.TypeName -eq
[dSPACE.InstallationManager.API.Internal.InstallationType]::ControlDesk}

#Iterate and print out hotfixes, patches and service packs
foreach ($Installation in $ControlDeskInstallations)
{
    foreach ($Product in $Installation.Products)
    {
        $Product.MaintenanceExtensions;
    }
}
```

Activating a specific dSPACE installation

```
#This demo script must be used with version 4.0 of Windows PowerShell.

#Load assembly dSPACE.InstallationManager.API
[System.Reflection.Assembly]::LoadWithPartialName("dSPACE.InstallationManager.API")

#Create ConfigurationChange object (or get the already created instance) for later use. (Only one item can be created in
one process.)
$change=[dSPACE.InstallationManager.API.ConfigurationChange]::Instance()

$rootPath = "d:\Program Files\dSPACE RCPHIL 2015-B"
$installation = $change.Installations | Where-Object {$_.RootPath.Equals($rootPath)}
$shouldBEPREFERREDMatlab = $false

try
{
    "Start activating " + $installation.RootPath
    $installation.Activate()
    "End activating " + $installation.RootPath
}
catch
{
    "An Error occurred while activating " + $installation.RootPath | Write-Host -ForegroundColor Red
    $_.Exception.Message | Write-Host -ForegroundColor Red
}
```

Adding all possible MATLAB integrations

```
#This demo script is intended to be used with version 4.0 of the Windows PowerShell (x86).

#Load assembly dSPACE.InstallationManager.API
[System.Reflection.Assembly]::LoadWithPartialName("dSPACE.InstallationManager.API")

#Create ConfigurationChange object (or get the already created instance) for later use. (Only one item can be created in
one process.)
$change=[dSPACE.InstallationManager.API.ConfigurationChange]::Instance()

$shouldBEPREFERREDMatlab = $false

foreach($mi in $change.MatlabInstallations)
{
    foreach($inst in $change.Installations)
    {
        if($change.MatlabIntegrations.CanAdd($inst, $mi))
        {
            try
            {
                "`nStart adding integration of " + $mi.RootPath + " and " + $inst.RootPath
                $change.MatlabIntegrations.Add($inst, $mi, $shouldBEPREFERREDMatlab)
                "End adding integration of " + $mi.RootPath + " and " + $inst.RootPath
            }
            catch
            {
                "Error occurred while modifying Matlab integration of " + $mi.RootPath + " with " + $inst.RootPath |
Write-Host -ForegroundColor Red
                $_.Exception.Message | Write-Host -ForegroundColor Red
            }
        }
        else
        {
            "Matlab integration of " + $mi.RootPath + " with " + $inst.RootPath + " can currently not be added."
        }
    }
}
}
```


Decrypting encrypted archives for a specific installation

```
#This demo script is intended to be used with version 4.0 of the Windows PowerShell.

#Load assembly dSPACE.InstallationManager.API
[System.Reflection.Assembly]::LoadWithPartialName("dSPACE.InstallationManager.API")

#Create ConfigurationChange object (or get the already created instance) for later use. (Only one item can be created in
one process.)

$change=[dSPACE.InstallationManager.API.ConfigurationChange]::Instance()

$rootPath = "d:\Program Files\dSPACE RCPHIL 2016-B"

$installation = $change.Installations | Where-Object {$_.RootPath.Equals($rootPath)}

try
{
    "`nStart decrypting of encrypted parts for installation " + $installation.RootPath
    $Succeeded = $installation.DecryptEncryptedParts($TRUE)
    if($Succeeded -eq $FALSE)
    {
        "Could not decrypt all encrypted parts." | Write-Host -ForegroundColor Red
    }
    "End decrypting of encrypted parts of " + $installation.RootPath
}
catch
{
    "An Error occurred while decrypting encrypted parts for installation " + $installation.RootPath | Write-Host -
ForegroundColor Red
    $_.Exception.Message | Write-Host -ForegroundColor Red
}
```

Generating diagnostic information

```
#This demo script is intended to be used with version 4.0 of the Windows PowerShell.

#Load assembly dSPACE.InstallationManager.API
[System.Reflection.Assembly]::LoadWithPartialName("dSPACE.InstallationManager.API")

function PrintException( [Exception] $ex)
{
    $ex |format-list -Force
    if ($ex.InnerException)
    {
        PrintException $ex.InnerException
    }
}

try
{
    #Create ConfigurationInfo object for later use
    $info=New-Object dSPACE.InstallationManager.API.ConfigurationInfo

    #Generate diagnostics report with basic information. Write the result to file DiagnosticsReport.zip and allow
    overwriting of an already existing file.
    $info.Diagnostics.GenerateReport("MySubject", "MyDescription",
    [dSPACE.InstallationManager.API.Internal.ReportContent]::Basic, "c:\temp\DiagnosticsReport.zip", $true)

    #Generate diagnostics report with basic information but with some file attachments and with information for some
    specific installations.
    $attachments = "c:\temp\AttachmentFile1.txt", "c:\temp\AttachmentFile2.txt"
    $installations = "d:\Program Files\dSPACE AutomationDesk 5.6", "d:\Program Files\dSPACE ControlDesk 6.3"
    #$info.Diagnostics.GenerateReport("MySubject", "MyDescription",
    [dSPACE.InstallationManager.API.Internal.ReportContent]::Basic, $installations, $attachments,
    "c:\temp\DiagnosticsReport.zip", $true)

    #Generate diagnostics report with all available information.
    #$info.Diagnostics.GenerateReport("MySubject", "MyDescription",
    [dSPACE.InstallationManager.API.Internal.ReportContent]::All, "c:\temp\DiagnosticsReport.zip", $true)
}
catch
{
    PrintException $_.Exception
}
```

Examples for Using Python

Getting and listing all dSPACE installations

```
import clr
clr.AddReference("dSPACE.InstallationManager.API, Version=2.0.0.0, Culture=neutral, PublicKeyToken=f9604847d8afbfb")
import dSPACE.InstallationManager.API
InstallationInfo = dSPACE.InstallationManager.API.ConfigurationInfo()

for Installation in InstallationInfo.Installations:
    print ("%50s RootPath : %s" % (Installation.DisplayName, Installation.RootPath))
```

Getting and listing all MATLAB installations

```
import clr
clr.AddReference("dSPACE.InstallationManager.API, Version=2.0.0.0, Culture=neutral, PublicKeyToken=f9604847d8afbfb")
import dSPACE.InstallationManager.API
MATLABInstallationInfo = dSPACE.InstallationManager.API.MATLABInstallationInfo()

for MATLABInstallation in MATLABInstallationInfo.MATLABInstallations:
    print ("%50s RootPath : %s" % (MATLABInstallation.DisplayName, MATLABInstallation.RootPath))
```

Generating diagnostic report

```
import clr
clr.AddReference("dSPACE.InstallationManager.API, Version=2.0.0.0, Culture=neutral, PublicKeyToken=f9604847d8afbfb")
import dSPACE.InstallationManager.API
Info = dSPACE.InstallationManager.API.ConfigurationInfo()

Info.Diagnostics.GenerateReport('MySubject', 'MyDescription',
dSPACE.InstallationManager.API.Internal.ReportContent.Basic, r'c:\temp\DiagnosticsReport.zip', False)
```


Namespaces

Where to go from here

Information in this section

dSPACE.InstallationManager.API Namespace.....	21
Contains types that users can create themselves.	
dSPACE.InstallationManager.API.Internal Namespace.....	22
Contains types that users cannot create themselves.	

dSPACE.InstallationManager.API Namespace

Purpose

The dSPACE.InstallationManager.API namespace contains types that users can create themselves.

Classes

Class	Description
ConfigurationChange	Changes settings of dSPACE installations, for example, modifies the integration of dSPACE software into MATLAB installations.
ConfigurationInfo	Provides information about available dSPACE installations, installed MATLAB software, and integrations of dSPACE software into MATLAB installations. This information represents the current state of the system. Any later changes will not be considered.

Related topics

References

ConfigurationChange Class.....	26
ConfigurationInfo Class.....	27

dSPACE.InstallationManager.API.Internal Namespace

Purpose

The dSPACE.InstallationManager.API.Internal namespace contains types that users cannot create themselves.

Classes

Class	Description
Configuration	Base class for providing access to dSPACE installations.
Diagnostics	Collects information about the dSPACE software environment on your PC and the problem you may have and saves it to a ZIP file.
Extension	Provides information about an extension (such as an add-on or a solution) for a specific dSPACE product.
Installation	Provides information about a dSPACE installation.
InstallationConfig	Provides information about a dSPACE installation, lets you activate/deactivate an installation and lets you decrypt the encrypted parts of the installed products.
InstallationManagerException	The exception that is thrown when an error occurs in the API of the dSPACE Installation Manager.
MaintenanceExtension	Provides information about a hotfix, a patch or a service pack for a dSPACE product.
Matlab	Provides information about a MATLAB installation.
MatlabIntegration	Represents an integration of a dSPACE installation into a specific MATLAB installation.
MatlabIntegrationConfig	Modifies an existing MATLAB integration (integration of dSPACE software into a specific MATLAB installation).

Class	Description
MatlabIntegrationsConfig	Provides information about integrations into MATLAB and lets you add or remove integrations to/from the dSPACE Installation Manager.
Product	Provides information about a dSPACE product.

Enumerations

Enumeration	Description
InstallationType	The type name of dSPACE installations.
MaintenanceExtensionKind	Represents the kind of a maintenance extension (hotfix, patch or service pack).
MatlabVersionSupport	Indicates whether a dSPACE installation can be integrated into a specific Matlab software version.

Related topics

References

Configuration Class.....	29
Diagnostics Class.....	29
Extension Class.....	35
Installation Class.....	36
InstallationConfig Class.....	40
InstallationManagerException Class.....	46
InstallationType Enumeration.....	69
MaintenanceExtension Class.....	46
MaintenanceExtensionKind Enumeration.....	70
Matlab Class.....	48
MatlabIntegration Class.....	50
MatlabIntegrationConfig Class.....	53
MatlabIntegrationsConfig Class.....	55
MatlabVersionSupport Enumeration.....	71
Product Class.....	66

Classes

Where to go from here

Information in this section

ConfigurationChange Class.....	26
Changes settings of dSPACE installations, for example, modifies the integration of dSPACE software into MATLAB installations.	
ConfigurationInfo Class.....	27
Provides information about available dSPACE installations, installed MATLAB software, and integrations of dSPACE software into MATLAB installations.	
Configuration Class.....	29
Is a base class and provides access to dSPACE installations.	
Diagnostics Class.....	29
Collects information about the dSPACE software environment on your PC and the problem you may have and saves it to a ZIP file.	
Extension Class.....	35
Provides information about an extension (such as an add-on or a solution) for a specific dSPACE product.	
Installation Class.....	36
Provides information about a dSPACE installation.	
InstallationConfig Class.....	40
Provides information about a dSPACE installation, lets you activate/deactivate an installation and lets you decrypt the encrypted parts of the installed products.	
InstallationManagerException Class.....	46
The exception that is thrown when an error occurs in the API of the dSPACE Installation Manager.	
MaintenanceExtension Class.....	46
Provides information about a hotfix, a patch or a service pack for a dSPACE product.	

Matlab Class.....	48
Provides information about a MATLAB installation.	
MatlabIntegration Class.....	50
Represents an integration of a dSPACE installation into a MATLAB installation.	
MatlabIntegrationConfig Class.....	53
Modifies an existing MATLAB integration (integration of dSPACE software into a specific MATLAB installation).	
MatlabIntegrationsConfig Class.....	55
Provides information about MATLAB integrations and lets you add or remove MATLAB integrations to/from the dSPACE Installation Manager.	
Product Class.....	66
Provides information about a dSPACE product.	

ConfigurationChange Class

Purpose Changes settings of dSPACE installations. For example, modifies the integration of dSPACE software into MATLAB installations.

Declaration

C#	<code>public class ConfigurationChange : Configuration</code>
Visual Basic	<code>Public Class ConfigurationChange Inherits Configuration</code>

Related namespace

dSPACE.InstallationManager.API

Methods

The class has the following methods:

Name	Description	Return Value	Type	Declaration	
Instance	Gets the ConfigurationChange instance. Only one item is supported per process.	The only ConfigurationChange instance.	ConfigurationChange	C#	<code>public static ConfigurationChange Instance()</code>
				Visual Basic	<code>Public Shared Function Instance As ConfigurationChange</code>

Properties

The class has the following properties:

Name	Description	Type	Declaration	
Installations	Gets a collection of all dSPACE installations on the local host PC.	ReadOnlyCollection (Of InstallationConfig)	C#	public ReadOnlyCollection<InstallationConfig> Installations { get; }
			Visual Basic	Public ReadOnly Property Installations As ReadOnlyCollection(Of InstallationConfig) Get
MatlabInstallations	Gets a collection of all MATLAB installations (32-bit and 64-bit). The list contains all local installations and the installations on a network that users add manually.	ReadOnlyCollection (Of Matlab)	C#	public ReadOnlyCollection<Matlab> MatlabInstallations { get; }
			Visual Basic	Public ReadOnly Property MatlabInstallations As ReadOnlyCollection(Of Matlab) Get
MatlabIntegrations	Gets a collection of all MATLAB installations with integrated dSPACE software.	MatlabIntegrationsConfig	C#	public MatlabIntegrationsConfig MatlabIntegrations { get; }
			Visual Basic	Public ReadOnly Property MatlabIntegrations As MatlabIntegrationsConfig Get

Exceptions

Name	Condition
InstallationManagerException	Any error that occurs in the API.

Related topics**References**

dSPACE.InstallationManager.API Namespace..... 21

ConfigurationInfo Class

Purpose

Provides information about available dSPACE installations, installed MATLAB software, and integrations of dSPACE software into MATLAB installations.

Declaration	C#	<code>public class ConfigurationInfo : Configuration</code>
	Visual Basic	<code>Public Class ConfigurationInfo Inherits Configuration</code>

Related namespace dSPACE.InstallationManager.API

Description This information represents the current state of the system. Any later changes will not be considered.

Constructors The class has the following constructors:

Name	Description	Declaration	
ConfigurationInfo	Initializes a new instance of the ConfigurationInfo class.	C#	<code>public ConfigurationInfo()</code>
		Visual Basic	<code>Public Sub New</code>

Properties The class has the following properties:

Name	Description	Type	Declaration	
Diagnostics	Gets a diagnostics item that is saved to a specified ZIP file.	Diagnostics	C#	<code>public Diagnostics Diagnostics { get; }</code>
			Visual Basic	<code>Public ReadOnly Property Diagnostics As Diagnostics Get</code>
Installations	Gets a collection of all dSPACE installations on the local host PC.	ReadOnlyCollection (Installation)	C#	<code>public ReadOnlyCollection<Installation> Installations { get; }</code>
			Visual Basic	<code>Public ReadOnly Property Installations As ReadOnlyCollection(Of Installation) Get</code>
MatlabInstallations	Gets a collection of all MATLAB installations (32-bit and 64-bit). The list contains all local installations and the installations on a network that users add manually.	ReadOnlyCollection (Matlab)	C#	<code>public ReadOnlyCollection<Matlab> MatlabInstallations { get; }</code>
			Visual Basic	<code>Public ReadOnly Property MatlabInstallations As ReadOnlyCollection(Of Matlab) Get</code>
MatlabIntegrations	Gets a collection of all MATLAB installations with integrated dSPACE software.	ReadOnlyCollection (MatlabIntegration)	C#	<code>public ReadOnlyCollection<MatlabIntegration> MatlabIntegrations { get; }</code>
			Visual Basic	<code>Public ReadOnly Property MatlabIntegrations As ReadOnlyCollection(Of MatlabIntegration) Get</code>

Exceptions

Name	Condition
InstallationManagerException	Any error that occurs in the API.

Related topics**References**

[dSPACE.InstallationManager.API Namespace..... 21](#)

Configuration Class

Purpose

Is a base class and provides access to dSPACE installations.

Declaration

C#	<code>public class Configuration</code>
Visual Basic	<code>Public Class Configuration</code>

Related namespace

dSPACE.InstallationManager.API.Internal

Related topics**References**

[dSPACE.InstallationManager.API.Internal Namespace..... 22](#)

Diagnostics Class

Purpose

Collects information about the dSPACE software environment on your PC and the problem you may have and saves it to a ZIP file.

Declaration

C#	<code>public class Diagnostics</code>
Visual Basic	<code>Public Class Diagnostics</code>

Related namespace

dSPACE.InstallationManager.API.Internal

Methods

The class has the following methods:

Name	Characteristics		
Diagnostics.GenerateReport Method (String, String, ReportContent, String, Boolean)	Description		Declaration
	Collects diagnostic information and saves it to a ZIP file.		C#
			public void GenerateReport(string subject, string problemDescription, ReportContent reportContent, string reportFileFullPath, bool overwriteExisting)
			Visual Basic
			Public Sub GenerateReport (subject As String, problemDescription As String, reportContent As ReportContent, reportFileFullPath As String, overwriteExisting As Boolean)
	Parameter	Description	Type
	subject	The subject of the problem that leads to the support request.	System.String
	problemdescription	Information describing the problem.	System.String
	reportcontent	Indicates which general content will be collected and saved in the ZIP file. Refer to ReportContent Enumeration on page 72.	dSPACE.InstallationManager.API.Internal.ReportContent
	reportFileFullPath	Full path and name of the ZIP file that will be created. Example: "c:\Diagnostics.zip".	System.String
	overwriteExisting	<ul style="list-style-type: none"> True: An existing diagnostic ZIP file will be overwritten. False: An existing diagnostic ZIP file will not be overwritten. 	System.Boolean

Name	Characteristics		
Diagnostics.GenerateReport Method (String, String, ReportContent, String(), String, Boolean)	Description		Declaration
	Collects diagnostic information and saves it to a ZIP file. Additionally, you can attach your own files to be saved to the ZIP file.		C#
			public void GenerateReport(string subject, string problemDescription, ReportContent reportContent, string[] attachments, string reportFileFullPath, bool overwriteExisting)
			Visual Basic
			Public Sub GenerateReport (subject As String, problemDescription As String, reportContent As ReportContent, attachments As String(), reportFileFullPath As String, overwriteExisting As Boolean)
	Parameter	Description	Type
	subject	The subject of the problem that leads to the support request.	System.String
	problemdescription	Information describing the problem.	System.String
	reportcontent	Indicates which content will be collected and saved in the ZIP file. Refer to ReportContent Enumeration on page 72.	dSPACE.InstallationManager.API.Internal.ReportContent
	attachments	Full paths of additional files that will be saved in the ZIP file. Can be null. Strings that do not represent an existing file will be ignored.	System.String
	reportFileFullPath	Full path and name of the ZIP file that will be created. Example: "c:\Diagnostics.zip".	System.String

Name	Characteristics		
	overwriteExisting	<ul style="list-style-type: none">▪ True: An existing diagnostic ZIP file will be overwritten.▪ False: An existing diagnostic ZIP file will not be overwritten.	System.Boolean

Name	Characteristics		
Diagnostics.GenerateReport Method (String, String, ReportContent, String(), String(), String, Boolean)	Description	Declaration	
		C#	<pre>public void GenerateReport(string subject, string problemDescription, ReportContent reportContent, string[] installations, string[] attachments, string reportFileFullPath, bool overwriteExisting)</pre>
		Visual Basic	<pre>Public Sub GenerateReport (subject As String, problemDescription As String, reportContent As ReportContent, installations As String(), attachments As String(), reportFileFullPath As String, overwriteExisting As Boolean)</pre>
	Parameter	Description	Type
	subject	The subject of the problem that leads to a support request.	System.String
	problemdescription	Information describing the problem.	System.String
	reportcontent	Indicates which general content will be collected and saved in the ZIP file. Refer to ReportContent Enumeration on page 72.	dSPACE.InstallationManager.API.Internal.ReportContent

Name	Characteristics		
	installations	The root directories of the software installations that will be included for collecting diagnostic information. Can be null. Strings not representing an existing installation will be ignored. Note that this parameter will be ignored if the reportContent parameter is set to ReportContent.All.	System.String
	attachments	Full paths of additional files that will be saved in the ZIP file. Can be null. Strings that do not represent an existing file will be ignored.	System.String
	reportFileFullPath	Full path and name of the ZIP file that will be created. Example: "c:\Diagnostics.zip".	System.String
	overwriteExisting	<ul style="list-style-type: none"> ▪ True: An existing diagnostic ZIP file will be overwritten. ▪ False: An existing diagnostic ZIP file will not be overwritten. 	System.Boolean

Exceptions

Name	Condition
InstallationManagerException	Any error that occurs in the API.

Related topics

References

[dSPACE.InstallationManager.API.Internal Namespace..... 22](#)

Extension Class

Purpose Provides information about an extension (such as an add-on or a solution) for a specific dSPACE product.

Declaration	C#	<code>public class Extension</code>
	Visual Basic	<code>Public Class Extension</code>

Related namespace dSPACE.InstallationManager.API.Internal

Methods The class has the following methods:

Name	Description	Return Value	Type	Declaration	
ToString	Returns the display name to represent this instance. Overrides <code>Object.ToString()</code> .	DisplayName	String	C#	<code>public override string ToString()</code>
				Visual Basic	<code>Public Overrides Function ToString As String</code>

Properties The class has the following properties:

Name	Description	Type	Declaration	
Description	Gets the description.	String	C#	<code>public string Description { get; }</code>
			Visual Basic	<code>Public ReadOnly Property Description As String Get</code>
DisplayName	Gets the display name.	String	C#	<code>public string DisplayName { get; }</code>
			Visual Basic	<code>Public ReadOnly Property DisplayName As String Get</code>
RootPath	Gets the root path of the installation of this extension.	String	C#	<code>public string RootPath { get; }</code>
			Visual Basic	<code>Public ReadOnly Property RootPath As String Get</code>
UniqueName	Gets the unique name that identifies this extension.	String	C#	<code>public string UniqueName { get; }</code>
			Visual Basic	<code>Public ReadOnly Property UniqueName As String Get</code>
Version	Gets the version.	String	C#	<code>public string Version { get; }</code>
			Visual Basic	<code>Public ReadOnly Property Version As String Get</code>

Exceptions

Name	Condition
InstallationManagerException	Any error that occurs in the API.

Related topics**References**

[dSPACE.InstallationManager.API.Internal Namespace..... 22](#)

Installation Class

Purpose

Provides information about a dSPACE installation.

Declaration

C#	<code>public class Installation : IEquatable<Installation></code>
Visual Basic	<code>Public Class Installation Implements IEquatable(Of Installation)</code>

Related namespace

dSPACE.InstallationManager.API.Internal

Methods

The class has the following methods:

Name	Characteristics				
Equals(Object)	Description	Return Value	Type	Declaration	
	Indicates whether the specified object is equal to this instance. Overrides <code>Object.Equals(Object)</code> .	<ul style="list-style-type: none"> True: The specified object is equal to this instance. False: The specified object is unequal to this instance. 	Boolean	C#	<code>public override bool Equals(Object obj)</code>
				Visual Basic	<code>Public Overrides Function Equals (obj As Object) As Boolean</code>
	Parameter	Description	Type		
	obj	The object to compare with this instance.	System.Object		

Name	Characteristics				
Equals(Installation)	Description	Return Value	Type	Declaration	
	Indicates whether the specified object is equal to this object. Overrides <code>Object.Equals(Object)</code> .	<ul style="list-style-type: none">▪ True: The specified object is equal to this object.▪ False: The specified object is unequal to this object.	Boolean	C#	<code>public bool Equals(Installation other)</code>
				Visual Basic	<code>Public Function Equals (other As Installation) As Boolean</code>
	Parameter	Description	Type		
	Other	The object to compare with this object.	dSPACE.InstallationManager.API.Internal.Installation		
	Implements				
	<code>IEquatable<Of T> .Equals(T)</code>				
GetHashCode	Description	Return Value	Type	Declaration	
	Returns a hash code for this instance. Overrides <code>Object.GetHashCode()</code> .	A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.	Int32	C#	<code>public override int GetHashCode()</code>
				Visual Basic	<code>Public Overrides Function GetHashCode As Integer</code>
ToString	Description	Return Value	Type	Declaration	
	Returns the display name to represent this instance. Overrides <code>Object.ToString()</code> .	DisplayName	String	C#	<code>public override string ToString()</code>
				Visual Basic	<code>Public Overrides Function ToString As String</code>

Properties

The class has the following properties.

Name	Description	Type	Declaration	
Description	Gets the description of the installation.	String	C#	public string Description { get; }
			Visual Basic	Public ReadOnly Property Description As String Get

Name	Description	Type	Declaration	
DisplayName	Gets the display name of the installation.	String	C#	public string DisplayName { get; }
			Visual Basic	Public ReadOnly Property DisplayName As String Get
Extensions	Gets a list of extensions that are provided by this installation.	IEnumerable (Extension)	C#	public IEnumerable<Extension> Extensions { get; }
			Visual Basic	Public ReadOnly Property Extensions As IEnumerable(Of Extension) Get
InstallationDate	Gets information about when the installation was installed.	String	C#	public string InstallationDate { get; }
			Visual Basic	Public ReadOnly Property InstallationDate As String Get
InstallationUser	Gets the user who installed the installation.	String	C#	public string InstallationUser { get; }
			Visual Basic	Public ReadOnly Property InstallationUser As String Get
Is64Bit	Gets a value indicating whether this is a 64-bit installation.	Boolean	C#	public bool Is64Bit { get; }
			Visual Basic	Public ReadOnly Property Is64Bit As Boolean Get
IsActive	Gets a value indicating whether the installation is the active installation. The meaning of active depends on the type of the installation (= product).	Boolean	C#	public bool IsActive { get; }
			Visual Basic	Public ReadOnly Property IsActive As Boolean Get
IsPrerelease	Gets a value indicating whether this installation is a pre-released version.	Boolean	C#	public bool IsPrerelease { get; }
			Visual Basic	Public ReadOnly Property IsPrerelease As Boolean Get

Name	Description	Type	Declaration	
LeastSupportedMatlabVersion32Bit	Gets the oldest supported MATLAB version (32-bit) of this installation. MATLAB versions that are not older than this version can be used with this installation although this is not officially supported.	String	C#	public string LeastSupportedMatlabVersion32Bit { get; }
			Visual Basic	Public ReadOnly Property LeastSupportedMatlabVersion32Bit As String Get
LeastSupportedMatlabVersion64Bit	Gets the oldest supported MATLAB version (64-bit) of this installation. MATLAB versions that are not older than this version can be used with this installation although this is not officially supported.	String	C#	public string LeastSupportedMatlabVersion64Bit { get; }
			Visual Basic	Public ReadOnly Property LeastSupportedMatlabVersion64Bit As String Get
Products	Gets a list of products that belong to this installation.	IEnumerable (Product)	C#	public IEnumerable<Product> Products { get; }
			Visual Basic	Public ReadOnly Property Products As IEnumerable(Of Product) Get
Release	Gets the dSPACE release that the installation was (first) published with.	String	C#	public string Release { get; }
			Visual Basic	Public ReadOnly Property Release As String Get
RootPath	Gets the root directory of the installation.	String	C#	public string RootPath { get; }
			Visual Basic	Public ReadOnly Property RootPath As String Get
SupportedMatlabVersions32Bit	Gets a list of MATLAB versions (32-bit) that are officially supported by this installation.	IEnumerable (String)	C#	public IEnumerable<string> SupportedMatlabVersions32Bit { get; }
			Visual Basic	Public ReadOnly Property SupportedMatlabVersions32Bit As IEnumerable(Of String) Get

Name	Description	Type	Declaration	
SupportedMatlabVersions64Bit	Gets a list of MATLAB versions (64-bit) that are officially supported by this installation.	IEnumerable (String)	C#	<code>public IEnumerable<string> SupportedMatlabVersions64Bit { get; }</code>
			Visual Basic	<code>Public ReadOnly Property SupportedMatlabVersions64Bit As IEnumerable(Of String) Get</code>
SupportsActivation	Gets a value indicating whether the installation can be activated and deactivated by the user.	Boolean	C#	<code>public bool SupportsActivation { get; }</code>
			Visual Basic	<code>Public ReadOnly Property SupportsActivation As Boolean Get</code>
TypeName	Gets the type name of the installation.	InstallationType	C#	<code>public InstallationType TypeName { get; }</code>
			Visual Basic	<code>Public ReadOnly Property TypeName As InstallationType Get</code>
Version	Gets the version of the installation.	String	C#	<code>public string Version { get; }</code>
			Visual Basic	<code>Public ReadOnly Property Version As String Get</code>

Exceptions

Name	Condition
InstallationManagerException	Any error that occurs in the API.

Related topics

References

dSPACE.InstallationManager.API.Internal Namespace.....	22
InstallationType Enumeration.....	69

InstallationConfig Class

Purpose

Provides information about a dSPACE installation, lets you activate/deactivate an installation and lets you decrypt the encrypted parts of the installed products.

Declaration

C#	<code>public class InstallationConfig : Installation</code>
Visual Basic	<code>Public Class InstallationConfig Inherits Installation</code>

Related namespace

dSPACE.InstallationManager.API.Internal

Constructors

The class has the following constructors:

Name	Characteristics		
InstallationConfig	Description		Declaration
	Initializes a new instance of the InstallationConfig class.		C#
			<code>public InstallationConfig(ISESSIONManager relevantSessionManager, IInstallationSet installationSet)</code>
			Visual Basic
			<code>Public Sub New (relevantSessionManager As ISESSIONManager, installationSet As IInstallationSet)</code>
	Parameter	Description	Type
	relevantSessionManager	The relevant version of the dSPACE Installation Manager.	ISessionManager
	installationSet	The relevant installation set.	InstallationSet

Methods

The class has the following methods:

Name	Description	Return Value	Type	Declaration	
Activate	Activates this installation. As a precondition, the installation must support being activated and deactivated.	–	–	C#	<code>public void Activate()</code>
				Visual Basic	<code>Public Sub Activate</code>
Deactivate	Deactivates this installation. As a precondition the installation must support being activated and deactivated.	–	–	C#	<code>public void Deactivate()</code>
				Visual Basic	<code>Public Sub Deactivate</code>

Name	Description	Return Value	Type	Declaration	
DecryptEncryptedParts	Description	Return Value	Type	Declaration	
	Decrypts the encrypted parts of the installed products. This method is only supported when the API runs in a 64-bit process.	<ul style="list-style-type: none"> True: All items have been successfully decrypted. False: At least one item could not be decrypted. 	Boolean	C#	public bool DecryptEncryptedParts(bool force)
				Visual Basic	Public Function DecryptEncryptedParts (force As Boolean) As Boolean
	Parameter	Description	Type		
	force	To replace decrypted files with new files.	System.Boolean		
Equals(Object)	Description	Return Value	Type	Declaration	
	Determines whether the specified object is equal to this instance. Inherited from the Installation class.	<ul style="list-style-type: none"> True: The specified object is equal to this instance. False: The specified object is unequal to this instance. 	Boolean	C#	public override bool Equals(Object obj)
				Visual Basic	Public Overrides Function Equals (obj As Object) As Boolean
Equals(Installation)	Indicates whether the current object is equal to another object of the same type. Inherited from the Installation class.	<ul style="list-style-type: none"> True: The specified object is equal to this object. False: The specified object is unequal to this object. 	Boolean	C#	public bool Equals(Installation other)
				Visual Basic	Public Function Equals (other As Installation) As Boolean
GetHashCode	Returns a hash code for this instance. Inherited from the Installation class.	A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.	Int32	C#	public override int GetHashCode()
				Visual Basic	Public Overrides Function GetHashCode As Integer
ToString	Returns the display name to represent this instance. Inherited from the Installation class.	DisplayName	String	C#	public override string ToString()
				Visual Basic	Public Overrides Function ToString As String

Exceptions

Name	Methods	Condition
InstallationManagerException	Activate	An error occurs, if the installation does not support being activated and deactivated.
	Deactivate	
	DecryptEncryptedParts	An error occurs, if decryption is not possible, for example, when the API of dSPACE Installation Manager runs in a 32-bit process.

Properties

The class has the following properties:

Name	Description	Type	Declaration	
Description	Gets the description of the installation.	String	C#	public string Description { get; }
			Visual Basic	Public ReadOnly Property Description As StringGet
DisplayName	Gets the display name of the installation.	String	C#	public string DisplayName { get; }
			Visual Basic	Public ReadOnly Property DisplayName As StringGet
Extensions	Gets a list of extensions that are provided by this installation.	IEnumerable (Extension)	C#	public IEnumerable<Extension> Extensions { get; }
			Visual Basic	Public ReadOnly Property Extensions As IEnumerable(Of Extension)Get
InstallationDate	Gets information about when the installation was installed.	String	C#	public string InstallationDate { get; }
			Visual Basic	Public ReadOnly Property InstallationDate As StringGet
InstallationUser	Gets the user who installed the installation.	String	C#	public string InstallationUser { get; }
			Visual Basic	Public ReadOnly Property InstallationUser As StringGet
Is64Bit	Gets a value indicating whether this is a 64-bit installation.	Boolean	C#	public bool Is64Bit { get; }
			Visual Basic	Public ReadOnly Property Is64Bit As BooleanGet

Name	Description	Type	Declaration	
IsActive	Gets a value indicating whether the installation is the active installation. The meaning of active depends on the type of the installation (i.e., the product).	Boolean	C#	<code>public bool IsActive { get; }</code>
			Visual Basic	<code>Public ReadOnly Property IsActive As BooleanGet</code>
IsPrerelease	Gets a value indicating whether this installation is a pre-released version.	Boolean	C#	<code>public bool IsPrerelease { get; }</code>
			Visual Basic	<code>Public ReadOnly Property IsPrerelease As BooleanGet</code>
LeastSupportedMatlabVersion32Bit	Gets the oldest supported MATLAB version (32-bit) of this installation. MATLAB versions that are not older than this version can be used with this installation although this is not officially supported.	String	C#	<code>public string LeastSupportedMatlabVersion32Bit { get; }</code>
			Visual Basic	<code>Public ReadOnly Property LeastSupportedMatlabVersion32Bit As StringGet</code>
LeastSupportedMatlabVersion64Bit	Gets the oldest supported MATLAB version (64-bit) of this installation. MATLAB versions that are not older than this version can be used with this installation although this is not officially supported.	String	C#	<code>public string LeastSupportedMatlabVersion64Bit { get; }</code>
			Visual Basic	<code>Public ReadOnly Property LeastSupportedMatlabVersion64Bit As StringGet</code>
Products	Gets a list of products that belong to this installation.	IEnumerable (Product)	C#	<code>public IEnumerable<Product> Products { get; }</code>
			Visual Basic	<code>Public ReadOnly Property Products As IEnumerable(Of Product)Get</code>

Name	Description	Type	Declaration	
Release	Gets the dSPACE Release that the installation was (first) published with.	String	C#	public string Release { get; }
			Visual Basic	Public ReadOnly Property Release As StringGet
RootPath	Gets the root directory of the installation.	String	C#	public string RootPath { get; }
			Visual Basic	Public ReadOnly Property RootPath As StringGet
SupportedMatlabVersions32Bit	Gets a list of MATLAB versions (32-bit) that are officially supported by this installation.	IEnumerable (String)	C#	public IEnumerable<string> SupportedMatlabVersions32Bit { get; }
			Visual Basic	Public ReadOnly Property SupportedMatlabVersions32Bit As IEnumerable(Of String)Get
SupportedMatlabVersions64Bit	Gets a list of MATLAB versions (64-bit) that are officially supported by this installation.	IEnumerable (String)	C#	public IEnumerable<string> SupportedMatlabVersions64Bit { get; }
			Visual Basic	Public ReadOnly Property SupportedMatlabVersions64Bit As IEnumerable(Of String)Get
SupportsActivation	Gets a value indicating whether the installation can be activated and deactivated by the user.	Boolean	C#	public bool SupportsActivation { get; }
			Visual Basic	Public ReadOnly Property SupportsActivation As BooleanGet
TypeName	Gets the type name of the installation.	InstallationType	C#	public InstallationType TypeName { get; }
			Visual Basic	Public ReadOnly Property TypeName As InstallationTypeGet
Version	Gets the version of the installation.	String	C#	public string Version { get; }
			Visual Basic	Public ReadOnly Property Version As StringGet

Exceptions

Name	Condition
InstallationManagerException	Any error that occurs in the API.

Related topics

References

dSPACE.InstallationManager.API.Internal Namespace..... 22

InstallationManagerException Class

Purpose The exception that is thrown when an error occurs in the API of the dSPACE Installation Manager.

Declaration

C#	[SerializableAttribute] public class InstallationManagerException : Exception
Visual Basic	<SerializableAttribute> Public Class InstallationManagerException Inherits Exception

Related namespace

dSPACE.InstallationManager.API.Internal

Related topics

References

[dSPACE.InstallationManager.API.Internal Namespace..... 22](#)

MaintenanceExtension Class

Purpose Provides information about a hotfix, a patch or a service pack for a dSPACE product.

Declaration

C#	public class MaintenanceExtension
Visual Basic	Public Class MaintenanceExtension

Related namespace

dSPACE.InstallationManager.API.Internal

Methods

The class has the following methods:

Name	Description	Return Value	Type	Declaration	
ToString	Returns the display name to represent this instance. Overrides Object.ToString().	DisplayName	String	C#	public override string ToString()
				Visual Basic	Public Overrides Function ToString As String

Properties

The class has the following properties:

Name	Description	Type	Declaration	
DisplayName	Gets the display name.	String	C#	public string DisplayName { get; }
			Visual Basic	Public ReadOnly Property DisplayName As String Get
InstallationDate	Gets the installation date of this maintenance extension.	String	C#	public string InstallationDate { get; }
			Visual Basic	Public ReadOnly Property InstallationDate As String Get
InstallationUser	Gets the user who installed this maintenance extension.	String	C#	public string InstallationUser { get; }
			Visual Basic	Public ReadOnly Property InstallationUser As String Get
Kind	Gets the kind of this maintenance extension (e.g., hotfix).	MaintenanceExtensionKind	C#	public MaintenanceExtensionKind Kind { get; }
			Visual Basic	Public ReadOnly Property Kind As MaintenanceExtensionKind Get
Version	Gets the version of this maintenance extension.	String	C#	public string Version { get; }
			Visual Basic	Public ReadOnly Property Version As String Get

Exceptions

Name	Condition
InstallationManagerException	Any error that occurs in the API.

Related topics**References**

dSPACE.InstallationManager.API.Internal Namespace.....	22
MaintenanceExtensionKind Enumeration.....	70

Matlab Class

Purpose Provides information about a MATLAB installation.

Declaration

C#	<code>public class Matlab : IEquatable<Matlab></code>
Visual Basic	<code>Public Class Matlab Implements IEquatable(Of Matlab)</code>

Related namespace dSPACE.InstallationManager.API.Internal

Methods The class has the following methods:

Name	Characteristics				
Equals(Object)	Description	Return Value	Type	Declaration	
	Indicates whether the specified object is equal to this instance. Overrides <code>Object.Equals(Object)</code> .	<ul style="list-style-type: none"> True: The specified object is equal to this instance. False: The specified object is unequal to this instance. 	Boolean	C#	<code>public override bool Equals(Object obj)</code>
				Visual Basic	<code>Public Overrides Function Equals (obj As Object) As Boolean</code>
	Parameter	Description	Type		
	obj	The object to compare with this instance.	System.Object		

Name	Characteristics				
Equals(Matlab)	Description	Return Value	Type	Declaration	
	Indicates whether the current object is equal to another object of the same type.	<ul style="list-style-type: none">▪ True: The specified object is equal to this object.▪ False: The specified object is unequal to this object.	Boolean	C#	public bool Equals(Matlab other)
				Visual Basic	Public Function Equals (other As Matlab) As Boolean
	Parameter	Description	Type		
	Other	The object to compare with this object.	dSPACE.InstallationManager.API.Internal.Installation		
	Implements				
	IEquatable<Of T> .Equals(T)				
GetHashCode	Description	Return Value	Type	Declaration	
	Returns a hash code for this instance . Overrides Object.GetHashCode().	A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.	Int32	C#	public override int GetHashCode()
				Visual Basic	Public Overrides Function GetHashCode As Integer
ToString	Description	Return Value	Type	Declaration	
	Returns the display name to represent this instance. Overrides Object.ToString().	DisplayName	String	C#	public override string ToString()
				Visual Basic	Public Overrides Function ToString As String

Properties

The class has the following properties:

Name	Description	Type	Declaration	
DisplayName	Gets the display name of the product.	String	C#	public string DisplayName { get; }
			Visual Basic	Public ReadOnly Property DisplayName As String Get
Is64Bit	Gets a value indicating whether this MATLAB installation is a 64-bit installation.	Boolean	C#	public bool Is64Bit { get; }
			Visual Basic	Public ReadOnly Property Is64Bit As Boolean Get

Name	Description	Type	Declaration	
Release	Gets the MATLAB release this MATLAB installation belongs to.	String	C#	<code>public string Release { get; }</code>
			Visual Basic	<code>Public ReadOnly Property Release As String Get</code>
RootPath	Gets the root directory of this MATLAB installation.	String	C#	<code>public string RootPath { get; }</code>
			Visual Basic	<code>Public ReadOnly Property RootPath As String Get</code>
Version	Gets the version of this MATLAB installation.	String	C#	<code>public string Version { get; }</code>
			Visual Basic	<code>Public ReadOnly Property Version As String Get</code>

Exceptions

Name	Condition
InstallationManagerException	Any error that occurs in the API.

Related topics**References**

[dSPACE.InstallationManager.API.Internal Namespace..... 22](#)

MatlabIntegration Class

Purpose

Represents an integration of a dSPACE installation into a MATLAB installation.

Declaration

C#	<code>public class MatlabIntegration : IEquatable<MatlabIntegration></code>
Visual Basic	<code>Public Class MatlabIntegration Implements IEquatable(Of MatlabIntegration)</code>

Related namespace

dSPACE.InstallationManager.API.Internal

Methods

The class has the following methods:

Name	Characteristics				
Equals(Object)	Description	Return Value	Type	Declaration	
	Indicates whether the specified object is equal to this instance. Overrides <code>Object.Equals(Object)</code> .	<ul style="list-style-type: none"> True: The specified object is equal to this instance. False: The specified object is unequal to this instance. 	Boolean	C#	<code>public override bool Equals(Object obj)</code>
				Visual Basic	<code>Public Overrides Function Equals (obj As Object) As Boolean</code>
	Parameter	Description	Type		
	obj	The object to compare with this instance.	System.Object		

Name	Characteristics				
Equals(MatlabIntegration)	Description	Return Value	Type	Declaration	
	Indicates whether the current object is equal to another object of the same type.	<ul style="list-style-type: none">▪ True: The specified object is equal to this object.▪ False: The specified object is unequal to this object.	Boolean	C#	public bool Equals(MatlabIntegration other)
				Visual Basic	Public Function Equals (other As MatlabIntegration) As Boolean
	Parameter	Description	Type		
	Other	The object to compare with this object.	dSPACE.InstallationManager.API.Internal.MatlabIntegration		
	Implements				
	IEquatable(Of T).Equals(T)				
GetHashCode	Description	Return Value	Type	Declaration	
	Returns a hash code for this instance . Overrides Object.GetHashCode().	A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.	Int32	C#	public override int GetHashCode()
				Visual Basic	Public Overrides Function GetHashCode As Integer
ToString	Description	Return Value	Type	Declaration	
	Returns the display name to represent this instance. Overrides Object.ToString().	DisplayName	String	C#	public override string ToString()
				Visual Basic	Public Overrides Function ToString As String

Properties

The class has the following properties:

Name	Description	Type	Declaration	
DisplayName	Gets the display name of this integration.	String	C#	public string DisplayName { get; }
			Visual Basic	Public ReadOnly Property DisplayName As String Get
Installation	Gets the dSPACE installation of this integration.	Installation	C#	public Installation Installation { get; }
			Visual Basic	Public ReadOnly Property Installation As Installation Get
IsPreferred	Gets a value indicating whether the MATLAB installation of this integration is the one that the dSPACE installation will use by default. True if the MATLAB installation is the preferred one, false otherwise.	Boolean	C#	public bool IsPreferred { get; }
			Visual Basic	Public ReadOnly Property IsPreferred As Boolean Get
Matlab	Gets the MATLAB installation of this integration.	Matlab	C#	public Matlab Matlab { get; }
			Visual Basic	Public ReadOnly Property Matlab As Matlab Get

Exceptions

Name	Condition
InstallationManagerException	Any error that occurs in the API.

Related topics**References**

dSPACE.InstallationManager.API.Internal Namespace..... 22

MatlabIntegrationConfig Class

Purpose

Modifies an existing MATLAB integration (integration of dSPACE software into a specific MATLAB installation).

Declaration

C#	public class MatlabIntegrationConfig : MatlabIntegration
Visual Basic	Public Class MatlabIntegrationConfig Inherits MatlabIntegration

Related namespace

dSPACE.InstallationManager.API.Internal

Methods

The class has the following methods:

Name	Description	Return Value	Type	Declaration	
Equals(Object)	Indicates whether the specified object is equal to this instance.	<ul style="list-style-type: none"> True: The specified object is equal to this instance. False: The specified object is unequal to this instance. 	Boolean	C#	public override bool Equals(Object obj)
				Visual Basic	Public Overrides Function Equals (obj As Object) As Boolean
Equals(MatlabIntegration)	Indicates whether the specified object is equal to this object.	<ul style="list-style-type: none"> True: The specified object is equal to this object. False: The specified object is unequal to this object. 	Boolean	C#	public bool Equals(MatlabIntegration other)
				Visual Basic	Public Function Equals (other As MatlabIntegration) As Boolean
GetHashCode	Returns a hash code for this instance.	A hash code for this instance, suitable for use in hashing algorithms and data structures like a hash table.	Int32	C#	public override int GetHashCode()
				Visual Basic	Public Overrides Function GetHashCode As Integer
SetAsPreferred	Sets this integration as the preferred one.	–	–	C#	public void SetAsPreferred()
				Visual Basic	Public Sub SetAsPreferred
ToString	Returns the display name to represent this instance.	DisplayName	String	C#	public override string ToString()
				Visual Basic	Public Overrides Function ToString As String

Properties

The class has the following properties:

Name	Description	Type	Declaration	
DisplayName	Gets the display name of this integration.	String	C#	public string DisplayName { get; }
			Visual Basic	Public ReadOnly Property DisplayName As String Get

Name	Description	Type	Declaration	
Installation	Gets the dSPACE installation of this integration.	Installation	C#	public Installation Installation { get; }
			Visual Basic	Public ReadOnly Property Installation As Installation Get
IsPreferred	Gets a value indicating whether the MATLAB installation of this integration is the one that the dSPACE installation will use by default (i.e., the preferred integration) .	Boolean	C#	public bool IsPreferred { get; }
			Visual Basic	Public ReadOnly Property IsPreferred As Boolean Get
Matlab	Gets the MATLAB installation of this integration.	Matlab	C#	public Matlab Matlab { get; }
			Visual Basic	Public ReadOnly Property Matlab As Matlab Get

Exceptions

Name	Condition
InstallationManagerException	Any error that occurs in the API.

Related topics**References**

[dSPACE.InstallationManager.API.Internal Namespace..... 22](#)

MatlabIntegrationsConfig Class

Purpose

Provides information about MATLAB integrations and lets you add or remove MATLAB integrations to/from the dSPACE Installation Manager.

Declaration

C#	public class MatlabIntegrationsConfig : IReadOnlyList, IReadOnlyCollection, IEnumerable<MatlabIntegrationConfig>, ICollection, IEnumerable
Visual Basic	Public Class MatlabIntegrationsConfig Implements IReadOnlyList, IReadOnlyCollection, IEnumerable(Of MatlabIntegrationConfig), ICollection, IEnumerable

Related namespace

dSPACE.InstallationManager.API.Internal

Methods

The class has the following methods:

Name	Characteristics				
Add(MatlabIntegration)	Description	Return Value	Type	Declaration	
	Adds a new MATLAB integration according to the specified parameter.	–	–	C#	public void Add(MatlabIntegration matlabIntegration)
				Visual Basic	Public Sub Add (matlabIntegration As MatlabIntegration)
	Parameter	Description	Type		
	matlabIntegration	The MATLAB integration instance containing the relevant information.	dSPACE.InstallationManager.API.Internal.MatlabIntegration		
Add(String, String, Boolean)	Description	Return Value	Type	Declaration	
	Adds a new MATLAB integration according to the specified arguments. If the MATLAB integration already exists, nothing will be done.	–	–	C#	public void Add(string dspaceInstallationRoot, string matlabInstallationRoot, bool shouldBePreferredIntegration)
				Visual Basic	Public Sub Add (dspaceInstallationRoot As String, matlabInstallationRoot As String, shouldBePreferredIntegration As Boolean)
	Parameter	Description	Type		
	dspaceInstallationRoot	The dSPACE installation root.	System.String		
	matlabInstallationRoot	The MATLAB installation root.	System.String		
	shouldBePreferredIntegration	Indicates whether the new integration is the preferred one.	System.Boolean		

Name	Characteristics				
Add(Installation, Matlab, Boolean)	Description	Return Value	Type	Declaration	
	Adds a new MATLAB integration according to the specified parameter.	–	–	C#	public void Add(Installation dspaceInstallation, Matlab matlabInstallation, bool shouldBePreferredIntegration)
				Visual Basic	Public Sub Add (dspaceInstallation As Installation, matlabInstallation As Matlab, shouldBePreferredIntegration As Boolean)
	Parameter	Description	Type		
	dspaceInstallation	The dSPACE installation.	dSPACE.InstallationManager.API.Internal.Installation		
	matlabInstallation	The MATLAB installation.	dSPACE.InstallationManager.API.Internal.Matlab		
CanAdd(String, String)	Description	Return Value	Type	Declaration	
	Determines whether the specified dSPACE installation can be integrated into the specified MATLAB installation.	<ul style="list-style-type: none"> ▪ True: Integration is possible. ▪ False: Integration is not possible. 	Boolean	C#	public bool CanAdd(string dspaceInstallationRoot, string matlabInstallationRoot)
				Visual Basic	Public Function CanAdd (dspaceInstallationRoot As String, matlabInstallationRoot As String) As Boolean
	Parameter	Description	Type		
	dspaceInstallationRoot	The dSPACE installation root.	System.String		
	matlabInstallationRoot	The MATLAB installation root.	System.String		

Name	Characteristics				
CanAdd(Installation, Matlab)	Description	Return Value	Type	Declaration	
	Determines whether the specified dSPACE software installation can be integrated to the specified MATLAB installation.	<ul style="list-style-type: none"> ▪ True: Integration is possible. ▪ False: Integration is not possible. 	Boolean	C#	public bool CanAdd(Installation dspaceInstallation, Matlab matlabInstallation)
				Visual Basic	Public Function CanAdd (dspaceInstallation As Installation, matlabInstallation As Matlab) As Boolean
	Parameter	Description	Type		
	dspaceInstallation	The dSPACE installation.	dSPACE.InstallationManager.API.Internal.Installation		
	matlabInstallation	The MATLAB installation.	dSPACE.InstallationManager.API.Internal.Matlab		
Contains(MatlabIntegration)	Description	Return Value	Type	Declaration	
	Determines whether the MATLAB integration according to the specified argument exists.	<ul style="list-style-type: none"> ▪ True: The integration exists. ▪ False: The integration does not exist. 	Boolean	C#	public bool Contains(MatlabIntegration matlabIntegration)
				Visual Basic	Public Function Contains (matlabIntegration As MatlabIntegration) As Boolean
	Parameter	Description	Type		
	matlabIntegration	The MATLAB integration instance containing the relevant information.	dSPACE.InstallationManager.API.Internal.MatlabIntegration		

Name	Characteristics				
Contains(String, String)	Description	Return Value	Type	Declaration	
	Determines whether the MATLAB integration corresponding to the specified arguments currently exists.	<ul style="list-style-type: none"> ▪ True: The integration exists. ▪ False: The integration does not exist. 	Boolean	C#	public bool Contains(string dspaceInstallationRoot, string matlabInstallationRoot)
				Visual Basic	Public Function Contains (dspaceInstallationRoot As String, matlabInstallationRoot As String) As Boolean
	Parameter	Description	Type		
	dspaceInstallationRoot	The dSPACE installation root.	System.String		
	matlabInstallationRoot	The MATLAB installation root.	System.String		

Name	Characteristics				
CopyTo	Description	Return Value	Type	Declaration	
	Copies the elements of the collection to an array, starting at a particular array index.	–	–	C#	public void CopyTo(Array array,int index)
				Visual Basic	Public Sub CopyTo (array As Array,index As Integer)
	Parameter	Description	Type		
	array	The one-dimensional array that is the destination of the elements copied from the collection. The array must have zero-based indexing.	System.Array		
	index	The zero-based index in array at which copying begins.	System.Int32		
	Implements				
	ICollection.CopyTo(Array, Int32)				
Find(MatlabIntegration)	Description	Return Value	Type	Declaration	
	Gets the MATLAB integration for the specified parameter if such a MATLAB integration exists.	The requested MATLAB integration instance or null if it does not exist.	MatlabIntegration Config	C#	public MatlabIntegrationConfig Find(MatlabIntegration matlabIntegration)
				Visual Basic	Public Function Find (matlabIntegration As MatlabIntegration) As MatlabIntegrationConfig
	Parameter	Description	Type		
	matlabIntegration	The MATLAB integration instance containing the relevant information.	dSPACE.InstallationManager.API.Internal.MatlabIntegration		

Name	Characteristics				
Find(String, String)	Description	Return Value	Type	Declaration	
	Gets the MATLAB integration for the specified parameters if such a MATLAB integration exists.	The requested MATLAB integration instance or null if it does not exist.	MatlabIntegration Config	C#	public MatlabIntegrationConfig Find(string dspaceInstallationRoot, string matlabInstallationRoot)
				Visual Basic	Public Function Find (dspaceInstallationRoot As String, matlabInstallationRoot As String) As MatlabIntegrationConfig
	Parameter	Description	Type		
	dspaceInstallationRoot	The dSPACE installation root.	System.String		
	matlabInstallationRoot	The MATLAB installation root.	System.String		

Name	Characteristics				
GetEnumerator	Description	Return Value	Type	Declaration	
	Returns an enumerator that iterates through the collection.	An <code>IEnumerator(Of T)</code> that can be used to iterate through the collection.	<code>IEnumerator(Of MatlabIntegrationConfig)</code>	C#	<code>public IEnumerator<MatlabIntegrationConfig> GetEnumerator()</code>
				Visual Basic	<code>Public Function GetEnumerator As IEnumerator(Of MatlabIntegrationConfig)</code>
	Implements				
	<code>IEnumerable(Of T).GetEnumerator</code>				
GetMatlabVersionSupport(String, String)	Description	Return Value	Type	Declaration	
	Gets a value indicating whether the specified dSPACE installation can be integrated into the version of the specified MATLAB installation.	The requested value.	<code>MatlabVersionSupport</code>	C#	<code>public MatlabVersionSupport GetMatlabVersionSupport(string dspaceInstallationRoot, string matlabInstallationRoot)</code>
				Visual Basic	<code>Public Function GetMatlabVersionSupport (dspaceInstallationRoot As String, matlabInstallationRoot As String) As MatlabVersionSupport</code>
	Parameter	Description	Type		
	<code>dspaceInstallationRoot</code>	The dSPACE installation root.	<code>System.String</code>		
	<code>matlabInstallationRoot</code>	The MATLAB installation root.	<code>System.String</code>		

Name	Characteristics				
GetMatlabVersionSupport(Installation, Matlab)	Description	Return Value	Type	Declaration	
	Gets a value indicating whether the specified dSPACE installation can be integrated into the version of the specified MATLAB installation.	The requested value.	MatlabVersionSupport	C#	public MatlabVersionSupport GetMatlabVersionSupport(Installation dspaceInstallation, Matlab matlabInstallation)
				Visual Basic	Public Function GetMatlabVersionSupport (dspaceInstallation As Installation, matlabInstallation As Matlab) As MatlabVersionSupport
	Parameter	Description	Type		
	dspaceInstallation	The dSPACE installation.	dSPACE.InstallationManager.API.Internal.Installation		
	matlabInstallation	The MATLAB installation.	dSPACE.InstallationManager.API.Internal.Matlab		
Remove(MatlabIntegration)	Description	Return Value	Type	Declaration	
	Removes the MATLAB integration that corresponds to the specified argument. If the MATLAB integration does not exist, nothing will be done.	–	–	C#	public void Remove(MatlabIntegration matlabIntegration)
				Visual Basic	Public Sub Remove (matlabIntegration As MatlabIntegration)
	Parameter	Description	Type		
	matlabIntegration	The MatlabIntegration instance containing the relevant information.	dSPACE.InstallationManager.API.Internal.MatlabIntegration		

Name	Characteristics				
Remove(String, String)	Description	Return Value	Type	Declaration	
	Removes the MATLAB integration that corresponds to the specified arguments. If the MATLAB integration does not exist, nothing will be done.	–	–	C#	public void Remove(string dspaceInstallationRoot, string matlabInstallationRoot)
				Visual Basic	Public Sub Remove (dspaceInstallationRoot As String, matlabInstallationRoot As String)
	Parameter	Description	Type		
	dspaceInstallationRoot	The dSPACE installation root.	System.String		
	matlabInstallationRoot	The MATLAB installation root.	System.String		
Remove(Installation, Matlab)	Description	Return Value	Type	Declaration	
	Removes the MATLAB integration that corresponds to the specified parameters. If the MATLAB integration does not exist, nothing will be done.	–	–	C#	public void Remove(Installation dspaceInstallation, Matlab matlabInstallation)
				Visual Basic	Public Sub Remove (dspaceInstallation As Installation, matlabInstallation As Matlab)
	Parameter	Description	Type		
	dspaceInstallation	The dSPACE installation.	dSPACE.InstallationManager.API.Internal.Installation		
	matlabInstallation	The MATLAB installation.	dSPACE.InstallationManager.API.Internal.Matlab		
RemoveAll	Description	Return Value	Type	Declaration	
	Removes all existing MATLAB integrations.	–	–	C#	public void RemoveAll()
				Visual Basic	Public Sub RemoveAll

Exceptions

Name	Condition
InstallationManagerException	Any error that occurs in the API.

Properties

The class has the following properties:

Name					
Count	Description	Return Value	Type	Declaration	
	Gets the number of elements in the collection.	The number of elements in the collection.	Int32	C#	public int Count { get; }
				Visual Basic	Public ReadOnly Property Count As Integer Get
	Implements				
ICollection.Count					
IsSynchronized	Description	Return Value	Type	Declaration	
	Gets a value indicating whether access to the collection is synchronized (thread safe).	<ul style="list-style-type: none">▪ True: Access to the ICollection is synchronized (thread safe).▪ False: Access to the ICollection is not synchronized.	Boolean	C#	public bool IsSynchronized { get; }
				Visual Basic	Public ReadOnly Property IsSynchronized As Boolean Get
	Implements				
ICollection.IsSynchronized					
Item	Description	Return Value	Type	Declaration	
	Gets the element at the specified index in the read-only list.	The requested element.	MatlabIntegrationConfig	C#	public MatlabIntegrationConfig this[int index] { get; }
				Visual Basic	Public ReadOnly Default Property Item (index As Integer) As MatlabIntegrationConfig Get
	Parameter	Description	Type		
	index	The index.	System.Int32		
SyncRoot	Description	Return Value	Type	Declaration	
	Gets an object that can be used to synchronize access to the collection.	An object that can be used to synchronize access to the collection.	Object	C#	public Object SyncRoot { get; }
				Visual Basic	Public ReadOnly Property SyncRoot As Object Get
	Implements				
ICollection.SyncRoot					

Exceptions

Name	Condition
InstallationManagerException	Any error that occurs in the API.

Related topics**References**

[dSPACE.InstallationManager.API.Internal Namespace..... 22](#)

Product Class

Purpose

Provides information about a dSPACE product.

Declaration

C#	public class Product
Visual Basic	Public Class Product

Related namespace

dSPACE.InstallationManager.API.Internal

Methods

The class has the following methods:

Name	Description	Return Value	Type	Declaration	
ToString	Returns the display name to represent this instance. Overrides <code>Object.ToString()</code> .	DisplayName	String	C#	public override string ToString()
				Visual Basic	Public Overrides Function ToString As String

Properties

The class has the following properties.

Name	Description	Type	Declaration	
CompatibleExtensions	Gets a list of available extensions that can be used by this product.	ReadOnlyCollection (Extension)	C#	public ReadOnlyCollection<Extension> CompatibleExtensions { get; }
			Visual Basic	Public ReadOnly Property CompatibleExtensions As ReadOnlyCollection(Of Extension) Get

Name	Description	Type	Declaration	
Description	Gets the description of the product.	String	C#	public string Description { get; }
			Visual Basic	Public ReadOnly Property Description As String Get
DisplayName	Gets the display name of the product.	String	C#	public string DisplayName { get; }
			Visual Basic	Public ReadOnly Property DisplayName As String Get
InstallationDate	Gets the installation date of this maintenance extension.	String	C#	public string InstallationDate { get; }
			Visual Basic	Public ReadOnly Property InstallationDate As String Get
InstallationUser	Gets the user who installed this product.	String	C#	public string InstallationUser { get; }
			Visual Basic	Public ReadOnly Property InstallationUser As String Get
MaintenanceExtensions	Gets a list of maintenance extensions that belong to this product.	ReadOnlyCollection (MaintenanceExtension)	C#	public ReadOnlyCollection<MaintenanceExtension> MaintenanceExtensions { get; }
			Visual Basic	Public ReadOnly Property MaintenanceExtensions As ReadOnlyCollection(Of MaintenanceExtension) Get
UniqueName	Gets the unique name that identifies this product.	String	C#	public string UniqueName { get; }
			Visual Basic	Public ReadOnly Property UniqueName As String Get
Version	Gets the version of this product.	String	C#	public string Version { get; }
			Visual Basic	Public ReadOnly Property Version As String Get

Exceptions

Name	Condition
InstallationManagerException	Any error that occurs in the API.

Related topics

References

[dSPACE.InstallationManager.API.Internal Namespace](#)..... 22

Enumerations

Where to go from here

Information in this section

InstallationType Enumeration.....	69
Represents the type name of dSPACE installations.	
MaintenanceExtensionKind Enumeration.....	70
Represents the kind of a maintenance extension (hotfix, patch or service pack).	
MatlabVersionSupport Enumeration.....	71
Specifies whether a dSPACE installation can be integrated into a specific Matlab software version.	
ReportContent Enumeration.....	72
Indicates which scope of the diagnostic information you want to collect and save to a ZIP file.	

InstallationType Enumeration

Purpose

Represents the type name of dSPACE installations.

Declaration

C#	<code>public enum InstallationType</code>
Visual Basic	<code>Public Enumeration InstallationType</code>

Related namespace

dSPACE.InstallationManager.API.Internal

Enumeration values

The enumeration has the following values:

Value	Name	Description
0	Unknown	An unknown installation type
1	General	A utility or a base installation
2	Solution	A solution installation
3	dSPACEReleaseClassic	An installation released before dSPACE Release 2010-B
4	RCP_HIL	An RCP and HIL software installation
5	TargetLink	A TargetLink installation
6	ModelCompare	A Model Compare installation
7	TargetLinkBlocksetStandalone	A (stand-alone) TargetLink Blockset installation
8	SystemDesk	A SystemDesk installation
9	ControlDesk	A ControlDesk installation
10	RealTimeTesting	A Real-Time Testing installation
11	PythonExtensions	A dSPACE Python Extension installation
12	AutomationDesk	An AutomationDesk installation
13	HIL_XIL_APIDotNet	A dSPACE HIL API .NET and XIL API .NET installation
14	SYNECT	A SYNECT installation
15	VEOS	A VEOS installation
16	FirmwareArchives	A dSPACE Firmware Archives installation
17	FirmwareManager	A dSPACE Firmware Manager installation.
18	MCD3Client	A MCD3 client installation
19	SYNECTServer	A SYNECT server installation
20	SYNECTTestExecutor	A SYNECT test executor installation
21	VariableEditor	A Variable Editor installation

Related topics**References**

dSPACE.InstallationManager.API.Internal Namespace.....	22
Installation Class.....	36
MatlabIntegration Class.....	50

MaintenanceExtensionKind Enumeration

Purpose

Represents the kind of a maintenance extension (hotfix, patch or service pack).

Declaration

C#	<code>public enum MaintenanceExtensionKind</code>
Visual Basic	<code>Public Enumeration MaintenanceExtensionKind</code>

Namespace dSPACE.InstallationManager.API.Internal

Enumeration values The enumeration has the following values:

Value	Name	Description
0	Undefined	An undefined maintenance extension
1	Hotfix	A hotfix
2	Patch	A patch
3	ServicePack	A service pack

Related topics

References

dSPACE.InstallationManager.API.Internal Namespace.....	22
Installation Class.....	36
MaintenanceExtension Class.....	46

MatlabVersionSupport Enumeration

Purpose Specifies whether a dSPACE installation can be integrated into a specific MATLAB software version.

Declaration

C#	public enum MatlabVersionSupport
Visual Basic	Public Enumeration MatlabVersionSupport

Namespace dSPACE.InstallationManager.API.Internal

Enumeration values The enumeration has the following values:

Value	Name	Description
0	NotSupported	The dSPACE installation cannot be integrated into the MATLAB software version.
1	NotOfficiallySupported	The dSPACE installation can be integrated into the MATLAB software version, although this MATLAB version is not officially supported.
2	Supported	The dSPACE installation can be integrated into the MATLAB software version. This MATLAB version is officially supported.

Related topics

References

[dSPACE.InstallationManager.API.Internal Namespace..... 22](#)

ReportContent Enumeration

Purpose

Indicates which scope of the diagnostic information you want to collect and save to a ZIP file.

Declaration

C#	<code>public enum ReportContent</code>
Visual Basic	<code>Public Enumeration ReportContent</code>

Related namespace

dSPACE.InstallationManager.API.Internal

Enumeration values

The enumeration has the following values:

Value	Name	Description
0	Basic	Only the diagnostic information in the Basic General Information group is collected and saved to the ZIP file.
1	Extended	Diagnostic information in the Basic General Information group and in the Extended General Information group is collected and saved to the ZIP file.
2	All	All relevant diagnostic information is collected and saved to the ZIP file. This includes diagnostic information in the Basic General Information group, in the Extended General Information groups and the relevant diagnostic information of all available dSPACE installations on your PC.

Related topics

References

[Diagnostics Class..... 29](#)
[dSPACE.InstallationManager.API.Internal Namespace..... 22](#)

A

accessing the API 9

C

Class

- Configuration 29
- ConfigurationChange 26
- ConfigurationInfo 27
- Diagnostics 29
- Extension 35
- Installation 36
- InstallationConfig 40
- InstallationManagerException 46
- MaintenanceExtension 46
- Matlab 48
- MatlabIntegration 50
- MatlabIntegrationConfig 53
- MatlabIntegrationsConfig 55
- Product 66

Common Program Data folder 6

CommonProgramDataFolder 6

Configuration Class 29

ConfigurationChange Class 26

ConfigurationInfo Class 27

D

decryption 40

Diagnostics Class 29

Documents folder 6

DocumentsFolder 6

dSPACE.InstallationManager.API

- Internal Namespace 22

- Namespace 21

dSPACE.InstallationManager.API Namespace 21

dSPACE.InstallationManager.API.Internal

Namespace 22

E

Enumeration

- InstallationType 69

- MaintenanceExtensionKind 70

- MatlabVersionSupport 71

- ReportContent 72

examples

- Python 19

- Windows PowerShell 13

examples for using Python 19

examples Windows PowerShell 13

Extension Class 35

I

Installation Class 36

InstallationConfig Class 40

InstallationManagerException Class 46

InstallationType Enumeration 69

L

Local Program Data folder 6

LocalProgramDataFolder 6

M

MaintenanceExtension Class 46

MaintenanceExtensionKind Enumeration 70

Matlab Class 48

MatlabIntegration Class 50

MatlabIntegrationConfig Class 53

MatlabIntegrationsConfig Class 55

MatlabVersionSupport Enumeration 71

N

Namespaces 21

P

Product Class 66

R

ReportContent Enumeration 72

