

Namespace MES.MQC.DataSourceLibrary.Adapters

Classes

[Adapter](#)

@deprecated For backwards compatibility only

[AdapterBase](#)

The AdapterBase class is the base for all MQC Adapters.

[AdapterContext](#)

@deprecated For backwards compatibility only

[AdapterBaseContext](#)

[ApiConnector](#)

The Api Adapter class is the base for all MQC API Connectors

[ApiConnectorContext](#)

[FileReader](#)

The FileAdapter class is the base for all MQC File Readers.

[FileReaderContext](#)

Class Adapter

Namespace: [MES.MQC.DataSourceLibrary.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

@deprecated For backwards compatibility only

```
public abstract class Adapter : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← Adapter

Inherited Members

[FileReader.Priority](#) , [FileReader.FileExtensions](#) , [FileReader.HumanReadableFileCount](#) ,
[FileReader.HumanReadableFileExtensions](#) , [FileReader.HumanReadableStartFileNames](#) ,
[FileReader.ReportDateTimeExpected](#) , [FileReader.IsValid\(FileReaderContext\)](#) ,
[FileReader.Read\(FileReaderContext\)](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Name](#) , [AdapterBase.Description](#) ,
[AdapterBase.DataSource](#) , [AdapterBase.Options](#) , [AdapterBase.TransformStringToValue\(string\)](#) ,
[AdapterBase.TransformStringToDate\(string\)](#) ,
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Class AdapterBase

Namespace: [MES.MQC.DataSourceLibrary.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The AdapterBase class is the base for all MQC Adapters.

```
public abstract class AdapterBase
```

Inheritance

[object](#) ← AdapterBase

Derived

[ApiConnector](#), [FileReader](#)

Inherited Members

[object.ToString\(\)](#), [object.Equals\(object\)](#), [object.Equals\(object, object\)](#),
[object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public virtual string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is

selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public virtual string Description { get; }
```

Property Value

[string](#)

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public virtual string Name { get; }
```

Property Value

[string](#)

Options

Adapter options for the specific adapter.

```
public AdapterOptions Options { get; set; }
```

Property Value

AdapterOptions

Methods

ReadFilePathsInZipFile(string)

```
protected List<string> ReadFilePathsInZipFile(string zipFilePath)
```

Parameters

`zipFilePath string`

Returns

`List<string>`

ThrowWarningForEmptyDataProperty(AdapterDataBase,
AdapterContextBase, string, string)

```
protected void ThrowWarningForEmptyDataProperty(AdapterDataBase adapterDataBase,  
AdapterContextBase context, string propertyName, string value)
```

Parameters

`adapterDataBase AdapterDataBase`

`context AdapterContextBase`

`propertyName string`

`value string`

TransformStringToDateTIme(AdapterContext, string)

This method is deprecated, use

```
TransformStringToDateTIme(value)
```

instead

```
protected DateTime TransformStringToDateTIme(AdapterContext context, string value)
```

Parameters

`context AdapterContext`

AdapterContext

value [string](#)

Date as string

Returns

[DateTime](#)

Datetime transformed from the string or the creation date, if string was not a valid datetime

TransformStringToDate([string](#))

TransformStringToValue is a Utility method. Parse a string value to a DateTime.

protected [DateTime?](#) TransformStringToDate([string](#) **value**)

Parameters

value [string](#)

Date as string

Returns

[DateTime](#)?

Datetime transformed from the string or null, if string was not a valid datetime

TransformStringToValue([string](#))

TransformStringToValue is a Utility method. Parse a string value to a double value in a culture independent (invariant) way.

protected static double? TransformStringToValue([string](#) **value**)

Parameters

value [string](#)

Returns

double?

Class AdapterContext

Namespace: [MES.MQC.DataSourceLibrary.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

@deprecated For backwards compatibility only

```
public class AdapterContext : FileReaderContext, IDisposable
```

Inheritance

[object](#) ← [AdapterContextBase](#) ← [FileReaderContext](#) ← AdapterContext

Implements

[IDisposable](#)

Inherited Members

[FileReaderContext.Path](#) , [FileReaderContext.DirectoryPath](#) , [FileReaderContext.Name](#) ,
[FileReaderContext.NameWithoutExtension](#) , [FileReaderContext.Extension](#) , [FileReaderContext.Content](#) ,
[FileReaderContext.CreationDate](#) , [FileReaderContext.Dispose\(\)](#) , [FileReaderContext.AsXDocument\(\)](#) ,
[FileReaderContext.AsHtmlDocument\(string\)](#) , [FileReaderContext.AsDataSet\(int\)](#) ,
[AdapterContextBase.ImportFindings](#) , [AdapterContextBase.MessageBag](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Constructors

AdapterContext(string, bool)

```
public AdapterContext(string filePath, bool importFindings = false)
```

Parameters

filePath [string](#)

importFindings [bool](#)

Class AdapterContextBase

Namespace: [MES.MQC.DataSourceLibrary.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public abstract class AdapterContextBase
```

Inheritance

[object](#) ← AdapterContextBase

Derived

[ApiConnectorContext](#), [FileReaderContext](#)

Inherited Members

[object.ToString\(\)](#), [object.Equals\(object\)](#), [object.Equals\(object, object\)](#),
[object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#)

Properties

ImportFindings

Should the adapter import findings?

```
public bool ImportFindings { get; }
```

Property Value

[bool](#)

MessageBag

Message bag, where errors and warnings can be thrown.

```
public AdapterMessageBag MessageBag { get; }
```

Property Value

[AdapterMessageBag](#)

Class ApiConnector

Namespace: [MES.MQC.DataSourceLibrary.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The Api Adapter class is the base for all MQC API Connectors

```
public abstract class ApiConnector : AdapterBase, IFormProvider
```

Inheritance

[object](#) ← [AdapterBase](#) ← ApiConnector

Implements

[IFormProvider](#)

Derived

[GitlabApiConnector](#), [JenkinsPipelineArtifactsApiConnector](#)

Inherited Members

[AdapterBase.Name](#), [AdapterBase.Description](#), [AdapterBase.Options](#),
[AdapterBase.TransformStringToValue\(string\)](#), [AdapterBase.TransformStringToDateTIme\(string\)](#),
[AdapterBase.TransformStringToDateTIme\(AdapterContext, string\)](#),
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#),
,
[AdapterBase.ReadFilePathsInZipFile\(string\)](#), [object.ToString\(\)](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be "Unknown" and the DataSource of each AdapterData object has to be defined. DataSource should be "Undefined", if only files are downloaded.

```
public override string DataSource { get; }
```

Property Value

[string](#) ↗

Methods

CheckAvailable(ApiConnectorContext)

CheckAvailable must be implemented by the Api Adapter class. This method is called to show a warning for configured but unavailable data sources in the UI.

```
public abstract bool CheckAvailable(ApiConnectorContext context)
```

Parameters

context [ApiConnectorContext](#)

ApiConnectorContext

Returns

[bool](#) ↗

boolean (is the api available/accessible)

CheckModified(ApiConnectorContext)

CheckModified must be implemented by the Api Adapter class. This method is called by the local and server side monitoring, to check if the data sources should be updated.

```
public abstract bool CheckModified(ApiConnectorContext context)
```

Parameters

context [ApiConnectorContext](#)

ApiConnectorContext

Returns

[bool](#)

boolean (are there any changes)

ConfigureForm(IForm, string[])

ConfigureForm can be implemented by an API adapter. It is used to modify the FormSchema of the Add/Edit DataSource dialog for the AdapterConfiguration depending on the values entered (e.g. to make options or input fields visible and/or deactivate them based on a correct API request). It is called for the initial creation of the FormSchema with all fields as modified and when a field that was enabled by RefreshOnModified has been changed by the user. This method should also validate the configuration, not if the schema is valid, but if the values are usable (e.g. if the api is accessible and the authentication is valid).

```
public virtual FormError[] ConfigureForm(IForm form, string[] modifiedFields)
```

Parameters

[form](#) [IForm](#)

IForm, the AdapterConfiguration

[modifiedFields](#) [string](#)[]

Array of the modified fields as string (Form-Subfields as field.field or field.index.field)

Returns

[FormError](#)[]

Array of FormError, if a validation failed

Download(ApiConnectorContext)

Download can be implemented by the Api Adapter class. At least one of Download and Read must be implemented. The implementation of this method should access the configured API, download and save files to the context.DownloadPath directory and return a AdapterDownloadResult.

```
protected virtual AdapterDownloadResult Download(ApiConnectorContext context)
```

Parameters

`context ApiConnectorContext`

ApiConnectorContext

Returns

[AdapterDownloadResult](#)

AdapterDownloadResult

Execute(ApiConnectorContext)

Internal method

```
public ApiConnectorResult Execute(ApiConnectorContext context)
```

Parameters

`context ApiConnectorContext`

ApiConnectorContext

Returns

[ApiConnectorResult](#)

ApiConnectorResult

PreviewForm(IForm, string, out int)

PreviewForm can be implemented by an API Connector. It is used to load the Preview, if one was defined in the AdapterConfiguration. Based on the focused form field, the Preview is shown in the Dialog and provides the ability to load by clicking on a button. This button click calls this method. Based on the configuration and the preview type, the api should load the relevant data, limited to ~100 for better performance, and a array of FormPreview should be returned.

```
public virtual FormPreview[] PreviewForm(IForm form, string preview, out int totalCount)
```

Parameters

form [IForm](#)

IForm, the AdapterConfiguration

preview [string](#)

The preview type (string)

totalCount [int](#)

Output of totalCount, if 0 shown as ? in the preview dialog

Returns

[FormPreview\[\]](#)

Array of FormPreview, to show Title and optionally Description and DateTime

Read(ApiConnectorContext)

Read can be implemented by the Api Adapter class. At least one of Download and Read must be implemented. The implementation of this method should access the configured API, create AdapterData, AdapterMeasures and/or AdapterFindings and return a AdapterReadResult.

```
protected virtual AdapterReadResult Read(ApiConnectorContext context)
```

Parameters

context [ApiConnectorContext](#)

ApiConnectorContext

Returns

[AdapterReadResult](#)

AdapterReadResult

Class ApiConnectorContext

Namespace: [MES.MQC.DataSourceLibrary.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class ApiConnectorContext : AdapterContextBase
```

Inheritance

[object](#) ← [AdapterContextBase](#) ← ApiConnectorContext

Inherited Members

[AdapterContextBase.ImportFindings](#), [AdapterContextBase.MessageBag](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

Constructors

ApiConnectorContext(ApiConnectorConfiguration, string, DateTime?, DateTime?, DateTime?, Action, Action<long>, Action<string, Action>, Func<string, int, string, IDisposable>)

```
public ApiConnectorContext(ApiConnectorConfiguration configuration, string appDataPath,  
DateTime? projectStartDate = null, DateTime? projectEndDate = null, DateTime? lastImportDate  
= null, Action progressCheckCancel = null, Action<long> progressTryReport = null,  
Action<string, Action> progressExecuteSubtask = null, Func<string, int, string, IDisposable>  
progressBeginSubtask = null)
```

Parameters

configuration [ApiConnectorConfiguration](#)

appDataPath [string](#)

projectStartDate [DateTime](#)?

projectEndDate [DateTime](#)?

lastImportDate [DateTime](#)?

`progressCheckCancel` [Action](#)

`progressTryReport` [Action](#) <[long](#)>

`progressExecuteSubtask` [Action](#) <[string](#), [Action](#)>

`progressBeginSubtask` [Func](#) <[string](#), [int](#), [string](#), [IDisposable](#)>

Properties

Configuration

Configuration of the API DataSource. Has to be cast to the specific Configuration class of the Adapter:

```
Configuration is AdapterConfiguration configuration
```

```
.
```

```
public ApiConnectorConfiguration Configuration { get; }
```

Property Value

[ApiConnectorConfiguration](#)

DownloadPath

The DownloadPath is a cache-directory, where files have to be saved into in the Download-method.

```
public string DownloadPath { get; }
```

Property Value

[string](#)

LastImportDate

Date of the last import, if there was any. Set by the last execution in AdapterReadResult.DateTime and/or AdapterDownloadResult.DateTime. The Adapter should only read data / download files that are newer

than the last import date.

```
public DateTime? LastImportDate { get; }
```

Property Value

[DateTime](#)?

ProjectEndDate

Configured end date of the project. Can be used to filter out data / files.

```
public DateTime? ProjectEndDate { get; }
```

Property Value

[DateTime](#)?

ProjectStartDate

Configured start date of the project. Can be used to filter out data / files.

```
public DateTime? ProjectStartDate { get; }
```

Property Value

[DateTime](#)?

Methods

BeginTask(string, int, string)

BeginTask is a Utility method. Create a task that reports it's steps. A task shows up in the dialog of the frontend. If the total number is not known, the counter can be set to a arbitrary number and only {0} used in the description.

Example:

```
// {0} = current item number, {1} = total number of items
using (context.BeginSubtask("Title", items.Length, "{0} of {1}"))
{
    foreach (var item in items)
    {
        // increase the task counter {0}
        context.ReportProgressForTask();
        // do something
    }
}

public IDisposable BeginTask(string title, int counter, string description)
```

Parameters

title [string](#)

Title of the task.

counter [int](#)

Total number to be counted.

description [string](#)

Description with {0} and/or {1} to show the steps/total.

Returns

[IDisposable](#)

IDisposable, as long as it is not disposed the tasks runs

CheckCancel()

CheckCancel is a Utility method. Should be used in loops of long running operations. If the import was cancelled by the user, this method stops the execution.

```
public void CheckCancel()
```

ExecuteTask(string, Action)

ExecuteTask is a Utility method. Create a task that executes a action. A task shows up in the dialog of the frontend.

```
public void ExecuteTask(string title, Action action)
```

Parameters

title [string](#)

Title of the task.

action [Action](#)

Action to be executed within the task.

ReportProgressForTask(long)

ReportProgressForTask is a Utility method. Increase the progress counter {0}

```
public void ReportProgressForTask(long amount = 1)
```

Parameters

amount [long](#)

Amount to report progress for the task

Class FileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The FileAdapter class is the base for all MQC File Readers.

```
public abstract class FileReader : AdapterBase
```

Inheritance

[object](#) ↗ ← [AdapterBase](#) ← FileReader

Derived

[Adapter](#), [AstreeXmlFileReader](#), [CodeClimateJsonFileReader](#), [DataTemplateFileReader](#),
[DotCoverXmlFileReader](#), [EmbeddedTesterHtmlFileReader](#), [GenericDataSheetFileReader](#),
[JUnitXmlFileReader](#), [MtestHtmlFileReader](#), [MxSuiteXmlFileReader](#), [MxamExcelFileReader](#),
[MxamMxmlFileReader](#), [MxrayHtmlFileReader](#), [MxrayXmlFileReader](#), [PolyspaceCsvFileReader](#),
[PolyspaceXmlFileReader](#), [QacHtmlFileReader](#), [QacXmlFileReader](#), [RtrtHtmlFileReader](#),
[SICheckHtmlFileReader](#), [SIDesignVerifierHtmlFileReader](#), [SIRequirementHtmlFileReader](#),
[TargetLinkHtmlFileReader](#), [TessyXmlFileReader](#), [TestwellCtcHtmlFileReader](#), [TestwellCtcXmlFileReader](#),
[TptHtmlFileReader](#), [TptXmlFileReader](#), [VTestStudioHtmlFileReader](#)

Inherited Members

[AdapterBase.Name](#), [AdapterBase.Description](#), [AdapterBase.DataSource](#), [AdapterBase.Options](#),
[AdapterBase.TransformStringToValue\(string\)](#), [AdapterBase.TransformStringToDate\(string\)](#),
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#),
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#),
,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#), [object.ToString\(\)](#) ↗, [object.Equals\(object\)](#) ↗,
[object.Equals\(object, object\)](#) ↗, [object.ReferenceEquals\(object, object\)](#) ↗, [object.GetHashCode\(\)](#) ↗,
[object.GetType\(\)](#) ↗, [object.MemberwiseClone\(\)](#) ↗

Properties

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public abstract List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

HumanReadableFileCount

Number of Human Readable Files.

```
protected virtual int HumanReadableFileCount { get; }
```

Property Value

[int](#)

HumanReadableFileExtensions

Human Readable File Extensions of the Adapter.

```
protected virtual List<string> HumanReadableFileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

HumanReadableStartFileNames

Start of Human Readable File Names of the Adapter.

```
protected virtual List<string> HumanReadableStartFileNames { get; }
```

Property Value

[List](#)<[string](#)>

Priority

Priority of the Adapter. The execution order of all adapters depend on the defined priorities. The higher the priority the earlier the adapter is validated and executed. The default priority for Base Adapters is between 10-110, Custom Adapters should define a priority of 200 or higher if they should be executed before the base adapters. If two adapters have the same priority, the order is depending on the name of the adapter.

```
public virtual int Priority { get; }
```

Property Value

[int](#)

ReportDateTimeExpected

Is it expected to get a reportDateTime in the report? If this is false the ReportDateTime is the CreationDate of the file without a warning.

```
protected virtual bool ReportDateTimeExpected { get; }
```

Property Value

[bool](#)

Methods

Execute(FileReaderContext, bool)

Internal method

```
public FileReaderResult Execute(FileReaderContext context, bool useFileCreationTime = false)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

`useFileCreationTime bool`

Should the file creationTime always be used as the report date time?

Returns

[FileReaderResult](#)

AdapterValidationResult

GetHumanReadableFilePath(FileReaderContext)

Get the human-readable file path. Can be overridden with a user defined function. Can be used together with GetHumanReadableFilePaths, but it is recommended to only use one of the methods.

```
protected virtual string GetHumanReadableFilePath(FileReaderContext context)
```

Parameters

`context` [FileReaderContext](#)

FileReaderContext

Returns

[string](#)

Human readable file

GetHumanReadableFilePaths(FileReaderContext)

Get the human-readable file paths. Can be overridden with a user defined function. Can be used together with GetHumanReadableFilePath, but it is recommended to only use one of the methods.

```
protected virtual List<string> GetHumanReadableFilePaths(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

Returns

[List](#) <[string](#)>

List of human readable files

IsValid(FileReaderContext)

IsValid has to be implemented by the Adapter class. The method is called when the file extensions match. If the file extension is unique, this method can just return true, otherwise it should check if the file should be imported by the adapter. If true is returned, the current adapter is executed and the Read method is called. No other adapter is checked afterward.

```
protected abstract bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

Returns

[bool](#)

boolean (is this the correct adapter to read the file?)

Read(FileReaderContext)

Read has to be implemented by the Adapter class. This method is called when the file extensions match and isValid returns true, no other adapter is called. The data of the file should be read and returned as a AdapterReadResult.

```
protected abstract AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

Returns

[AdapterReadResult](#)

AdapterReadResult

ReadWithHumanReadableFiles(FileReaderContext, AdapterReadResult, IEnumerable<string>)

ReadWithHumanReadableFiles can be implemented by the Adapter class. This method is called after the Read method, it contains not only the context but also the AdapterReadResult of the Read method and the detected and/or configured HumanReadableFilePaths. The AdapterReadResult can be extended and has to be returned fully.

```
protected virtual AdapterReadResult ReadWithHumanReadableFiles(FileReaderContext context, AdapterReadResult adapterReadResult, IEnumerable<string> humanReadableFilePaths)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

adapterReadResult [AdapterReadResult](#)

AdapterReadResult

humanReadableFilePaths [IEnumerable<string>](#)

Loaded human readable file paths

Returns

[AdapterReadResult](#)

AdapterReadResult

Validate(FileReaderContext)

Internal method

```
public AdapterValidationResult Validate(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

Returns

[AdapterValidationResult](#)

AdapterValidationResult

Class FileReaderContext

Namespace: [MES.MQC.DataSourceLibrary.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class FileReaderContext : AdapterContextBase, IDisposable
```

Inheritance

[object](#) ← [AdapterContextBase](#) ← FileReaderContext

Implements

[IDisposable](#)

Derived

[AdapterContext](#)

Inherited Members

[AdapterContextBase.ImportFindings](#) , [AdapterContextBase.MessageBag](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Constructors

FileReaderContext(string, bool)

```
public FileReaderContext(string filePath, bool importFindings = false)
```

Parameters

filePath [string](#)

importFindings [bool](#)

Properties

Content

File content of the report file.

```
public string Content { get; }
```

Property Value

[string](#)

CreationDate

Modified date of the report file.

```
public DateTime CreationDate { get; }
```

Property Value

[DateTime](#)

DirectoryPath

Directory path of the report file.

```
public string DirectoryPath { get; }
```

Property Value

[string](#)

Extension

File extension of the report file.

```
public string Extension { get; }
```

Property Value

[string](#)

Name

File name of the report file.

```
public string Name { get; }
```

Property Value

[string](#)

NameWithoutExtension

File name without extension of the report file.

```
public string NameWithoutExtension { get; }
```

Property Value

[string](#)

Path

File Path of the report file.

```
public string Path { get; }
```

Property Value

[string](#)

Methods

AsDataSet(int)

AsDataSet is a Utility method. It loads the file path as an DataSet. The DataSet is stored in a property and returned. The reading of the file is therefore only done once.

```
public DataSet AsDataSet(int removeRows = 0)
```

Parameters

removeRows [int](#)

Number of rows to remove before parsing (e.g. invalid/merged rows above the header)

Returns

[DataSet](#)

DataSet (with DataTables, System.Data)

AsHtmlDocument(string)

AsHtmlDocument is a Utility method. It loads the file path as an HtmlDocument (HtmlAgilityPack). See <https://html-agility-pack.net/> for documentation. The document is stored in a property and returned. The reading of the file is therefore only done once.

```
public HtmlDocument AsHtmlDocument(string content = null)
```

Parameters

content [string](#)

Content of a html file, optional

Returns

HtmlDocument

HtmlDocument (HtmlAgilityPack)

AsXDocument()

It loads the file path as an XDocument, which can be traversed via XPath. See <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/linq-to-xml-overview> for documentation. The document is stored in a property and returned. The reading of the file is therefore only done once.

```
public XDocument AsXDocument()
```

Returns

[XDocument](#)

XDocument (System.Xml)

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

```
public void Dispose()
```

Namespace MES.MQC.DataSourceLibrary. Adapters.Apis

Classes

[GitlabApiConnector](#)

[GitlabPipelineApiConnector](#)

[GitlabPipelineArtifactsApiConnector](#)

[GitlabPipelineJUnitApiConnector](#)

[GitlabRepositoryApiConnector](#)

[JenkinsPipelineArtifactsApiConnector](#)

Class GitlabApiConnector

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Apis](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public abstract class GitlabApiConnector : ApiConnector, IFormProvider
```

Inheritance

[object](#) ← [AdapterBase](#) ← [ApiConnector](#) ← [GitlabApiConnector](#)

Implements

[IFormProvider](#)

Derived

[GitlabPipelineApiConnector](#), [GitlabRepositoryApiConnector](#)

Inherited Members

[ApiConnector.DataSource](#), [ApiConnector.PreviewForm\(IForm, string, out int\)](#),
[ApiConnector.CheckAvailable\(ApiConnectorContext\)](#),
[ApiConnector.CheckModified\(ApiConnectorContext\)](#), [ApiConnector.Read\(ApiConnectorContext\)](#),
[ApiConnector.Download\(ApiConnectorContext\)](#), [ApiConnector.Execute\(ApiConnectorContext\)](#),
[AdapterBase.Name](#), [AdapterBase.Description](#), [AdapterBase.Options](#),
[AdapterBase.TransformStringToValue\(string\)](#), [AdapterBase.TransformStringToDateTIme\(string\)](#),
[AdapterBase.TransformStringToDateTIme\(AdapterContext, string\)](#),
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#),
,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#), [object.ToString\(\)](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

Methods

ConfigureForm(IForm, string[])

ConfigureForm can be implemented by an API adapter. It is used to modify the FormSchema of the Add/Edit DataSource dialog for the AdapterConfiguration depending on the values entered (e.g. to make options or input fields visible and/or deactivate them based on a correct API request). It is called for the initial creation of the FormSchema with all fields as modified and when a field that was enabled by RefreshOnModified has been changed by the user. This method should also validate the configuration,

not if the schema is valid, but if the values are usable (e.g. if the api is accessible and the authentication is valid).

```
public override FormError[] ConfigureForm(IForm form, string[] modifiedFields)
```

Parameters

form [IForm](#)

IForm, the AdapterConfiguration

modifiedFields [string](#)[]

Array of the modified fields as string (Form-Subfields as field.field or field.index.field)

Returns

[FormError](#)[]

Array of FormError, if a validation failed

GetClient(ApiConnectorContext, bool)

```
protected RestClient GetClient(ApiConnectorContext context, bool checkAuth = false)
```

Parameters

context [ApiConnectorContext](#)

checkAuth [bool](#)

Returns

RestClient

GetClient(GitlabConfiguration, out List<FormError>, bool)

```
protected RestClient GetClient(GitlabApiConnector.GitlabConfiguration configuration, out List<FormError> errors, bool checkAuth = false)
```

Parameters

configuration [GitlabApiConnector.GitlabConfiguration](#)

errors [List<FormError>](#)

checkAuth [bool](#)

Returns

RestClient

Request<T>(RestClient, RestRequest)

protected T[] [Request<T>\(RestClient client, RestRequest request\)](#)

Parameters

client RestClient

request RestRequest

Returns

T[]

Type Parameters

T

Request<T>(RestClient, RestRequest, out RestResponse, int, int?)

protected T[] [Request<T>\(RestClient client, RestRequest request, **out** RestResponse response, int page = 1, **int?** limit = null\)](#)

Parameters

client RestClient

request RestRequest

response RestResponse

page [int](#)

limit [int](#)?

Returns

T[]

Type Parameters

T

Class GitlabPipelineApiConnector

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Apis](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public abstract class GitlabPipelineApiConnector : GitlabApiConnector, IFormProvider
```

Inheritance

[object](#) ← [AdapterBase](#) ← [ApiConnector](#) ← [GitlabApiConnector](#) ← [GitlabPipelineApiConnector](#)

Implements

[IFormProvider](#)

Derived

[GitlabPipelineArtifactsApiConnector](#), [GitlabPipelineJUnitApiConnector](#)

Inherited Members

[GitlabApiConnector.GetClient\(ApiConnectorContext, bool\)](#),
[GitlabApiConnector.GetClient\(GitlabApiConnector.GitlabConfiguration, out List<FormError>, bool\)](#),
[GitlabApiConnector.Request<T>\(RestClient, RestRequest\)](#),
[GitlabApiConnector.Request<T>\(RestClient, RestRequest, out RestResponse, int, int?\)](#),
[GitlabApiConnector.ConfigureForm\(IForm, string\[\]\)](#), [ApiConnector.DataSource](#),
[ApiConnector.Read\(ApiConnectorContext\)](#), [ApiConnector.Download\(ApiConnectorContext\)](#),
[ApiConnector.Execute\(ApiConnectorContext\)](#), [AdapterBase.Name](#), [AdapterBase.Description](#),
[AdapterBase.Options](#), [AdapterBase.TransformStringToValue\(string\)](#),
[AdapterBase.TransformStringToDate\(string\)](#),
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#),
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#),

[AdapterBase.ReadFilePathsInZipFile\(string\)](#), [object.ToString\(\)](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

Methods

CheckAvailable(ApiConnectorContext)

CheckAvailable must be implemented by the Api Adapter class. This method is called to show a warning for configured but unavailable data sources in the UI.

```
public override bool CheckAvailable(ApiConnectorContext context)
```

Parameters

context [ApiConnectorContext](#)

ApiConnectorContext

Returns

[bool](#)

boolean (is the api available/accessible)

CheckModified(ApiConnectorContext)

CheckModified must be implemented by the Api Adapter class. This method is called by the local and server side monitoring, to check if the data sources should be updated.

```
public override bool CheckModified(ApiConnectorContext context)
```

Parameters

context [ApiConnectorContext](#)

ApiConnectorContext

Returns

[bool](#)

boolean (are there any changes)

LoadJobs(GitlabPipelineConfiguration, RestClient, Gitlab Pipeline)

```
protected GitlabPipelineApiConnector.GitlabJob[]
```

```
LoadJobs(GitlabPipelineApiConnector.GitlabPipelineConfiguration configuration, RestClient
```

```
client, GitlabPipelineApiConnector.GitlabPipeline pipeline)
```

Parameters

configuration [GitlabPipelineApiConnector.GitlabPipelineConfiguration](#)

client RestClient

pipeline [GitlabPipelineApiConnector.GitlabPipeline](#)

Returns

[GitlabJob\[\]](#)

LoadPipelines(ApiConnectorContext, GitlabPipelineConfiguration, RestClient, out DateTime, int?)

```
protected GitlabPipelineApiConnector.GitlabPipeline[] LoadPipelines(ApiConnectorContext context, GitlabPipelineApiConnector.GitlabPipelineConfiguration configuration, RestClient client, out DateTime responseDateTime, int? limit = null)
```

Parameters

context [ApiConnectorContext](#)

configuration [GitlabPipelineApiConnector.GitlabPipelineConfiguration](#)

client RestClient

responseDateTime [DateTime](#)

limit [int](#)?

Returns

[GitlabPipeline\[\]](#)

LoadPipelines(GitlabPipelineConfiguration, RestClient, ProjectBranch, out DateTime, out int, int?, DateTime?, DateTime?, DateTime?)

Time?)

```
protected GitlabPipelineApiConnector.GitlabPipeline[]  
LoadPipelines(GitlabPipelineApiConnector.GitlabPipelineConfiguration configuration,  
RestClient client, GitlabApiConnector.ProjectBranch projectBranch, out DateTime  
responseDateTime, out int totalCount, int? limit = null, DateTime? lastImportDate = null,  
DateTime? projectStartDate = null, DateTime? projectEndDate = null)
```

Parameters

configuration [GitlabPipelineApiConnector.GitlabPipelineConfiguration](#)

client RestClient

projectBranch [GitlabApiConnector.ProjectBranch](#)

responseDateTime [DateTime](#)?

totalCount [int](#)?

limit [int](#)? ?

lastImportDate [DateTime](#)?

projectStartDate [DateTime](#)?

projectEndDate [DateTime](#)?

Returns

[GitlabPipeline](#)[]

PreviewForm(IForm, string, out int)

PreviewForm can be implemented by an API Connector. It is used to load the Preview, if one was defined in the AdapterConfiguration. Based on the focused form field, the Preview is shown in the Dialog and provides the ability to load by clicking on a button. This button click calls this method. Based on the configuration and the preview type, the api should load the relevant data, limited to ~100 for better performance, and a array of FormPreview should be returned.

```
public override FormPreview[] PreviewForm(IForm form, string preview, out int totalCount)
```

Parameters

form [IForm](#)

IForm, the AdapterConfiguration

preview [string](#)

The preview type (string)

totalCount [int](#)

Output of totalCount, if 0 shown as ? in the preview dialog

Returns

[FormPreview\[\]](#)

Array of FormPreview, to show Title and optionally Description and DateTime

Class GitlabPipelineArtifactsApiConnector

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Apis](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class GitlabPipelineArtifactsApiConnector : GitlabPipelineApiConnector, IFormProvider
```

Inheritance

[object](#) ← [AdapterBase](#) ← [ApiConnector](#) ← [GitlabApiConnector](#) ← [GitlabPipelineApiConnector](#) ← [GitlabPipelineArtifactsApiConnector](#)

Implements

[IFormProvider](#)

Inherited Members

[GitlabPipelineApiConnector.CheckAvailable\(ApiConnectorContext\)](#),
[GitlabPipelineApiConnector.CheckModified\(ApiConnectorContext\)](#),
[GitlabPipelineApiConnector.LoadPipelines\(ApiConnectorContext, GitlabPipelineApiConnector.GitlabPipelineConfiguration, RestClient, out DateTime, int?\)](#),
[GitlabPipelineApiConnector.LoadPipelines\(GitlabPipelineApiConnector.GitlabPipelineConfiguration, RestClient, GitlabApiConnector.ProjectBranch, out DateTime, out int, int?, DateTime?, DateTime?, DateTime?, DateTime?\)](#),
[GitlabPipelineApiConnector.LoadJobs\(GitlabPipelineApiConnector.GitlabPipelineConfiguration, RestClient, GitlabPipelineApiConnector.GitlabPipeline\)](#),
[GitlabPipelineApiConnector.PreviewForm\(IForm, string, out int\)](#),
[GitlabApiConnector.GetClient\(ApiConnectorContext, bool\)](#),
[GitlabApiConnector.GetClient\(GitlabApiConnector.GitlabConfiguration, out List<FormError>, bool\)](#),
[GitlabApiConnector.Request<T>\(RestClient, RestRequest\)](#),
[GitlabApiConnector.Request<T>\(RestClient, RestRequest, out RestResponse, int, int?\)](#),
[GitlabApiConnector.ConfigureForm\(IForm, string\[\]\)](#), [ApiConnector.DataSource](#),
[ApiConnector.Read\(ApiConnectorContext\)](#), [ApiConnector.Execute\(ApiConnectorContext\)](#),
[AdapterBase.Options](#), [AdapterBase.TransformStringToValue\(string\)](#),
[AdapterBase.TransformStringToDate\(string\)](#),
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#),
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#),

[AdapterBase.ReadFilePathsInZipFile\(string\)](#), [object.ToString\(\)](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

Properties

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#) ↗

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#) ↗

Methods

Download(ApiConnectorContext)

Download can be implemented by the Api Adapter class. At least one of Download and Read must be implemented. The implementation of this method should access the configured API, download and save files to the context.DownloadPath directory and return a AdapterDownloadResult.

```
protected override AdapterDownloadResult Download(ApiConnectorContext context)
```

Parameters

`context` [ApiConnectorContext](#)

ApiConnectorContext

Returns

[AdapterDownloadResult](#)

AdapterDownloadResult

Class GitlabPipelineJUnitApiConnector

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Apis](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class GitlabPipelineJUnitApiConnector : GitlabPipelineApiConnector, IFormProvider
```

Inheritance

[object](#) ← [AdapterBase](#) ← [ApiConnector](#) ← [GitlabApiConnector](#) ← [GitlabPipelineApiConnector](#) ← [GitlabPipelineJUnitApiConnector](#)

Implements

[IFormProvider](#)

Inherited Members

[GitlabPipelineApiConnector.CheckAvailable\(ApiConnectorContext\)](#),
[GitlabPipelineApiConnector.CheckModified\(ApiConnectorContext\)](#),
[GitlabPipelineApiConnector.LoadPipelines\(ApiConnectorContext, GitlabPipelineApiConnector.GitlabPipelineConfiguration, RestClient, out DateTime, int?\)](#),
[GitlabPipelineApiConnector.LoadPipelines\(GitlabPipelineApiConnector.GitlabPipelineConfiguration, RestClient, GitlabApiConnector.ProjectBranch, out DateTime, out int, int?, DateTime?, DateTime?, DateTime?, DateTime?\)](#),
[GitlabPipelineApiConnector.LoadJobs\(GitlabPipelineApiConnector.GitlabPipelineConfiguration, RestClient, GitlabPipelineApiConnector.GitlabPipeline\)](#),
[GitlabPipelineApiConnector.PreviewForm\(IForm, string, out int\)](#),
[GitlabApiConnector.GetClient\(ApiConnectorContext, bool\)](#),
[GitlabApiConnector.GetClient\(GitlabApiConnector.GitlabConfiguration, out List<FormError>, bool\)](#),
[GitlabApiConnector.Request<T>\(RestClient, RestRequest\)](#),
[GitlabApiConnector.Request<T>\(RestClient, RestRequest, out RestResponse, int, int?\)](#),
[GitlabApiConnector.ConfigureForm\(IForm, string\[\]\)](#), [ApiConnector.DataSource](#),
[ApiConnector.Download\(ApiConnectorContext\)](#), [ApiConnector.Execute\(ApiConnectorContext\)](#),
[AdapterBase.Options](#), [AdapterBase.TransformStringToValue\(string\)](#),
[AdapterBase.TransformStringToDate\(string\)](#),
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#),
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#),

[AdapterBase.ReadFilePathsInZipFile\(string\)](#), [object.ToString\(\)](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

Properties

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#) ↗

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#) ↗

Methods

Read(ApiConnectorContext)

Read can be implemented by the Api Adapter class. At least one of Download and Read must be implemented. The implementation of this method should access the configured API, create AdapterData, AdapterMeasures and/or AdapterFindings and return a AdapterReadResult.

```
protected override AdapterReadResult Read(ApiConnectorContext context)
```

Parameters

`context` [ApiConnectorContext](#)

`ApiConnectorContext`

Returns

[AdapterReadResult](#)

`AdapterReadResult`

Class GitlabRepositoryApiConnector

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Apis](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class GitlabRepositoryApiConnector : GitlabApiConnector, IFormProvider
```

Inheritance

[object](#) ← [AdapterBase](#) ← [ApiConnector](#) ← [GitlabApiConnector](#) ← [GitlabRepositoryApiConnector](#)

Implements

[IFormProvider](#)

Inherited Members

[GitlabApiConnector.GetClient\(ApiConnectorContext, bool\)](#),
[GitlabApiConnector.GetClient\(GitlabApiConnector.GitlabConfiguration, out List<FormError>, bool\)](#),
[GitlabApiConnector.Request<T>\(RestClient, RestRequest\)](#),
[GitlabApiConnector.Request<T>\(RestClient, RestRequest, out RestResponse, int, int?\)](#),
[GitlabApiConnector.ConfigureForm\(IForm, string\[\]\)](#), [ApiConnector.DataSource](#),
[ApiConnector.Read\(ApiConnectorContext\)](#), [ApiConnector.Execute\(ApiConnectorContext\)](#),
[AdapterBase.Options](#), [AdapterBase.TransformStringToValue\(string\)](#),
[AdapterBase.TransformStringToDate\(string\)](#),
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#),
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#),
,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#), [object.ToString\(\)](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#),
[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#)

Properties

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

CheckAvailable(ApiConnectorContext)

CheckAvailable must be implemented by the Api Adapter class. This method is called to show a warning for configured but unavailable data sources in the UI.

```
public override bool CheckAvailable(ApiConnectorContext context)
```

Parameters

context [ApiConnectorContext](#)

ApiConnectorContext

Returns

[bool](#)

boolean (is the api available/accessible)

CheckModified(ApiConnectorContext)

CheckModified must be implemented by the Api Adapter class. This method is called by the local and server side monitoring, to check if the data sources should be updated.

```
public override bool CheckModified(ApiConnectorContext context)
```

Parameters

context [ApiConnectorContext](#)

ApiConnectorContext

Returns

[bool](#)

boolean (are there any changes)

Download(ApiConnectorContext)

Download can be implemented by the Api Adapter class. At least one of Download and Read must be implemented. The implementation of this method should access the configured API, download and save files to the context.DownloadPath directory and return a AdapterDownloadResult.

```
protected override AdapterDownloadResult Download(ApiConnectorContext context)
```

Parameters

context [ApiConnectorContext](#)

ApiConnectorContext

Returns

[AdapterDownloadResult](#)

AdapterDownloadResult

PreviewForm(IForm, string, out int)

PreviewForm can be implemented by an API Connector. It is used to load the Preview, if one was defined in the AdapterConfiguration. Based on the focused form field, the Preview is shown in the Dialog and provides the ability to load by clicking on a button. This button click calls this method. Based on the configuration and the preview type, the api should load the relevant data, limited to ~100 for better performance, and a array of FormPreview should be returned.

```
public override FormPreview[] PreviewForm(IForm form, string preview, out int totalCount)
```

Parameters

form [IForm](#)

IForm, the AdapterConfiguration

preview [string](#) ↗

The preview type (string)

totalCount [int](#) ↗

Output of totalCount, if 0 shown as ? in the preview dialog

Returns

[FormPreview\[\]](#)

Array of FormPreview, to show Title and optionally Description and DateTime

Class JenkinsPipelineArtifactsApiConnector

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Apis](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class JenkinsPipelineArtifactsApiConnector : ApiConnector, IFormProvider
```

Inheritance

[object](#) ← [AdapterBase](#) ← [ApiConnector](#) ← JenkinsPipelineArtifactsApiConnector

Implements

[IFormProvider](#)

Inherited Members

[ApiConnector.DataSource](#) , [ApiConnector.Read\(ApiConnectorContext\)](#) ,
[ApiConnector.Execute\(ApiConnectorContext\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDateTIme\(string\)](#) ,
[AdapterBase.TransformStringToDateTIme\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

CheckAvailable(ApiConnectorContext)

CheckAvailable must be implemented by the Api Adapter class. This method is called to show a warning for configured but unavailable data sources in the UI.

```
public override bool CheckAvailable(ApiConnectorContext context)
```

Parameters

[context](#) [ApiConnectorContext](#)

ApiConnectorContext

Returns

[bool](#)

boolean (is the api available/accessible)

CheckModified(ApiConnectorContext)

CheckModified must be implemented by the Api Adapter class. This method is called by the local and server side monitoring, to check if the data sources should be updated.

```
public override bool CheckModified(ApiConnectorContext context)
```

Parameters

context [ApiConnectorContext](#)

ApiConnectorContext

Returns

[bool](#)

boolean (are there any changes)

ConfigureForm(IForm, string[])

ConfigureForm can be implemented by an API adapter. It is used to modify the FormSchema of the Add/Edit DataSource dialog for the AdapterConfiguration depending on the values entered (e.g. to make options or input fields visible and/or deactivate them based on a correct API request). It is called for the initial creation of the FormSchema with all fields as modified and when a field that was enabled by RefreshOnModified has been changed by the user. This method should also validate the configuration, not if the schema is valid, but if the values are usable (e.g. if the api is accessible and the authentication is valid).

```
public override FormError[] ConfigureForm(IForm form, string[] modifiedFields)
```

Parameters

form [IForm](#)

IForm, the AdapterConfiguration

modifiedFields [string\[\]](#)

Array of the modified fields as string (Form-Subfields as field.field or field.index.field)

Returns

[FormError\[\]](#)

Array of FormError, if a validation failed

Download(ApiConnectorContext)

Download can be implemented by the Api Adapter class. At least one of Download and Read must be implemented. The implementation of this method should access the configured API, download and save files to the context.DownloadPath directory and return a AdapterDownloadResult.

```
protected override AdapterDownloadResult Download(ApiConnectorContext context)
```

Parameters

context [ApiConnectorContext](#)

ApiConnectorContext

Returns

[AdapterDownloadResult](#)

AdapterDownloadResult

GetClient(ApiConnectorContext, bool)

```
protected RestClient GetClient(ApiConnectorContext context, bool checkAuth = false)
```

Parameters

context [ApiConnectorContext](#)

checkAuth [bool](#) ↗

Returns

RestClient

GetClient(JenkinsPipelineArtifactsConfiguration, out List<FormError>, bool)

```
protected RestClient  
GetClient(JenkinsPipelineArtifactsApiConnector.JenkinsPipelineArtifactsConfiguration  
configuration, out List<FormError> errors, bool checkAuth = false)
```

Parameters

configuration [JenkinsPipelineArtifactsApiConnector.JenkinsPipelineArtifactsConfiguration](#)

errors [List<FormError>](#)

checkAuth [bool](#)

Returns

RestClient

GetFilterDateRange(JenkinsPipelineArtifactsConfiguration, ApiConnectorContext, out long?, out long?, out DateTime?, out DateTime?)

```
protected void  
GetFilterDateRange(JenkinsPipelineArtifactsApiConnector.JenkinsPipelineArtifactsConfiguration  
configuration, ApiConnectorContext context, out long? unixStartDate, out long?  
unixEndDate, out DateTime? startDate, out DateTime? endDate)
```

Parameters

configuration [JenkinsPipelineArtifactsApiConnector.JenkinsPipelineArtifactsConfiguration](#)

context [ApiConnectorContext](#)

unixStartDate [long](#)?

unixEndDate [long](#)?

startDate [DateTime](#)?

`endDate` [DateTime](#)?

`LoadBuilds(JenkinsPipelineArtifactsConfiguration, RestClient, out
DateTime, ApiConnectorContext, int?)`

```
protected JenkinsPipelineArtifactsApiConnector.JenkinsBuild[]  
LoadBuilds(JenkinsPipelineArtifactsApiConnector.JenkinsPipelineArtifactsConfiguration  
configuration, RestClient client, out DateTime responseDateTime, ApiConnectorContext context  
= null, int? limit = null)
```

Parameters

`configuration` [JenkinsPipelineArtifactsApiConnector.JenkinsPipelineArtifactsConfiguration](#)

`client` RestClient

`responseDateTime` [DateTime](#)

`context` [ApiConnectorContext](#)

`limit` [int](#)?

Returns

[JenkinsBuild](#)[]

`LoadBuildsForJob(RestClient, string, long?, long?, int?)`

```
protected List<JenkinsPipelineArtifactsApiConnector.JenkinsBuild>  
LoadBuildsForJob(RestClient client, string jobName, long? unixStartDate = null, long?  
unixEndDate = null, int? limit = null)
```

Parameters

`client` RestClient

`jobName` [string](#)

`unixStartDate` [long](#)?

`unixEndDate` `long`?

`limit` `int`?

Returns

`List<JenkinsPipelineArtifactsApiConnector.JenkinsBuild>`

LoadJobNames(RestClient)

`protected string[] LoadJobNames(RestClient client)`

Parameters

`client` RestClient

Returns

`string[]`

PreviewForm(IForm, string, out int)

PreviewForm can be implemented by an API Connector. It is used to load the Preview, if one was defined in the AdapterConfiguration. Based on the focused form field, the Preview is shown in the Dialog and provides the ability to load by clicking on a button. This button click calls this method. Based on the configuration and the preview type, the api should load the relevant data, limited to ~100 for better performance, and a array of FormPreview should be returned.

`public override FormPreview[] PreviewForm(IForm form, string preview, out int totalCount)`

Parameters

`form` `IForm`

IForm, the AdapterConfiguration

`preview` `string`

The preview type (string)

totalCount [int](#)

Output of totalCount, if 0 shown as ? in the preview dialog

Returns

[FormPreview\[\]](#)

Array of FormPreview, to show Title and optionally Description and DateTime

Namespace MES.MQC.DataSourceLibrary.Adapters.Files

Classes

[AstreeXmlFileReader](#)

[CodeClimateJsonFileReader](#)

[DataTemplateFileReader](#)

[DotCoverXmlFileReader](#)

[EmbeddedTesterHtmlFileReader](#)

[GenericDataSheetFileReader](#)

[JUnitXmlFileReader](#)

[MtestHtmlFileReader](#)

[MxSuiteXmlFileReader](#)

[MxamExcelFileReader](#)

[MxamMxmrFileReader](#)

[MxrayHtmlFileReader](#)

[MxrayXmlFileReader](#)

[PolyspaceCsvFileReader](#)

[PolyspaceXmlFileReader](#)

[QacHtmlFileReader](#)

[QacXmlFileReader](#)

[RtrtHtmlFileReader](#)

[SICheckHtmlFileReader](#)

[SIDesignVerifierHtmlFileReader](#)

[SIRequirementHtmlFileReader](#)

[TargetLinkHtmlFileReader](#)

[TessyXmlFileReader](#)

[TestwellCtcHtmlFileReader](#)

[TestwellCtcXmlFileReader](#)

[IptHtmlFileReader](#)

[IptXmlFileReader](#)

[VTestStudioHtmlFileReader](#)

Class AstreeXmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class AstreeXmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← [AstreeXmlFileReader](#)

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDateTIme\(string\)](#) ,
[AdapterBase.TransformStringToDateTIme\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

GetHumanReadableFilePaths(FileReaderContext)

Get the human-readable file paths. Can be overridden with a user defined function. Can be used together with GetHumanReadableFilePath, but it is recommended to only use one of the methods.

```
protected override List<string> GetHumanReadableFilePaths(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

Returns

[List](#)<[string](#)>

List of human readable files

IsValid(FileReaderContext)

Check if element exists in Document

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid Astree report file

Read(FileReaderContext)

Read 'errors','runtime_errors','flow_anomalies','data_races','rule_violations' attributes from summary element inside analysis. Also read 'statement_count','not_reached_count' attributes from 'coverage' element inside analysis.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

Class CodeClimateJsonFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class CodeClimateJsonFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← CodeClimateJsonFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Description](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformString.ToDateTime\(string\)](#) ,
[AdapterBase.TransformString.ToDateTime\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

ReportDateTimeExpected

Is it expected to get a reportDateTime in the report? If this is false the ReportDateTime is the CreationDate of the file without a warning.

```
protected override bool ReportDateTimeExpected { get; }
```

Property Value

[bool](#)

Methods

IsValid(FileReaderContext)

IsValid has to be implemented by the Adapter class. The method is called when the file extensions match. If the file extension is unique, this method can just return true, otherwise it should check if the file should be imported by the adapter. If true is returned, the current adapter is executed and the Read method is called. No other adapter is checked afterward.

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

Returns

[bool](#)

boolean (is this the correct adapter to read the file?)

Read(FileReaderContext)

Read has to be implemented by the Adapter class. This method is called when the file extensions match and isValid returns true, no other adapter is called. The data of the file should be read and returned as a AdapterReadResult.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

Returns

[AdapterReadResult](#)

AdapterReadResult

Class DataTemplateFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class DataTemplateFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← DataTemplateFileReader

Inherited Members

[FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformString.ToDateTime\(string\)](#) ,
[AdapterBase.TransformString.ToDateTime\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Priority

Lower priority than base adapters, the default is 100, for the manual import template

```
public override int Priority { get; }
```

Property Value

[int](#)

Methods

IsValid(FileReaderContext)

Check if the file name does not contain "Configuration"

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid manual import file

Read(FileReaderContext)

There are 2 excel formats that can be imported by this Adapter : First format has this headers (naming and order of columns is important): Artifact|ReportDateTime|DataSource|Measure|Variable|Value autopilot|20.12.2017|MXAM|FindingCount|Aborted|3 Second format has this headers (naming and order of the first two columns is important):

Artifact|ReportDateTime|DataSource.Measurement.Measure.Variable|... autopilot|20.12.2017|3|... means that number of columns is in range 3...n and columns format 2 onwards is like

"DataSource.Measurement.Measure.Variable", number and order of items is important. Each Excel file can contain more than one sheet, each sheet may have different format

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

example of header for excel in format 2 : Artifact|ReportDateTime|MTest.ModelTest.Assessment Work Progress.Reached|MXAM.GuidelineAnalysis.FindingCount.Aborted

Class DotCoverXmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class DotCoverXmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← DotCoverXmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDateTIme\(string\)](#) ,
[AdapterBase.TransformStringToDateTIme\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

ReportDateTimeExpected

Is it expected to get a reportDateTime in the report? If this is false the ReportDateTime is the CreationDate of the file without a warning.

```
protected override bool ReportDateTimeExpected { get; }
```

Property Value

[bool](#)

Methods

IsValid(FileReaderContext)

Check if the Root element has DotCoverVersion attribute

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

[context](#) [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid dotCover report file

Read(FileReaderContext)

Parses the dotCover XML file to find the coverage values for the namespace elements.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

`context FileReaderContext`

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

Class EmbeddedTesterHtmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class EmbeddedTesterHtmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← EmbeddedTesterHtmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDate\(string\)](#) ,
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if the file contains 'Requirement-based Testing' (for version = 3) Check if the file contains '' for MiL or SiL report (for version = 2) Check if the file contains '' for B2B report (for version = 2)

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid EmbeddedTester report file

Read(FileReaderContext)

This function calls separate functions to fetch: ReadReportVersion ReadArtifact and ReportDateTime TestCase results and findings (MiL and/or SiL) B2B test results and findings Requirement results and findings StructuralCoverage results and findings (model and code)

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Class GenericDataSheetFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class GenericDataSheetFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← GenericDataSheetFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableStartFileNames](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDateTIme\(string\)](#) ,
[AdapterBase.TransformStringToDateTIme\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

HumanReadableFileExtensions

No human readable auto detect in same directory

```
protected override List<string> HumanReadableFileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

ReportDateTimeExpected

Is it expected to get a reportDateTime in the report? If this is false the ReportDateTime is the CreationDate of the file without a warning.

```
protected override bool ReportDateTimeExpected { get; }
```

Property Value

[bool](#)

Methods

IsValid(FileReaderContext)

Read adapter options and Check if the file name matches with the FileExpression adapter option.

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

[context](#) [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid GenericTableData Excel file based on file expression

Read(FileReaderContext)

Merge data tables from different sheets, filter rows, read rows as adapter data and then aggregate based on adapter options.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Class JUnitXmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class JUnitXmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← JUnitXmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDate\(string\)](#) ,
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

IsValid has to be implemented by the Adapter class. The method is called when the file extensions match. If the file extension is unique, this method can just return true, otherwise it should check if the file should be imported by the adapter. If true is returned, the current adapter is executed and the Read method is called. No other adapter is checked afterward.

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

Returns

[bool](#) ↗

boolean (is this the correct adapter to read the file?)

Read(FileReaderContext)

Read has to be implemented by the Adapter class. This method is called when the file extensions match and isValid returns true, no other adapter is called. The data of the file should be read and returned as a AdapterReadResult.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

Returns

[AdapterReadResult](#)

AdapterReadResult

Class MtestHtmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class MtestHtmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← MtestHtmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDateTIme\(string\)](#) ,
[AdapterBase.TransformStringToDateTIme\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Fields

TestcaseCatalogStartFileName

```
public const string TestcaseCatalogStartFileName = "Testcatalog_"
```

Field Value

[string](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

HumanReadableFileExtensions

Human Readable File Extensions of the Adapter.

```
protected override List<string> HumanReadableFileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

HumanReadableStartFileNames

Start of Human Readable File Names of the Adapter.

```
protected override List<string> HumanReadableStartFileNames { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

`IsValid` has to be implemented by the Adapter class. The method is called when the file extensions match. If the file extension is unique, this method can just return true, otherwise it should check if the file should be imported by the adapter. If true is returned, the current adapter is executed and the `Read` method is called. No other adapter is checked afterward.

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

`context` [FileReaderContext](#)

FileReaderContext

Returns

[bool](#)

boolean (is this the correct adapter to read the file?)

LoadHtmlDocument(string, string)

```
public static HtmlDocument LoadHtmlDocument(string mainFilePath, string filePath)
```

Parameters

`mainFilePath` [string](#)

`filePath` [string](#)

Returns

HtmlDocument

LoadHtmlDocuments(string, out HtmlDocument, out Html Document, out HtmlDocument, out string, out string, out string)

```
public static void LoadHtmlDocuments(string mainFilePath, out HtmlDocument requirementDocument, out HtmlDocument assessmentDocument, out HtmlDocument testcaseDocument,
```

```
out string requirementCatalogFilePath, out string assessmentCatalogFilePath, out  
string testcaseCatalogFilePath)
```

Parameters

mainFilePath [string](#)

requirementDocument HtmlDocument

assessmentDocument HtmlDocument

testcaseDocument HtmlDocument

requirementCatalogFilePath [string](#)

assessmentCatalogFilePath [string](#)

testcaseCatalogFilePath [string](#)

Read(FileReaderContext)

Read has to be implemented by the Adapter class. This method is called when the file extensions match and isValid returns true, no other adapter is called. The data of the file should be read and returned as a AdapterReadResult.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

Returns

[AdapterReadResult](#)

AdapterReadResult

Class MxSuiteXmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class MxSuiteXmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← MxSuiteXmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDate\(string\)](#) ,
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if the file name is equal to "Report.RegResults.xml"

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid MxSuite report file

Read(FileReaderContext)

Parses the MxSuite XML file to find artifacts and measures from Regression tags as described below

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

```
<Regression> ArtifactPath : <Overview> <Project> <Name>xxx</Name> => ArtifactPath = xxx  
MeasureName = Scenario\Result , Variable = Total, Passed, Failed, Not Executed (mapped Skipped to  
that), Error (mapped RunTimeError to that), Missing <StatisticsTotals Total="1" Passed="0" Failed="1"  
Skipped="0" RunTimeError="0" Missing="0" ...> MeasureName = Testcase\Result , Variable = Total,  
Passed, Failed, Not Executed (mapped Skipped to that), Error (mapped RunTimeError to that), Missing  
<Testcases Total="1" Passed="0" Failed="1" Skipped="0" RunTimeError="0" Missing="0" .../>  
</StatisticsTotals> </Regression>
```

Class MxamExcelFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class MxamExcelFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← MxamExcelFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDate\(string\)](#) ,
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be set to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if the excel contain "Project Overview" sheet

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid MXAM Excel file

Read(FileReaderContext)

From 'Project Overview' sheet fetch 'ReportDateTime'. From 'Findings' sheets fetch list of MxamExcelDataModel. For setting MeasureName = FindingCount : group all findings by MeasurementName, ArtifactName and VariableName then value is the count of group. For setting MeasureName = GuidelineCount : first group all findings by MeasurementName, ArtifactName and GuidelineName then VariableName is the worst finding of group, then group result by MeasurementName, ArtifactName and VariableName then value is the count of group.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Class MxamMxmrFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class MxamMxmrFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← MxamMxmrFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDateTime\(string\)](#) ,
[AdapterBase.TransformStringToDateTime\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if file contains '<RReport:RReport' and '<artifacts'

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid MXAM report file

Read(FileReaderContext)

Read data from different functions based on version of report and if report contains one or more artifacts. Read report data time from date attribute of root element.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

<RReport:RReport xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" date="2018-08-02T13:34:21.136+0200"> </RReport:RReport>

Class MxrayHtmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class MxrayHtmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← MxrayHtmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDate\(string\)](#) ,
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if the 'reportTitle' div element is equal to 'M-XRAY Report'

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid MXRAY html report file

Examples

M-XRAY Report

Read(FileReaderContext)

Fetch Info from SingleSystemsResults chapter of an MXRay Html report. Reports containing multiple artifacts are supported as well.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in a list of AdapterData Model objects

Class MxrayXmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class MxrayXmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← MxrayXmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDate\(string\)](#) ,
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if the root element is 'M-XRAY_Report'

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid MXRAY report file

Read(FileReaderContext)

Read each 'System' elements inside 'SingleSystemResults' and fetch value of 'FullPathOfFile' as ArtifactPath (file name is used as ArtifactPath measures for this artifact fetch from 3 functions: ReadQualityMetrics, ReadCloneGroupSummary, ReadGlobalValueSummary

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

```
<M-XRAY_Report schemaVersion="2.0" xmlns="http://www.model-engineers.com/mxrayXMLSchema">  
C:\Program Files\MTest_5_2\demo\R2016b\EV3Control_ec\Model\EV3Control_demo_ec.slx =>  
ArtifactPath = 'EV3Control_demo_ec' </M-XRAY_Report>
```

Class PolyspaceCsvFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class PolyspaceCsvFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← PolyspaceCsvFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDateTIme\(string\)](#) ,
[AdapterBase.TransformStringToDateTIme\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

ReportDateTimeExpected

Is it expected to get a reportDateTime in the report? If this is false the ReportDateTime is the CreationDate of the file without a warning.

```
protected override bool ReportDateTimeExpected { get; }
```

Property Value

[bool](#)

Methods

IsValid(FileReaderContext)

Check if the file is a valid csv file containing the mandatory columns 'File', 'Family', and 'Color'

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

[context](#) [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid Polyspace csv file

Read(FileReaderContext)

Read each csv line, that is not a comment, and remove empty columns. Aggregate unique data lines by summing up and counting each data line as a value of 1. For Run time checks and MISRA, measure count contains both with and without comments. Violations is sum of all MISRA checks (to be consistent with xml adapter).

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Class PolyspaceXmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class PolyspaceXmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← PolyspaceXmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformString.ToDateTime\(string\)](#) ,
[AdapterBase.TransformString.ToDateTime\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if the file is a valid xml file containing an Artifact Name, the Execution Date, the Polyspace Version and the Result file path

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid Polyspace xml file

Read(FileReaderContext)

Reads the entries of all table elements

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Class QacHtmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class QacHtmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← QacHtmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDate\(string\)](#) ,
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if the file name contains 'RCR'

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid QAC XML report file

Read(FileReaderContext)

FINDINGS: Read each table from <body><div id='content'><div class='dpp'><div class='rgtable'><table ...> for each Rule Group as Findings for each file/Artifact. Read artifactPath and artifactName from first <td> ('Files') for each row from table. Read findingsFailed from last <td> ('Total Violations') for each row from table. Read measurementName from <body><div id='content'><div class='dpp'><div class='subsec'><h5 ...>. Concat Findings with measurement name to get measurementName.

GUIDELINES: Read each table from <body><div id='content'><div class='worstrules'><div class='rgtable'><table ...> for each Rule Group as Guidelines for each file/Artifact. Read artifactPath and artifactName from first <td> ('Files') for each row from table. Count the columns with rules and use as guidelinesTotal. Read each row from column and count all <td> with violation to calculate guidelinesFailed. Calculate guidelinesPassed as guidelinesTotal - guidelinesFailed

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Returns

[AdapterReadResult](#)

Examples

```
<body> ... <div id="content"> <div class="dpp"> ... <div class="subsec"><h5>M3CM</h5></div>
<div class="rgtable"> <table ....> <tr><th>Files</th><th>Rule 0</th><th>Rule 1</th><th>Rule
2</th><th>Rule 3</th>...<td><b>Total Violations</b></td></tr> <tr><td align = "left" ><a
href="xxx" title="xxx"> xxx </a></td><td> xxx </td><td>xxx</td>...<td><b>0</b></td></tr> ...
</table> </div> </div> <div class='worstrules'> <div class="subsec"><h5>xxx</h5></div> <div
class='rgtable'> <table ...> <tr><th>Files</th><th>Rule 1 </th><th>Rule 2 </th> ... <th>RuleZ</th>
</tr> <tr><td ... ><a href="xxx" title="xxx"> xxx </a></td><td> xxx </td><td> xxx </td> ... <td> xxx
</td></tr> ... </table> </div> </div> </body>
```

Class QacXmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class QacXmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← QacXmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDate\(string\)](#) ,
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

GetHumanReadableFilePath(FileReaderContext)

Get the human-readable file path. Can be overridden with a user defined function. Can be used together with GetHumanReadableFilePaths, but it is recommended to only use one of the methods.

```
protected override string GetHumanReadableFilePath(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

Returns

[string](#)

Human readable file

IsValid(FileReaderContext)

Check if the file name is equal to 'results_data.xml' and check if the root element is equal to 'AnalysisData'

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid QAC XML report file

Read(FileReaderContext)

Read each 'File' element inside <dataroot type="per-file"> and fetch 'path' attribute as ArtifactPath if there is 'File' element with same 'path' attribute inside <dataroot type="project"> <tree type="files"> and fetch 'basename' attribute as ArtifactName. For each Artifact these measures should be read : Guidelines, Findings Measures should be read for different measurements per artifact. The measurement name is taken from the attribute 'name' of each <tree type="rules"> <RuleGroup name = "xxx"> MeasureName: Guidelines From 'Rule' elements inside <dataroot type="project"> <tree type="rules"> <RuleGroup name = "xxx"> read number of 'Rule' elements that has 'Message' elements as child element as value of 'Total' variable From 'Rule' elements inside <dataroot type="per-file"> <File path=".../example.h"> <tree type="rules"> <RuleGroup name = "xxx"> read number of 'Rule' elements that has 'Message' elements as child element and active="0" as value of 'Suppressed' variable read number of all 'Rule' elements that has 'Message' elements as child element minus value of 'Suppressed' as value of 'Failed' variable calculate value of 'Total' minus value of 'Failed' as value of 'Passed' variable All above measures read for measurement = 'name' attribute of 'RuleGroup' Note: Guidelines.Passed contains Guidelines.Suppressed MeasureName: Findings From 'Rule' elements inside <dataroot type="per-file"> <File path=".../example.h"> <tree type="rules"> <RuleGroup name = "xxx"> read 'active' attribute as value of 'Failed' variable read 'total' attribute minus value of 'Failed' as value of 'Suppressed' variable All above measures read for measurement = 'name' attribute of 'RuleGroup'

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

=> Guidelines.Total = 3 for MeasurementName 'xxx' => ArtifactPath = '.../example.h' , ArtifactName = 'example.h' => Findings.Failed = 4 , Findings.Suppressed = 3 for MeasurementName 'xxx' => Guidelines.Failed = 1 , Guidelines.Suppressed = 2, Guidelines.Passed = 2 for MeasurementName 'xxx'

Class RtrtHtmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class RtrtHtmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← RtrtHtmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDate\(string\)](#) ,
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if the file name is equal to 'index.html' and the 'title' element contains 'Test RealTime Report'

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid RTRT report file

Examples

Read(FileReaderContext)

Read ReportDateTime from 'l' element inside 'body' element that contains 'Generated on'. Read Test and Coverage results from separate files and then sum up values for same artifact, measure and variable.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

```
<body> <hr><h3><p ALIGN="RIGHT"><i>Generated on 21.08.2021 02:59:08</i></p></h3> </body>
```

Class SICheckHtmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class SICheckHtmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← SICheckHtmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDateTime\(string\)](#) ,
[AdapterBase.TransformStringToDateTime\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if file contains 'Run Summary'

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid SLCheck Model Advisor

Examples

```
<b>Run Summary</b></font><br />
```

Read(FileReaderContext)

Fetch Guideline infos by reading rows of 'table' element after 'Run Summary' each 'th' element in first row is variable and corresponding 'tr' in second row is value measure name is 'GuidelineCount'

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

```
<b>Run Summary</b></font><br /> <table class="AdvTableNoBorder" width="60%" border="0">
<tr> <th align="left" valign="top"><b>Pass</b></th> <th align="left" valign="top"><b>Fail</b>
</th> <th align="left" valign="top"><b>Warning</b></th> <th align="left" valign="top"><b>Not
Run</b></th> <th align="left" valign="top"><b>Total</b></th> </tr> <tr> <td align="left"
valign="top">&#160;&#160; 38</td> <td align="left"
valign="top">&#160;&#160; 8</td> <td align="left"
valign="top">&#160;&#160; 7</td> <td align="left"
valign="top">&#160;&#160; 363</td> <td align="left"
valign="top">416</td> </tr> </table >
```

Class S1DesignVerifierHtmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class S1DesignVerifierHtmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← [S1DesignVerifierHtmlFileReader](#)

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDate\(string\)](#) ,
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if title element is "Simulink Design Verifier Report"

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid Simulink Design Verifier

Examples

Read(FileReaderContext)

Fetch the rows in first table after the 'p' element with title= "Objectives Status" and read first td as variable name(fro measure name = 'Objectives') and second td as value.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

```
<p title="Objectives Status"><b>Objectives Status</b><a name="d0e51"></a></p> <div class="table"> <div class="table-contents"> <table summary = ""> <thead> <tr><th align = "left" > Number of Objectives:</th><th align = "left" > 5 </ th ></ tr > </thead > <tbody> <tr><td align="left">Objectives Proven Valid: </td><td align = "left" > 4 </ td ></ tr > <tr><td align="left">Objectives Falsified with Test Cases: </td><td align = "left" > 1 </td></tr> </tbody> </table > </div></div>
```

Class SIRequirementHtmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class SIRequirementHtmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← SIRequirementHtmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDate\(string\)](#) ,
[AdapterBase.TransformStringToDate\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if file contains span element with 'SLReqReportTitle' as value of class attribute

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid SLRequirement Model Advisor

Examples

```
<span class="SLReqReportTitle">
```

Read(FileReaderContext)

Fetch data from table with "Implementation Status" as header and 'SLReqReqSetImplementationTitle' as class of that title. Read headers as variable name and second row as value for 'SLRequirement' as measure name. Read file name as artifact path and name.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

Implementation Status

Total	Implemented	Justified	None
11	3	0	8

Class TargetLinkHtmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class TargetLinkHtmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← TargetLinkHtmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDateTime\(string\)](#) ,
[AdapterBase.TransformStringToDateTime\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if the file name ends with 'ccdoc_Main.html'

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid Target Link report

Read(FileReaderContext)

Fetch Info from first table: read artifact path from third td where first td is 'TL Code Coverage Report for Application', read report date time from third td where first td is 'Date and time of report generation', read measure name from third td where first td is 'Code Coverage level'. Fetch Info from second table: read variables from columns of first row (column name='Code Coverage'=> variable='Ratio', column name='Total Branches'=> variable='Total', column name='Reached Branches'=> variable='Reached') read values for that variable from second row.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

TL Code Coverage Report for Application	:	Artifact
Date and time of report generation	:	2021 - 02 - 13 04:37:08
Code Coverage level	:	Decision Coverage (C1)

	Code Coverage	Total Branches	Reached Branches	Unreached Branches
Artifact	96.60 %	676	653	23

Class TessyXmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class TessyXmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← TessyXmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDateTime\(string\)](#) ,
[AdapterBase.TransformStringToDateTime\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if report contains 'tessy_version'

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid Tessy report file

Read(FileReaderContext)

From adapter options read that 'tessyobject' elements with which level should be read (if its not defined ad adapter option default level is '0'). Iterate over 'tessyobject' elements and read 'name' attribute as 'ArtifactPath' and fetch measures from elements inside.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

....

Class TestwellCtcHtmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class TestwellCtcHtmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← [TestwellCtcHtmlFileReader](#)

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformString.ToDateTime\(string\)](#) ,
[AdapterBase.TransformString.ToDateTime\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if the file name is equal to 'index.html' Check if 'content' attribute of 'meta' element inside 'head' element, contains 'CTC++'

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid CTC report file

Examples

```
<head> <meta name="generator" content="Testwell CTC++ add-on utility ctc2html v5.1"> </head>
```

Read(FileReaderContext)

Process all rows of second table inside body element ignore trs with just one td and with class="dirb" or class="ruler" or if value inside 'a' element in last td is equal to "DIRECTORY OVERALL" or "OVERALL" from remaining rows fetch ArtifactPath from 'a' element inside last td Names of base measures are fetch from th inside of 'thead' element, start from 3rd th and every 3rd th For each measure there is 3 variables : 'Covered', 'Uncovered' and 'Total' Values for base measures are fetch from each row start from 3rd td and every 4th td value is like 41/42 then Covered = 41 , Total = 42, Uncovered = 1

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

```
<table> <thead> <tr><th>TER %</th><th>-</th><th colspan = "2" > decision </th> =>
MeasureNames = decision < th > TER %</ th >< th > -</th > < th colspan="2">statement</th> =>
MeasureNames = statement <th>File</th></tr> </thead> <tbody> <tr><td class="dirb"
colspan="9"></td>...</tr> => ignored <tr><td class="below">98 %</td><td class="below">-</td>
<td class="below">(41/42)</td> => decision.Reached = 41 , decision.Total = 42 <td width="115"><img
><img ></td><td class="below">99 %</td><td class="below">-</td><td class="below">(86/87)
</td> => statement.Reached = 86 , statement.Total = 87 <td width="115"><img ><img ></td><td>
<a >Autopilot_Mode_Logic_sil_sil_ec.c</a></td></tr> => ArtifactPath =
'Autopilot_Mode_Logic_sil_sil_ec.c' <tr>.... <td class="dirb"><a > DIRECTORY OVERALL</a></td></tr>
=> ignored <tr><td class="ruler" colspan="9">&nbsp;</td></tr> => ignored <tr>....<td><a href =
"indexO.html" class="underlineb">OVERALL</a></td></tr> => ignored </tbody> </table>
```

Class TestwellCtcXmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class TestwellCtcXmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← [TestwellCtcXmlFileReader](#)

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDateTIme\(string\)](#) ,
[AdapterBase.TransformStringToDateTIme\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

HumanReadableFileExtensions

Human Readable File Extensions of the Adapter.

```
protected override List<string> HumanReadableFileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

HumanReadableStartFileNames

Start of Human Readable File Names of the Adapter.

```
protected override List<string> HumanReadableStartFileNames { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

GetHumanReadableFilePath(FileReaderContext)

Get the human-readable file path. Can be overridden with a user defined function. Can be used together with GetHumanReadableFilePaths, but it is recommended to only use one of the methods.

```
protected override string GetHumanReadableFilePath(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

FileReaderContext

Returns

[string](#)

Human readable file

IsValid(FileReaderContext)

Check if the root element is <ctc_xml_report>

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#) ↗

true if file is a valid CTC report file

Read(FileReaderContext)

Read each file element and fetch name attribute as ArtifactPath from file_summary element inside file element these elements are read as measures: lines element => MeasureName = "Source lines" , VariableName="count" measurement_points element => MeasureName = "Measurement points" , VariableName="count" xx_hit and xx_all => MeasureName = "xxx" , VariableName="Reached" (for hit value), VariableName="Total" (for all value) This measure (xx) is added if both xx_hit and xx_all elements are exist

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

AdapterReadResult

All Data is stored in an AdapterData Model object

Examples

```
<ctc_xml_report> <file_summary> 77 200 <measurement_points>16</measurement_points> 68 343  
505 <statement_ter>92</statement_ter> <statement_hits>5310</statement_hits>  
<statement_all>5752</statement_all> <statement_na_functions>0</statement_na_functions>  
</file_summary> ... </ctc_xml_report>
```

Class TptHtmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class TptHtmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← TptHtmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformString.ToDateTime\(string\)](#) ,
[AdapterBase.TransformString.ToDateTime\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if the 'title' element is equal to 'TPT Report: Overview'

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid TPT report file

Examples

Read(FileReaderContext)

Fetch Info from table with class='table-wide': read artifact path from second td where td is 'System Under Test' or 'SUT', read report date time from second td where td is 'Date' or 'SUT'. Fetch measurement from table with caption 'Platform Information': res second td of tr as measurement where first td is 'Variable 'TestRun'''

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

Date	07-Dec-2020
Variable 'TestRun'	SIL

Platform Information

ReadAssessletFindings(FileReaderContext, HtmlDocument, DateTime?, string, string, bool, List<AdapterData>, List<Requirement>, List<TptAssesslet>, AdapterFinding[], AdapterFinding[])

```
public static List<AdapterFinding> ReadAssessletFindings(FileReaderContext context,
    HtmlDocument document, DateTime? dateTime, string artifactPath, string measurementName, bool
    hasRequirementLinkInfo, List<AdapterData> data, List<TptHtmlFileReader.Requirement>
    requirements, List<TptHtmlFileReader.TptAssesslet> assesslets = null, AdapterFinding[]
    requirementFindings = null, AdapterFinding[] testcaseFindings = null)
```

Parameters

context [FileReaderContext](#)

document [HtmlDocument](#)

dateTime [DateTime](#)?

artifactPath [string](#)

measurementName [string](#)

hasRequirementLinkInfo [bool](#)

data [List](#)<[AdapterData](#)>

requirements [List](#)<[TptHtmlFileReader.Requirement](#)>

assesslets [List](#)<[TptHtmlFileReader.TptAssesslet](#)>

`requirementFindings AdapterFinding[]`

`testcaseFindings AdapterFinding[]`

Returns

`List<AdapterFinding>`

`ReadAssesslets(FileReaderContext, List<AdapterData>,
HtmlDocument, DateTime?, string, string, List<Requirement>)`

```
public static List<TptHtmlFileReader.TptAssesslet> ReadAssesslets(FileReaderContext context,  
List<AdapterData> data, HtmlDocument document, DateTime? dateTime, string artifactPath,  
string measurementName, List<TptHtmlFileReader.Requirement> requirements)
```

Parameters

`context FileReaderContext`

`data List<AdapterData>`

`document HtmlDocument`

`dateTime DateTime?`

`artifactPath string`

`measurementName string`

`requirements List<TptHtmlFileReader.Requirement>`

Returns

`List<TptHtmlFileReader.TptAssesslet>`

`ReadRequirementCoverage(HtmlDocument, DateTime?, string,
string, bool, List<Requirement>)`

Read table with 'Requirement Coverage Summary' caption and define these measure 'Requirements Status' with variables: Passed, Failed, ExecutionError, DontKnow, NotCovered Requirement Count.Testable

=> number of all variables of 'Requirements Status' measure. From 'requirement.html' file: read 'Requirements Results' table and 'Requirements Assesslet Results' tables: Requirement Count.Total => if it is reports from PikeTec then it is : number of requirements fetch from 'Requirements Results' table plus requirements not covered by testCases for other reports : number of requirement ids fetch from 'Requirements Results' table

```
public static List<AdapterData> ReadRequirementCoverage(HtmlDocument document, DateTime?  
dateTime, string artifactPath, string measurement, bool hasRequirementLinkInfo,  
List<TptHtmlFileReader.Requirement> requirements)
```

Parameters

document HtmlDocument

content of report

dateTime [DateTime](#)?

date of report

artifactPath [string](#)

name of artifact

measurement [string](#)

name of measurement

hasRequirementLinkInfo [bool](#)

requirements [List](#)<[TptHtmlFileReader.Requirement](#)>

from requirement.html file

Returns

[List](#)<[AdapterData](#)>

ReadRequirementFindings(FileReaderContext, List<AdapterData>, DateTime?, string, string, List<Requirement>)

Read all requirement findings found in the "Requirements Results" table in the requirement.html file.

```
public static List<AdapterFinding> ReadRequirementFindings(FileReaderContext context,  
List<AdapterData> data, DateTime? dateTime, string artifactPath, string measurementName,  
List<TptHtmlFileReader.Requirement> requirements)
```

Parameters

context [FileReaderContext](#)

context for reading the report

data [List](#)<[AdapterData](#)>

data read from report

dateTime [DateTime](#)?

date of the report

artifactPath [string](#)

artifact name

measurementName [string](#)

measurement name

requirements [List](#)<[TptHtmlFileReader.Requirement](#)>

Returns

[List](#)<[AdapterFinding](#)>

List of the findings found. Returns empty list if none are found.

ReadRequirementReport(string, out bool)

```
public static List<TptHtmlFileReader.Requirement> ReadRequirementReport(string filePath, out  
bool hasRequirementLinkInfo)
```

Parameters

filePath [string](#)

`hasRequirementLinkInfo bool`

Returns

`List<TptHtmlFileReader.Requirement>`

Class TptXmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class TptXmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← TptXmlFileReader

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDateTIme\(string\)](#) ,
[AdapterBase.TransformStringToDateTIme\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

GetHumanReadableFilePaths(FileReaderContext)

This is a private function that combine the result of GetHumanReadableFilePath and GetHumanReadableFilePaths. If the result is empty then return human-readable files based on settings.

```
protected override List<string> GetHumanReadableFilePaths(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Returns

[List](#)<[string](#)>

IsValid(FileReaderContext)

Check if the Header element has TptVersion attribute

```
protected override bool IsValid(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid TPT report file

Read(FileReaderContext)

Parses the TPT XML file to find the summary of test results in the execution summary section and structural and requirement coverage. The artifact name is read from the system under test property of

the header and when its empty from TptFileName attribute of Header.

```
protected override AdapterReadResult Read(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

```
<Header ...> <Property Name = .../> ... <Property Name = "System Under Test"  
Value="SystemUnderTest"/> </Header> ... <Summary AssessmentDuration="50.533"  
ExecutionDuration="1039.099"> <ExecutionSummary Errors = "3" Failed="1" Inconclusive="0"  
Succeeded="40" Tests="44"/> </Summary>
```

ReadWithHumanReadableFiles(FileReaderContext, AdapterReadResult, IEnumerable<string>)

Read findings from tpt xml, if an overview.html exists use it to read the assesslet finding instead

```
protected override AdapterReadResult ReadWithHumanReadableFiles(FileReaderContext context,  
AdapterReadResult result, IEnumerable<string> humanReadableFilePaths)
```

Parameters

context [FileReaderContext](#)

Context of file

result [AdapterReadResult](#)

State from Read

`humanReadableFilePaths` [IEnumerable](#)<[string](#)>

Human readable files (html)

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterReadResult

Class VTestStudioHtmlFileReader

Namespace: [MES.MQC.DataSourceLibrary.Adapters.Files](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public class VTestStudioHtmlFileReader : FileReader
```

Inheritance

[object](#) ← [AdapterBase](#) ← [FileReader](#) ← [VTestStudioHtmlFileReader](#)

Inherited Members

[FileReader.Priority](#) , [FileReader.HumanReadableFileCount](#) , [FileReader.HumanReadableFileExtensions](#) ,
[FileReader.HumanReadableStartFileNames](#) , [FileReader.ReportDateTimeExpected](#) ,
[FileReader.ReadWithHumanReadableFiles\(FileReaderContext, AdapterReadResult, IEnumerable<string>\)](#) ,
[FileReader.GetHumanReadableFilePath\(FileReaderContext\)](#) ,
[FileReader.GetHumanReadableFilePaths\(FileReaderContext\)](#) , [FileReader.Validate\(FileReaderContext\)](#) ,
[FileReader.Execute\(FileReaderContext, bool\)](#) , [AdapterBase.Options](#) ,
[AdapterBase.TransformStringToValue\(string\)](#) , [AdapterBase.TransformStringToDateTime\(string\)](#) ,
[AdapterBase.TransformStringToDateTime\(AdapterContext, string\)](#) ,
[AdapterBase.ThrowWarningForEmptyDataProperty\(AdapterDataBase, AdapterContextBase, string, string\)](#) ,

[AdapterBase.ReadFilePathsInZipFile\(string\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DataSource

The Data Source of the Adapter. If a report file contains data from multiple data sources, this property has to be to "Unknown" and the DataSource of each AdapterData object has to be defined.

```
public override string DataSource { get; }
```

Property Value

[string](#)

Description

Description of the Adapter that is visible in the Adapter Dialog as a popover. If the adapter is an API Connector, the description is also shown in the Add/Edit DataSource dialog, when the API Connector is selected. Absolute links get transformed into HTML Link Tags, line breaks (\n) get transformed into HTML line breaks (
), HTML tags are not allowed.

```
public override string Description { get; }
```

Property Value

[string](#)

FileExtensions

File Extensions of the Adapter. This property has to be defined and must have at least one file extension. The Adapter is only used for FilePaths with the defined file extensions. The IsValid method is not called unless the file extension matches.

```
public override List<string> FileExtensions { get; }
```

Property Value

[List](#)<[string](#)>

Name

Unique Name of the Adapter. Defaults to the ClassName (without Adapter / FileReader / ApiConnector suffix), can be overridden with a user defined Name.

```
public override string Name { get; }
```

Property Value

[string](#)

Methods

IsValid(FileReaderContext)

Check if the the report contains 'vteststudio'

```
protected override bool IsValid(FileReaderContext fileContext)
```

Parameters

`fileContext` [FileReaderContext](#)

Context of file

Returns

[bool](#)

true if file is a valid VTest Studio report file

Read(FileReaderContext)

Find div element with 'System Under Test' value and read ArtifactPath from the second column of the next table's row that first column is 'ECU Name'. Find a element with the 'TestOverview' as value of name attribute to fetch report date time and data.

```
protected override AdapterReadResult Read(FileReaderContext fileContext)
```

Parameters

`fileContext` [FileReaderContext](#)

Context of file

Returns

[AdapterReadResult](#)

All Data is stored in an AdapterData Model object

Examples

System Under Test

....

ECU Name:	ArtifactPath
-----------	--------------

Namespace MES.MQC.DataSourceLibrary.Models.Adapters

Classes

[AdapterData](#)

The AdapterData class contains the data for one data item to be imported. Each Adapter returns a List of with objects of this class on execution with the Read method.

[AdapterDataBase](#)

The AdapterDataBase class contains the data for one data item to be imported.

[AdapterDataExtBase](#)

The BasicAdapterData class contains the data for one data item to be imported.

[AdapterDataFromFilePath](#)

The AdapterDataFromFilePath class contains the data for a filePath read with basic adapter options.

[AdapterDownloadResult](#)

The Result of Download

[AdapterFinding](#)

The AdapterFinding class contains the data for one data item to be imported. Each Adapter returns a List of with objects of this class on execution with the Read method.

[AdapterFindingData](#)

Data Relation for Findings.

[AdapterFindingRelation](#)

Data Relation for Findings.

[AdapterMeasure](#)

The AdapterMeasure class contains the data for one measure item to be imported.

[AdapterMessage](#)

The Errors and Warnings contain multiple AdapterMessage with Title and Description.

[AdapterMessageBag](#)

The AdapterMessageBag contains multiple Error and Warning AdapterMessages.

[AdapterReadResult](#)

The Result of Read.

[AdapterResultBase](#)

The Result of Execute.

[AdapterValidationResult](#)

The Result of Validate.

[ApiConnectorConfiguration](#)

ApiConnectorConfiguration to be implemented as a nested subclass of a MQC API Connector.

Provides configuration of a API DataSource. Rendered with DynamicForm based on the Form-Attributes of the class and properties.

[ApiConnectorOptions](#)

[ApiConnectorResult](#)

The Result of ApiConnector.Execute.

[FileReaderOptions](#)

Default AdapterOptions for FileAdapters. Provides configuration to extract information from the file path (ArtifactPath, DataSourceName, MeasurementName, ReportDateTime).

[FileReaderOptions.DateTimeFromFilePath](#)

[FileReaderOptions.InputFromFilePath](#)

[FileReaderResult](#)

The Result of FileAdapter.Execute.

[ReaderResult](#)

@deprecated For backwards compatibility only

Class AdapterData

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The AdapterData class contains the data for one data item to be imported. Each Adapter returns a List of objects of this class on execution with the Read method.

```
public class AdapterData : AdapterDataExtBase
```

Inheritance

[object](#) ← [AdapterDataBase](#) ← [AdapterDataExtBase](#) ← AdapterData

Inherited Members

[AdapterDataExtBase.MeasurementName](#) , [AdapterDataExtBase.MeasureName](#) ,
[AdapterDataExtBase.VariableName](#) , [AdapterDataBase.DateTime](#) , [AdapterDataBase.DataSourceName](#) ,
[AdapterDataBase.ArtifactPath](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

ArtifactName

Name of the Artifact. This property is optional, the ArtifactName will be taken from the Project Structure if available or derived from the artifact path if not.

```
public string ArtifactName { get; set; }
```

Property Value

[string](#)

Value

The Value has to be of the double type.

```
public double? Value { get; set; }
```

Property Value

double?

Class AdapterDataBase

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The AdapterDataBase class contains the data for one data item to be imported.

```
public abstract class AdapterDataBase
```

Inheritance

[object](#) ← AdapterDataBase

Derived

[AdapterDataExtBase](#), [AdapterFinding](#)

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

ArtifactPath

Path of the Artifact.

```
public string ArtifactPath { get; set; }
```

Property Value

[string](#)

DataSourceName

Name of the Data Source. This property is optional and can remain null, in this case the DataSource property of the Adapter is used.

```
public string DataSourceName { get; set; }
```

Property Value

[string](#) ↗

DateTime

Date of the report.

```
public DateTime? DateTime { get; set; }
```

Property Value

[DateTime](#) ↗?

Class AdapterDataExtBase

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The BasicAdapterData class contains the data for one data item to be imported.

```
public abstract class AdapterDataExtBase : AdapterDataBase
```

Inheritance

[object](#) ← [AdapterDataBase](#) ← AdapterDataExtBase

Derived

[AdapterData](#), [AdapterMeasure](#)

Inherited Members

[AdapterDataBase.DateTime](#) , [AdapterDataBase.DataSourceName](#) , [AdapterDataBase.ArtifactPath](#) ,
[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

MeasureName

Name of the Measure.

```
public string MeasureName { get; set; }
```

Property Value

[string](#)

MeasurementName

Name of the Measurement.

```
public string MeasurementName { get; set; }
```

Property Value

[string](#) ↗

VariableName

Name of the Variable.

```
public string VariableName { get; set; }
```

Property Value

[string](#) ↗

Class AdapterDataFromFilePath

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The AdapterDataFromFilePath class contains the data for a filePath read with basic adapter options.

```
public class AdapterDataFromFilePath
```

Inheritance

[object](#) ← AdapterDataFromFilePath

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

ArtifactPath

Path of the Artifact.

```
public string ArtifactPath { get; set; }
```

Property Value

[string](#)

ArtifactPathFallback

Fallback Path of the Artifact.

```
public string ArtifactPathFallback { get; set; }
```

Property Value

[string](#)

DataSourceName

Name of the Data Source.

```
public string DataSourceName { get; set; }
```

Property Value

[string](#)

DataSourceNameFallback

Fallback Name of the Data Source.

```
public string DataSourceNameFallback { get; set; }
```

Property Value

[string](#)

DateTime

Date of the report.

```
public DateTime? DateTime { get; set; }
```

Property Value

[DateTime](#)?

DateTimeFallback

Fallback Date of the report.

```
public DateTime? DateTimeFallback { get; set; }
```

Property Value

[DateTime](#)?

MeasurementName

Name of the Measurement.

```
public string MeasurementName { get; set; }
```

Property Value

[string](#)

MeasurementNameFallback

Fallback Name of the Measurement.

```
public string MeasurementNameFallback { get; set; }
```

Property Value

[string](#)

Class AdapterDownloadResult

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The Result of Download

```
public class AdapterDownloadResult
```

Inheritance

[object](#) ← AdapterDownloadResult

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

DateTime

DateTime of the current api request, returned from the api server.

```
public DateTime DateTime { get; set; }
```

Property Value

[DateTime](#)

UseFileCreationTimeForAll

If true, the DateTime from the API will be used. as the report date, while importing all files, downloaded by this adapter, with file adapters. The files need to have the DateTime from the API set as creation time:

```
File.SetCreationTime
```

.

```
public bool UseFileCreationTimeForAll { get; set; }
```

Property Value

[bool](#)

UseFileCreationTimePaths

If UseFileCreationTimeForAll is not true, this list can provide directoryPaths or filePaths. For every file in a directory path and for a file path, the DateTime from the API will be used as the report date, while importing all files, downloaded by this adapter, with file adapters. The files need to have the DateTime from the API set as creation time:

```
File.SetCreationTime
```

```
public List<string> UseFileCreationTimePaths { get; set; }
```

Property Value

[List](#)<[string](#)>

Class AdapterFinding

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The AdapterFinding class contains the data for one data item to be imported. Each Adapter returns a List of objects of this class on execution with the Read method.

```
public class AdapterFinding : AdapterDataBase
```

Inheritance

[object](#) ← [AdapterDataBase](#) ← AdapterFinding

Inherited Members

[AdapterDataBase.DateTime](#) , [AdapterDataBase.DataSourceName](#) , [AdapterDataBase.ArtifactPath](#) ,
[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

ArtifactInnerPath

Artifact Inner Path of the Finding.

```
public string[] ArtifactInnerPath { get; set; }
```

Property Value

[string](#)[]

DataDateTime

Date of the report.

```
public DateTime? DataDateTime { get; set; }
```

Property Value

[DateTime](#)?

Description

Description of the Finding (contains information what the finding is about).

```
public string Description { get; set; }
```

Property Value

[string](#)

HumanReadableAnchor

Anchor of the related human readable html file.

```
public string HumanReadableAnchor { get; set; }
```

Property Value

[string](#)

HumanReadableFilePath

Human readable html file if different from the data.

```
public string HumanReadableFilePath { get; set; }
```

Property Value

[string](#)

IssueMessage

Issue Message of the Finding (contains the problem).

```
public string IssueMessage { get; set; }
```

Property Value

[string](#) ↗

Measure

Measure of the Finding (e.g. Result, Coverage of Testcases).

```
public string Measure { get; set; }
```

Property Value

[string](#) ↗

MeasurementName

Name of Measurement, used to replace \${MeasurementName} in FindingStructure and FindingStructureForArtifact (after verified with adapter options).

```
public string MeasurementName { get; set; }
```

Property Value

[string](#) ↗

State

State of the Finding (e.g. Warning, Failed).

```
public string State { get; set; }
```

Property Value

[string](#)

SubjectName

Subject Name of the Finding.

```
public string SubjectName { get; set; }
```

Property Value

[string](#)

SubjectPath

Subject Path of the Finding.

```
public string[] SubjectPath { get; set; }
```

Property Value

[string](#)[]

SubjectPathForArtifact

Subject Path of the Finding specific only for one artifact.

```
public string[] SubjectPathForArtifact { get; set; }
```

Property Value

[string](#)[]

SubjectType

Subject Type of the Finding (e.g. Requirement, Testcase, Check).

```
public string SubjectType { get; set; }
```

Property Value

[string](#)

Methods

AddData(AdapterData)

Add an adapterData entry to create a relation.

```
public void AddData(AdapterData data)
```

Parameters

data [AdapterData](#)

AdapterData

AddData(IEnumerable<AdapterData>)

Add multiple adapterData entries to create relations.

```
public void AddData(IEnumerable<AdapterData> data)
```

Parameters

data [IEnumerable](#)<[AdapterData](#)>

Array of AdapterData

AddRelation(AdapterFinding, string)

Add an adapterFinding entry to create a relation.

```
public void AddRelation(AdapterFinding finding, string result = null)
```

Parameters

finding [AdapterFinding](#)

AdapterFinding

result [string](#)

Result of the Relation

AddRelation(IEnumerable<AdapterFinding>, string)

Add multiple adapterFinding entries to create relations.

```
public void AddRelation(IEnumerable<AdapterFinding> findings, string result = null)
```

Parameters

findings [IEnumerable](#)<AdapterFinding>

Array of AdapterFinding

result [string](#)

Result of the Relation

CreateNew(string, string, DateTime?, string, string, string[], string[], string[], string, string, string, string, string, string, string, string, DateTime?)

Create a new finding based on the current finding. Arguments are optional and can provide different values for some properties.

```
public AdapterFinding CreateNew(string type = null, string measure = null, DateTime? dateT
```

```
ime = null, string dataSourceName = null, string artifactPath = null, string[] artifactInn
```

```
erPath = null, string[] subjectPath = null, string[] subjectPathForArtifact = null, string state = null, string subjectName = null, string description = null, string
```

```
issueMessage = null, string humanReadableFilePath = null, string humanReadableAnchor = null,  
string measurementName = null, DateTime? dataDateTime = null)
```

Parameters

type [string](#)

Type for the new Finding

measure [string](#)

Measure for the new Finding

dateTime [DateTime](#)?

DateTime for the new Finding

dataSourceName [string](#)

DataSourceName for the new Finding

artifactPath [string](#)

ArtifactPath for the new Finding

artifactInnerPath [string](#)[]

ArtifactInnerPath for the new Finding

subjectPath [string](#)[]

SubjectPath for the new Finding

subjectPathForArtifact [string](#)[]

SubjectPathForArtifact for the new Finding

state [string](#)

State for the new Finding

subjectName [string](#)

SubjectName for the new Finding

description [string](#)

Description for the new Finding

issueMessage [string](#)?

IssueMessage for the new Finding

humanReadableFilePath [string](#)?

HumanReadableFilePath for the new Finding

humanReadableAnchor [string](#)?

HumanReadableAnchor for the new Finding

measurementName [string](#)?

MeasurementName for the new Finding

dataDateTime [DateTime](#)?

DataDateTime for the new Finding

Returns

[AdapterFinding](#)

AdapterFinding (a new object)

Class AdapterFindingData

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

Data Relation for Findings.

```
public class AdapterFindingData
```

Inheritance

[object](#) ← AdapterFindingData

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

MeasureName

Name of the Measure.

```
public string MeasureName { get; set; }
```

Property Value

[string](#)

MeasurementName

Name of the Measurement.

```
public string MeasurementName { get; set; }
```

Property Value

[string](#) ↗

VariableName

Name of the Variable.

```
public string VariableName { get; set; }
```

Property Value

[string](#) ↗

Class AdapterFindingRelation

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

Data Relation for Findings.

```
public class AdapterFindingRelation
```

Inheritance

[object](#) ← AdapterFindingRelation

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

Finding

The Finding.

```
public AdapterFinding Finding { get; set; }
```

Property Value

[AdapterFinding](#)

RelationResult

Result of the relation (optional).

```
public string RelationResult { get; set; }
```

Property Value

[string](#) ↗

Class AdapterMeasure

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The AdapterMeasure class contains the data for one measure item to be imported.

```
public class AdapterMeasure : AdapterDataExtBase
```

Inheritance

[object](#) ← [AdapterDataBase](#) ← [AdapterDataExtBase](#) ← AdapterMeasure

Inherited Members

[AdapterDataExtBase.MeasurementName](#) , [AdapterDataExtBase.MeasureName](#) ,
[AdapterDataExtBase.VariableName](#) , [AdapterDataBase.DateTime](#) , [AdapterDataBase.DataSourceName](#) ,
[AdapterDataBase.ArtifactPath](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Class AdapterMessage

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The Errors and Warnings contain multiple AdapterMessage with Title and Description.

```
public class AdapterMessage
```

Inheritance

[object](#) ← AdapterMessage

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

Description

```
public string Description { get; set; }
```

Property Value

[string](#)

Files

```
public string Files { get; set; }
```

Property Value

[string](#)

Title

```
public string Title { get; set; }
```

Property Value

[string](#) ↗

Class AdapterMessageBag

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The AdapterMessageBag contains multiple Error and Warning AdapterMessages.

```
public class AdapterMessageBag
```

Inheritance

[object](#) ← AdapterMessageBag

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

AdapterName

```
public string AdapterName { get; set; }
```

Property Value

[string](#)

Errors

```
public HashSet<AdapterMessage> Errors { get; set; }
```

Property Value

[HashSet](#)<[AdapterMessage](#)>

Warnings

```
public HashSet<AdapterMessage> Warnings { get; set; }
```

Property Value

[HashSet](#)<[AdapterMessage](#)>

Methods

ThrowError(string, string)

Throws an Error message to be either ignored, displayed at a notification (on data source import or refresh) or shown in a validation dialog (if a report file is imported as a test). The import of the current filePath is aborted with an internal exception.

```
public void ThrowError(string title, string description)
```

Parameters

title [string](#)

Title of the error notification

description [string](#)

Description of the error notification

ThrowWarning(string, string)

Throws an Warning message to be either ignored, displayed at a notification (on data source import or refresh) or shown in a validation dialog (if a report file is imported as a test). The import of the current filePath continues unimpeded by the warning.

```
public void ThrowWarning(string title, string description)
```

Parameters

title [string ↗](#)

Title of the warning notification

description [string ↗](#)

Description of the warning notification

Class AdapterReadResult

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The Result of Read.

```
public class AdapterReadResult
```

Inheritance

[object](#) ← AdapterReadResult

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

Data

The imported data (aggregated numbers).

```
public List<AdapterData> Data { get; }
```

Property Value

[List](#) <[AdapterData](#)>

DateTime

The datetime of the current import (for api adapters).

```
public DateTime DateTime { get; set; }
```

Property Value

Findings

The imported findings (data details).

```
public List<AdapterFinding> Findings { get; }
```

Property Value

[List](#) <[AdapterFinding](#)>

HumanReadableUrl

The Url to a human-readable website for the data (for api adapters).

```
public string HumanReadableUrl { get; set; }
```

Property Value

[string](#)

Measures

The imported measures (which data should be defaulted if not imported).

```
public List<AdapterMeasure> Measures { get; }
```

Property Value

[List](#) <[AdapterMeasure](#)>

Class AdapterResultBase

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The Result of Execute.

```
public class AdapterResultBase
```

Inheritance

[object](#) ← AdapterResultBase

Derived

[ApiConnectorResult](#), [FileReaderResult](#)

Inherited Members

[object.ToString\(\)](#), [object.Equals\(object\)](#), [object.Equals\(object, object\)](#),
[object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#)

Constructors

AdapterResultBase()

```
public AdapterResultBase()
```

AdapterResultBase(AdapterMessageBag)

```
public AdapterResultBase(AdapterMessageBag messageBag)
```

Parameters

messageBag [AdapterMessageBag](#)

Properties

Data

```
public List<ImportedData> Data { get; set; }
```

Property Value

[List ↗ <ImportedData>](#)

Findings

```
public List<ImportedFinding> Findings { get; set; }
```

Property Value

[List ↗ <ImportedFinding>](#)

Measures

```
public List<ImportedMeasure> Measures { get; set; }
```

Property Value

[List ↗ <ImportedMeasure>](#)

MessageBag

```
public AdapterMessageBag MessageBag { get; set; }
```

Property Value

[AdapterMessageBag](#)

Class AdapterValidationResult

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The Result of Validate.

```
public class AdapterValidationResult
```

Inheritance

[object](#) ← AdapterValidationResult

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

IsValid

```
public bool IsValid { get; set; }
```

Property Value

[bool](#)

MessageBag

```
public AdapterMessageBag MessageBag { get; set; }
```

Property Value

[AdapterMessageBag](#)

Class ApiConnectorConfiguration

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

ApiConnectorConfiguration to be implemented as a nested subclass of a MQC API Connector. Provides configuration of a API DataSource. Rendered with DynamicForm based on the Form-Attributes of the class and properties.

```
public abstract class ApiConnectorConfiguration : IForm
```

Inheritance

[object](#) ← ApiConnectorConfiguration

Implements

[IForm](#)

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Constructors

ApiConnectorConfiguration()

```
protected ApiConnectorConfiguration()
```

Properties

FormSchema

The FormSchema, generated based on the Properties and Form-Attributes.

```
[YamlIgnore]  
public FormSchema FormSchema { get; set; }
```

Property Value

[FormSchema](#)

Type

```
[YamlIgnore]
[JsonProperty(Required = Required.Always, Order = -2147483648, PropertyName = "$type")]
public string Type { get; set; }
```

Property Value

[string](#)

Methods

GetHash()

```
public string GetHash()
```

Returns

[string](#)

Class ApiConnectorOptions

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
public abstract class ApiConnectorOptions : AdapterOptions
```

Inheritance

[object](#) ← AdapterOptions ← ApiConnectorOptions

Inherited Members

AdapterOptions.Type , [object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Class ApiConnectorResult

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The Result of ApiConnector.Execute.

```
public class ApiConnectorResult : AdapterResultBase
```

Inheritance

[object](#) ← [AdapterResultBase](#) ← ApiConnectorResult

Inherited Members

[AdapterResultBase.MessageBag](#) , [AdapterResultBase.Data](#) , [AdapterResultBase.Measures](#) ,
[AdapterResultBase.Findings](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Constructors

ApiConnectorResult()

```
public ApiConnectorResult()
```

ApiConnectorResult(ApiConnectorContext)

```
public ApiConnectorResult(ApiConnectorContext context)
```

Parameters

context [ApiConnectorContext](#)

Properties

DateTime

```
public DateTime DateTime { get; set; }
```

Property Value

[DateTime](#)

DirectoryPaths

```
public HashSet<string> DirectoryPaths { get; set; }
```

Property Value

[HashSet](#)<[string](#)>

UseFileCreationTimeFilePaths

```
public string[] UseFileCreationTimeFilePaths { get; set; }
```

Property Value

[string](#)[]

Class FileReaderOptions

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

Default AdapterOptions for FileAdapters. Provides configuration to extract information from the file path (ArtifactPath, DataSourceName, MeasurementName, ReportDateTime).

```
public abstract class FileReaderOptions : AdapterOptions
```

Inheritance

[object](#) ← AdapterOptions ← FileReaderOptions

Inherited Members

AdapterOptions.Type , [object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

ArtifactPathFromFilePath

Extract the ArtifactPath from the given file path.

```
[JsonProperty(Required = Required.DisallowNull, DefaultValueHandling  
= DefaultValueHandling.Ignore)]  
public FileReaderOptions.InputFromFilePath[] ArtifactPathFromFilePath { get; set; }
```

Property Value

[InputFromFilePath\[\]](#)

DataSourceNameFromFilePath

Extract the DataSourceName from the given file path.

```
[JsonProperty(Required = Required.DisallowNull, DefaultValueHandling  
= DefaultValueHandling.Ignore)]  
public FileReaderOptions.InputFromFilePath[] DataSourceNameFromFilePath { get; set; }
```

Property Value

[InputFromFilePath\[\]](#)

HumanReadableFilePaths

Find human readable file paths based on the imported data, usable in string pattern: {DataSourceName}, {ArtifactPath}, {MeasurementName}

```
[JsonProperty(Required = Required.DisallowNull, DefaultValueHandling  
= DefaultValueHandling.Ignore)]  
public string[] HumanReadableFilePaths { get; set; }
```

Property Value

[string\[\]](#)

ImportLatestOnlyFileExpressions

Expression for FilePaths, that if there are multiple matches, only the latest (creation date) is imported.

```
[JsonProperty(Required = Required.DisallowNull, DefaultValueHandling  
= DefaultValueHandling.Ignore)]  
public string[] ImportLatestOnlyFileExpressions { get; set; }
```

Property Value

[string\[\]](#)

MeasurementNameFromFilePath

Extract the MeasurementName from the given file path.

```
[JsonProperty(Required = Required.DisallowNull, DefaultValueHandling  
= DefaultValueHandling.Ignore)]  
public FileReaderOptions.InputFromFilePath[] MeasurementNameFromFilePath { get; set; }
```

Property Value

[InputFromFilePath\[\]](#)

ReportDateTimeFromFilePath

Extract the ReportDateTime from the given file path.

```
[JsonProperty(Required = Required.DisallowNull, DefaultValueHandling  
= DefaultValueHandling.Ignore)]  
public FileReaderOptions.DateTimeFromFilePath[] ReportDateTimeFromFilePath { get; set; }
```

Property Value

[DateTimeFromFilePath\[\]](#)

Class FileReaderOptions.DateTimeFromFilePath

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
[JsonObject(ItemRequired = Required.DisallowNull, ItemNullValueHandling  
= NullValueHandling.Ignore)]  
public class FileReaderOptions.DateTimeFromFilePath : FileReaderOptions.InputFromFilePath
```

Inheritance

[object](#) ← [FileReaderOptions.InputFromFilePath](#) ← [FileReaderOptions.DateTimeFromFilePath](#)

Inherited Members

[FileReaderOptions.InputFromFilePath.IsFallback](#) , [FileReaderOptions.InputFromFilePath.Regex](#) ,
[FileReaderOptions.InputFromFilePath.Result](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

Format

Format of the date string in the file path.

```
[JsonProperty(Required = Required.Always, DefaultValueHandling  
= DefaultValueHandling.Ignore)]  
public string Format { get; set; }
```

Property Value

[string](#)

Class FileReaderOptions.InputFromFilePath

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

```
[JsonObject(ItemRequired = Required.DisallowNull, ItemNullValueHandling  
= NullValueHandling.Ignore)]  
public class FileReaderOptions.InputFromFilePath
```

Inheritance

[object](#) ← FileReaderOptions.InputFromFilePath

Derived

[FileReaderOptions.DateTimeFromFilePath](#)

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

IsFallback

Should the value from file path only be used as a fallback (if it was empty/null).

```
[JsonProperty(Required = Required.DisallowNull, DefaultValueHandling  
= DefaultValueHandling.Ignore)]  
public bool IsFallback { get; set; }
```

Property Value

[bool](#)

Regex

Regex to extract the value from a file path.

```
[JsonProperty(Required = Required.Always, DefaultValueHandling  
= DefaultValueHandling.Ignore)]  
public string Regex { get; set; }
```

Property Value

[string](#) ↗

Result

Which result should be used from the regex? (e.g. \$1).

```
[JsonProperty(Required = Required.Always, DefaultValueHandling  
= DefaultValueHandling.Ignore)]  
public string Result { get; set; }
```

Property Value

[string](#) ↗

Class FileReaderResult

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

The Result of FileAdapter.Execute.

```
public class FileReaderResult : AdapterResultBase
```

Inheritance

[object](#) ← [AdapterResultBase](#) ← FileReaderResult

Derived

[ReaderResult](#)

Inherited Members

[AdapterResultBase.MessageBag](#) , [AdapterResultBase.Data](#) , [AdapterResultBase.Measures](#) ,
[AdapterResultBase.Findings](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Constructors

FileReaderResult()

```
public FileReaderResult()
```

FileReaderResult(FileReaderContext)

```
public FileReaderResult(FileReaderContext context)
```

Parameters

`context` [FileReaderContext](#)

Properties

HumanReadableFilePaths

```
public HashSet<string> HumanReadableFilePaths { get; set; }
```

Property Value

[HashSet](#) <[string](#)>

Class ReaderResult

Namespace: [MES.MQC.DataSourceLibrary.Models.Adapters](#)

Assembly: MES.MQC.DataSourceLibrary.dll

@deprecated For backwards compatibility only

```
public class ReaderResult : FileReaderResult
```

Inheritance

[object](#) ← [AdapterResultBase](#) ← [FileReaderResult](#) ← ReaderResult

Inherited Members

[FileReaderResult.HumanReadableFilePaths](#) , [AdapterResultBase.MessageBag](#) , [AdapterResultBase.Data](#) ,
[AdapterResultBase.Measures](#) , [AdapterResultBase.Findings](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Constructors

ReaderResult()

```
public ReaderResult()
```

ReaderResult(FileReaderContext)

```
public ReaderResult(FileReaderContext context)
```

Parameters

context [FileReaderContext](#)

Namespace MES.MQC.UtilityLibrary.Form

Classes

[FormClassAttribute](#)

[FormError](#)

[FormFieldSchema](#)

[FormPreview](#)

[FormSchema](#)

Interfaces

[IForm](#)

[IFormProvider](#)

Class FormClassAttribute

Namespace: [MES.MQC.UtilityLibrary.Form](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Class)]
public class FormClassAttribute : Attribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← FormClassAttribute

Implements

[Attribute](#)

Inherited Members

[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,
[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

Preview

```
public string Preview { get; set; }
```

Property Value

[string](#) ↗

Class FormError

Namespace: [MES.MQC.UtilityLibrary.Form](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
public class FormError
```

Inheritance

[object](#) ← FormError

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

Description

```
public string Description { get; set; }
```

Property Value

[string](#)

Message

```
public string Message { get; set; }
```

Property Value

[string](#)

Class FormFieldSchema

Namespace: [MES.MQC.UtilityLibrary.Form](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
public class FormFieldSchema
```

Inheritance

[object](#) ← FormFieldSchema

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

Config

```
public FieldAttribute Config { get; set; }
```

Property Value

[FieldAttribute](#)

Default

```
public object Default { get; set; }
```

Property Value

[object](#)

Name

```
public string Name { get; set; }
```

Property Value

[string](#)

Order

```
[JsonIgnore]  
public int Order { get; set; }
```

Property Value

[int](#)

Required

```
public bool Required { get; set; }
```

Property Value

[bool](#)

Schema

```
public FormSchema Schema { get; set; }
```

Property Value

[FormSchema](#)

Schemas

```
public Dictionary<int, FormSchema> Schemas { get; set; }
```

Property Value

[Dictionary](#)<[int](#), [FormSchema](#)>

Type

```
public string Type { get; set; }
```

Property Value

[string](#)

Class FormPreview

Namespace: [MES.MQC.UtilityLibrary.Form](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
public class FormPreview
```

Inheritance

[object](#) ← FormPreview

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

DateTime

```
public DateTime? DateTime { get; set; }
```

Property Value

[DateTime](#)?

Description

```
public string Description { get; set; }
```

Property Value

[string](#)

Title

```
public string Title { get; set; }
```

Property Value

[string](#) ↗

Class FormSchema

Namespace: [MES.MQC.UtilityLibrary.Form](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
public class FormSchema
```

Inheritance

[object](#) ← FormSchema

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

Config

```
public FormClassAttribute Config { get; set; }
```

Property Value

[FormClassAttribute](#)

Description

```
public string Description { get; set; }
```

Property Value

[string](#)

Fields

```
public List<FormFieldSchema> Fields { get; set; }
```

Property Value

[List](#)<[FormFieldSchema](#)>

Name

```
public string Name { get; set; }
```

Property Value

[string](#)

Type

```
public string Type { get; set; }
```

Property Value

[string](#)

Methods

GetField(string)

```
public FormFieldSchema GetField(string fieldName)
```

Parameters

fieldName [string](#)

Returns

[FormFieldSchema](#)

GetFieldConfig(string)

```
public FieldAttribute GetFieldConfig(string fieldName)
```

Parameters

fieldName [string](#)

Returns

[FieldAttribute](#)

GetFieldConfig<T>(string)

```
public T GetFieldConfig<T>(string fieldName) where T : FieldAttribute
```

Parameters

fieldName [string](#)

Returns

T

Type Parameters

T

GetFieldSchema(string, int?)

```
public FormSchema GetFieldSchema(string fieldName, int? index = null)
```

Parameters

`fieldName` [string](#)?

`index` [int](#)?

Returns

[FormSchema](#)

Interface IForm

Namespace: [MES.MQC.UtilityLibrary.Form](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
public interface IForm
```

Properties

FormSchema

```
FormSchema FormSchema { get; set; }
```

Property Value

[FormSchema](#)

Interface IFormProvider

Namespace: [MES.MQC.UtilityLibrary.Form](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
public interface IFormProvider
```

Properties

Description

```
string Description { get; }
```

Property Value

[string](#) ↗

Name

```
string Name { get; }
```

Property Value

[string](#) ↗

Methods

ConfigureForm(IForm, string[])

```
FormError[] ConfigureForm(IForm form, string[] modifiedFields)
```

Parameters

`form` [IForm](#)

`modifiedFields` [string](#)[]

Returns

[FormError](#)[]

PreviewForm(IForm, string, out int)

`FormPreview[] PreviewForm(IForm form, string preview, out int totalCount)`

Parameters

`form` [IForm](#)

`preview` [string](#)[]

`totalCount` [int](#)[]

Returns

[FormPreview](#)[]

Namespace MES.MQC.UtilityLibrary.Form.Field Classes

[CheckboxAttribute](#)

[CheckboxGroupAttribute](#)

[DatePickerAttribute](#)

[FieldAttribute](#)

[FormAttribute](#)

[FormsAttribute](#)

[InputAttribute](#)

[InputNumberAttribute](#)

[MonthPickerAttribute](#)

[RadioGroupAttribute](#)

[SelectAttribute](#)

[SelectAttribute.SelectOption](#)

[SwitchAttribute](#)

[TextareaAttribute](#)

[TimePickerAttribute](#)

[WeekPickerAttribute](#)

Enums

[FormDisplayMode](#)

[FormsDisplayMode](#)

[InputType](#)

[SelectMode](#)

Class CheckboxAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class CheckboxAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← [CheckboxAttribute](#)

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#), [FieldAttribute.Label](#), [FieldAttribute.LabelHidden](#), [FieldAttribute.Help](#),
[FieldAttribute.Visible](#), [FieldAttribute.Disabled](#), [FieldAttribute.RefreshOnModified](#),
[FieldAttribute.Preview](#), [FieldAttribute.AllowEnumType\(\)](#), [FieldAttribute.AllowClassType\(\)](#),
[FieldAttribute.IsArrayType\(\)](#), [Attribute.GetCustomAttributes\(MemberInfo, Type\)](#),
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(MemberInfo\)](#), [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#),
[Attribute.IsDefined\(MemberInfo, Type\)](#), [Attribute.IsDefined\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#),
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo\)](#), [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#), [Attribute.IsDefined\(ParameterInfo, Type\)](#),
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#), [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#),
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(Module, Type\)](#), [Attribute.GetCustomAttributes\(Module\)](#),
[Attribute.GetCustomAttributes\(Module, bool\)](#), [Attribute.GetCustomAttributes\(Module, Type, bool\)](#),
[Attribute.IsDefined\(Module, Type\)](#), [Attribute.IsDefined\(Module, Type, bool\)](#),
[Attribute.GetCustomAttribute\(Module, Type\)](#), [Attribute.GetCustomAttribute\(Module, Type, bool\)](#),
[Attribute.GetCustomAttributes\(Assembly, Type\)](#),
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#), [Attribute.GetCustomAttributes\(Assembly\)](#),
[Attribute.GetCustomAttributes\(Assembly, bool\)](#), [Attribute.IsDefined\(Assembly, Type\)](#),
[Attribute.IsDefined\(Assembly, Type, bool\)](#), [Attribute.GetCustomAttribute\(Assembly, Type\)](#),
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#), [Attribute.Equals\(object\)](#),
[Attribute.GetHashCode\(\)](#), [Attribute.Match\(object\)](#), [Attribute.IsDefaultAttribute\(\)](#),

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

CheckLabel

```
public string CheckLabel { get; set; }
```

Property Value

[string](#)

Methods

GetBasicTypes()

Allowed basic types for PropertyType

```
public override Type[] GetBasicTypes()
```

Returns

[Type](#)[]

Class CheckboxGroupAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class CheckboxGroupAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← [CheckboxGroupAttribute](#)

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#) , [FieldAttribute.Label](#) , [FieldAttribute.LabelHidden](#) , [FieldAttribute.Help](#) ,
[FieldAttribute.Visible](#) , [FieldAttribute.Disabled](#) , [FieldAttribute.RefreshOnModified](#) ,
[FieldAttribute.Preview](#) , [FieldAttribute.AllowClassType\(\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

Options

```
public string[] Options { get; set; }
```

Property Value

[string](#)[]

Methods

AllowEnumType()

.PropertyType for this attribute can be a enum

```
public override bool AllowEnumType()
```

Returns

[bool](#)

GetBasicTypes()

Allowed basic types for PropertyType

```
public override Type[] GetBasicTypes()
```

Returns

[Type](#)[]

IsArrayType()

.PropertyType for this attribute is an array of enum or a basic type

```
public override bool IsArrayType()
```

Returns

[bool](#)

Class DatePickerAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class DatePickerAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← DatePickerAttribute

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#) , [FieldAttribute.Label](#) , [FieldAttribute.LabelHidden](#) , [FieldAttribute.Help](#) ,
[FieldAttribute.Visible](#) , [FieldAttribute.Disabled](#) , [FieldAttribute.RefreshOnModified](#) ,
[FieldAttribute.Preview](#) , [FieldAttribute.AllowEnumType\(\)](#) , [FieldAttribute.AllowClassType\(\)](#) ,
[FieldAttribute.IsArrayType\(\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

MaxDateTime

```
public string MaxDateTime { get; set; }
```

Property Value

[string](#)

MinDateTime

```
public string MinDateTime { get; set; }
```

Property Value

[string](#)

ShowTime

```
public bool ShowTime { get; set; }
```

Property Value

[bool](#)

Methods

GetBasicTypes()

Allowed basic types for PropertyType

```
public override Type[] GetBasicTypes()
```

Returns

Type[]

Class FieldAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
public abstract class FieldAttribute : Attribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← FieldAttribute

Implements

[Attribute](#)

Derived

[CheckboxAttribute](#), [CheckboxGroupAttribute](#), [DatePickerAttribute](#), [FormAttribute](#), [FormsAttribute](#),
[InputAttribute](#), [InputNumberAttribute](#), [MonthPickerAttribute](#), [RadioGroupAttribute](#), [SelectAttribute](#),
[SwitchAttribute](#), [TextareaAttribute](#), [TimePickerAttribute](#), [WeekPickerAttribute](#)

Inherited Members

[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,

[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,
[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Constructors

FieldAttribute()

```
protected FieldAttribute()
```

Properties

Disabled

```
public bool Disabled { get; set; }
```

Property Value

[bool](#)

Help

```
public string Help { get; set; }
```

Property Value

[string](#)

Label

```
public string Label { get; set; }
```

Property Value

[string](#) ↗

LabelHidden

`public bool LabelHidden { get; set; }`

Property Value

[bool](#) ↗

Preview

`public string Preview { get; set; }`

Property Value

[string](#) ↗

RefreshOnModified

`public bool RefreshOnModified { get; set; }`

Property Value

[bool](#) ↗

Type

[JsonProperty(Required = Required.Always, Order = -2147483648, PropertyName = "\$type")]
`public string Type { get; }`

Property Value

[string](#) ↗

Visible

```
public bool Visible { get; set; }
```

Property Value

[bool](#)

Methods

AllowClassType()

.PropertyType for this attribute can be a class, that contains at least one property with a FieldAttribute

```
public virtual bool AllowClassType()
```

Returns

[bool](#)

AllowEnumType()

.PropertyType for this attribute can be a enum

```
public virtual bool AllowEnumType()
```

Returns

[bool](#)

GetBasicTypes()

Allowed basic types for PropertyType

```
public virtual Type[] GetBasicTypes()
```

Returns

Type[]

IsArrayType()

.PropertyType for this attribute is an array of enum or a basic type

```
public virtual bool IsArrayType()
```

Returns

bool

Class FormAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class FormAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← FormAttribute

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#) , [FieldAttribute.Label](#) , [FieldAttribute.LabelHidden](#) , [FieldAttribute.Help](#) ,
[FieldAttribute.Visible](#) , [FieldAttribute.Disabled](#) , [FieldAttribute.RefreshOnModified](#) ,
[FieldAttribute.Preview](#) , [FieldAttribute.GetBasicTypes\(\)](#) , [FieldAttribute.AllowEnumType\(\)](#) ,
[FieldAttribute.IsArrayType\(\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

DisplayMode

```
public FormDisplayStyle DisplayMode { get; set; }
```

Property Value

[FormDisplayStyle](#)

Methods

AllowClassType()

.PropertyType for this attribute can be a class, that contains at least one property with a FieldAttribute

```
public override bool AllowClassType()
```

Returns

[bool](#)

Enum FormDisplayMode

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[JsonConverter(typeof(StringEnumWithDefaultConverter), new object[] {  
    FormDisplayMode.Default })]  
public enum FormDisplayMode
```

Fields

Default = 0

Vertical = 1

Class FormsAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class FormsAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← FormsAttribute

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#) , [FieldAttribute.Label](#) , [FieldAttribute.LabelHidden](#) , [FieldAttribute.Help](#) ,
[FieldAttribute.Visible](#) , [FieldAttribute.Disabled](#) , [FieldAttribute.RefreshOnModified](#) ,
[FieldAttribute.Preview](#) , [FieldAttribute.GetBasicTypes\(\)](#) , [FieldAttribute.AllowEnumType\(\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

AddLabel

```
public string AddLabel { get; set; }
```

Property Value

[string](#)

DisplayMode

```
public FormsDisplayStyle DisplayMode { get; set; }
```

Property Value

[FormsDisplayStyle](#)

MaxItems

```
public int MaxItems { get; set; }
```

Property Value

[int](#)

MinItems

```
public int MinItems { get; set; }
```

Property Value

[int](#)

TitleFields

```
public string[] TitleFields { get; set; }
```

Property Value

[string](#)[]

Methods

AllowClassType()

.PropertyType for this attribute can be a class, that contains at least one property with a FieldAttribute

```
public override bool AllowClassType()
```

Returns

[bool](#)

IsArrayType()

.PropertyType for this attribute is an array of enum or a basic type

```
public override bool IsArrayType()
```

Returns

[bool](#)

Enum FormsDisplayMode

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[JsonConverter(typeof(StringEnumWithDefaultConverter), new object[] {  
    FormsDisplayStyle.Default })]  
public enum FormsDisplayStyle
```

Fields

Compact = 2

Default = 0

Vertical = 1

Class InputAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class InputAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← [InputAttribute](#)

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#), [FieldAttribute.Label](#), [FieldAttribute.LabelHidden](#), [FieldAttribute.Help](#),
[FieldAttribute.Visible](#), [FieldAttribute.Disabled](#), [FieldAttribute.RefreshOnModified](#),
[FieldAttribute.Preview](#), [FieldAttribute.AllowEnumType\(\)](#), [FieldAttribute.AllowClassType\(\)](#),
[FieldAttribute.IsArrayType\(\)](#), [Attribute.GetCustomAttributes\(MemberInfo, Type\)](#),
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(MemberInfo\)](#), [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#),
[Attribute.IsDefined\(MemberInfo, Type\)](#), [Attribute.IsDefined\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#),
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo\)](#), [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#), [Attribute.IsDefined\(ParameterInfo, Type\)](#),
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#), [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#),
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(Module, Type\)](#), [Attribute.GetCustomAttributes\(Module\)](#),
[Attribute.GetCustomAttributes\(Module, bool\)](#), [Attribute.GetCustomAttributes\(Module, Type, bool\)](#),
[Attribute.IsDefined\(Module, Type\)](#), [Attribute.IsDefined\(Module, Type, bool\)](#),
[Attribute.GetCustomAttribute\(Module, Type\)](#), [Attribute.GetCustomAttribute\(Module, Type, bool\)](#),
[Attribute.GetCustomAttributes\(Assembly, Type\)](#),
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#), [Attribute.GetCustomAttributes\(Assembly\)](#),
[Attribute.GetCustomAttributes\(Assembly, bool\)](#), [Attribute.IsDefined\(Assembly, Type\)](#),
[Attribute.IsDefined\(Assembly, Type, bool\)](#), [Attribute.GetCustomAttribute\(Assembly, Type\)](#),
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#), [Attribute.Equals\(object\)](#),
[Attribute.GetHashCode\(\)](#), [Attribute.Match\(object\)](#), [Attribute.IsDefaultAttribute\(\)](#),

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

InputType

```
public InputType InputType { get; set; }
```

Property Value

[InputType](#)

MaxLength

```
public int MaxLength { get; set; }
```

Property Value

[int](#)

MinLength

```
public int MinLength { get; set; }
```

Property Value

[int](#)

Methods

GetBasicTypes()

Allowed basic types for PropertyType

```
public override Type[] GetBasicTypes()
```

Returns

Type[]

Class InputNumberAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class InputNumberAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← [InputNumberAttribute](#)

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#), [FieldAttribute.Label](#), [FieldAttribute.LabelHidden](#), [FieldAttribute.Help](#),
[FieldAttribute.Visible](#), [FieldAttribute.Disabled](#), [FieldAttribute.RefreshOnModified](#),
[FieldAttribute.Preview](#), [FieldAttribute.AllowEnumType\(\)](#), [FieldAttribute.AllowClassType\(\)](#),
[FieldAttribute.IsArrayType\(\)](#), [Attribute.GetCustomAttributes\(MemberInfo, Type\)](#),
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(MemberInfo\)](#), [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#),
[Attribute.IsDefined\(MemberInfo, Type\)](#), [Attribute.IsDefined\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#),
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo\)](#), [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#), [Attribute.IsDefined\(ParameterInfo, Type\)](#),
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#), [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#),
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(Module, Type\)](#), [Attribute.GetCustomAttributes\(Module\)](#),
[Attribute.GetCustomAttributes\(Module, bool\)](#), [Attribute.GetCustomAttributes\(Module, Type, bool\)](#),
[Attribute.IsDefined\(Module, Type\)](#), [Attribute.IsDefined\(Module, Type, bool\)](#),
[Attribute.GetCustomAttribute\(Module, Type\)](#), [Attribute.GetCustomAttribute\(Module, Type, bool\)](#),
[Attribute.GetCustomAttributes\(Assembly, Type\)](#),
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#), [Attribute.GetCustomAttributes\(Assembly\)](#),
[Attribute.GetCustomAttributes\(Assembly, bool\)](#), [Attribute.IsDefined\(Assembly, Type\)](#),
[Attribute.IsDefined\(Assembly, Type, bool\)](#), [Attribute.GetCustomAttribute\(Assembly, Type\)](#),
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#), [Attribute.Equals\(object\)](#),
[Attribute.GetHashCode\(\)](#), [Attribute.Match\(object\)](#), [Attribute.IsDefaultAttribute\(\)](#),

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

Format

```
public string Format { get; set; }
```

Property Value

[string](#)

Max

```
public double Max { get; set; }
```

Property Value

[double](#)

Min

```
public double Min { get; set; }
```

Property Value

[double](#)

Precision

```
public int Precision { get; set; }
```

Property Value

[int ↗](#)

Step

```
public double Step { get; set; }
```

Property Value

[double ↗](#)

Methods

GetBasicTypes()

Allowed basic types for PropertyType

```
public override Type[] GetBasicTypes()
```

Returns

[Type ↗](#)[]

Enum InputType

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[JsonConverter(typeof(StringEnumWithDefaultConverter), new object[] { InputType.Text })]  
public enum InputType
```

Fields

Email = 1

Password = 2

Regex = 5

Tel = 3

Text = 0

Url = 4

Class MonthPickerAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class MonthPickerAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← MonthPickerAttribute

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#) , [FieldAttribute.Label](#) , [FieldAttribute.LabelHidden](#) , [FieldAttribute.Help](#) ,
[FieldAttribute.Visible](#) , [FieldAttribute.Disabled](#) , [FieldAttribute.RefreshOnModified](#) ,
[FieldAttribute.Preview](#) , [FieldAttribute.AllowEnumType\(\)](#) , [FieldAttribute.AllowClassType\(\)](#) ,
[FieldAttribute.IsArrayType\(\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

MaxDateTime

```
public string MaxDateTime { get; set; }
```

Property Value

[string](#)

MinDateTime

```
public string MinDateTime { get; set; }
```

Property Value

[string](#)

Methods

GetBasicTypes()

Allowed basic types for PropertyType

```
public override Type[] GetBasicTypes()
```

Returns

[Type](#)[]

Class RadioGroupAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class RadioGroupAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← RadioGroupAttribute

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#), [FieldAttribute.Label](#), [FieldAttribute.LabelHidden](#), [FieldAttribute.Help](#),
[FieldAttribute.Visible](#), [FieldAttribute.Disabled](#), [FieldAttribute.RefreshOnModified](#),
[FieldAttribute.Preview](#), [FieldAttribute.AllowClassType\(\)](#), [FieldAttribute.IsArrayType\(\)](#),
[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#),
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(MemberInfo\)](#), [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#),
[Attribute.IsDefined\(MemberInfo, Type\)](#), [Attribute.IsDefined\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#),
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo\)](#), [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#), [Attribute.IsDefined\(ParameterInfo, Type\)](#),
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#), [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#),
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(Module, Type\)](#), [Attribute.GetCustomAttributes\(Module\)](#),
[Attribute.GetCustomAttributes\(Module, bool\)](#), [Attribute.GetCustomAttributes\(Module, Type, bool\)](#),
[Attribute.IsDefined\(Module, Type\)](#), [Attribute.IsDefined\(Module, Type, bool\)](#),
[Attribute.GetCustomAttribute\(Module, Type\)](#), [Attribute.GetCustomAttribute\(Module, Type, bool\)](#),
[Attribute.GetCustomAttributes\(Assembly, Type\)](#),
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#), [Attribute.GetCustomAttributes\(Assembly\)](#),
[Attribute.GetCustomAttributes\(Assembly, bool\)](#), [Attribute.IsDefined\(Assembly, Type\)](#),
[Attribute.IsDefined\(Assembly, Type, bool\)](#), [Attribute.GetCustomAttribute\(Assembly, Type\)](#),
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#), [Attribute.Equals\(object\)](#),
[Attribute.GetHashCode\(\)](#), [Attribute.Match\(object\)](#), [Attribute.IsDefaultAttribute\(\)](#),

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

Options

```
public string[] Options { get; set; }
```

Property Value

[string](#)[]

Methods

AllowEnumType()

.PropertyType for this attribute can be a enum

```
public override bool AllowEnumType()
```

Returns

[bool](#)

GetBasicTypes()

Allowed basic types for PropertyType

```
public override Type[] GetBasicTypes()
```

Returns

[Type](#)[]

Class SelectAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class SelectAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← [SelectAttribute](#)

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#) , [FieldAttribute.Label](#) , [FieldAttribute.LabelHidden](#) , [FieldAttribute.Help](#) ,
[FieldAttribute.Visible](#) , [FieldAttribute.Disabled](#) , [FieldAttribute.RefreshOnModified](#) ,
[FieldAttribute.Preview](#) , [FieldAttribute.AllowClassType\(\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

Mode

```
public SelectMode Mode { get; set; }
```

Property Value

[SelectMode](#)

Options

```
public string[] Options { get; set; }
```

Property Value

[string](#)[]

Methods

AllowEnumType()

.PropertyType for this attribute can be a enum

```
public override bool AllowEnumType()
```

Returns

[bool](#)

GetBasicTypes()

Allowed basic types for PropertyType

```
public override Type[] GetBasicTypes()
```

Returns

[Type](#)[]

IsArrayType()

.PropertyType for this attribute is an array of enum or a basic type

```
public override bool IsArrayType()
```

Returns

[bool](#)

Class SelectAttribute.SelectOption

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
public class SelectAttribute.SelectOption
```

Inheritance

[object](#) ← SelectAttribute.SelectOption

Inherited Members

[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#)

Properties

Label

```
public string Label { get; set; }
```

Property Value

[string](#)

Value

```
public object Value { get; set; }
```

Property Value

[object](#)

Enum SelectMode

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[JsonConverter(typeof(StringEnumWithDefaultConverter), new object[] { SelectMode.Default })]  
public enum SelectMode
```

Fields

Default = 0

Multiple = 1

Class SwitchAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class SwitchAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← SwitchAttribute

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#) , [FieldAttribute.Label](#) , [FieldAttribute.LabelHidden](#) , [FieldAttribute.Help](#) ,
[FieldAttribute.Visible](#) , [FieldAttribute.Disabled](#) , [FieldAttribute.RefreshOnModified](#) ,
[FieldAttribute.Preview](#) , [FieldAttribute.AllowEnumType\(\)](#) , [FieldAttribute.AllowClassType\(\)](#) ,
[FieldAttribute.IsArrayType\(\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

CheckedIcon

```
public string CheckedIcon { get; set; }
```

Property Value

[string](#)

CheckedTooltip

```
public string CheckedTooltip { get; set; }
```

Property Value

[string](#)

SuccessDangerColoring

```
public bool SuccessDangerColoring { get; set; }
```

Property Value

[bool](#)

UncheckedIcon

```
public string UncheckedIcon { get; set; }
```

Property Value

[string](#) ↗

UncheckedTooltip

```
public string UncheckedTooltip { get; set; }
```

Property Value

[string](#) ↗

Methods

GetBasicTypes()

Allowed basic types for PropertyType

```
public override Type[] GetBasicTypes()
```

Returns

[Type](#) ↗ []

Class TextareaAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class TextareaAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← [TextareaAttribute](#)

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#), [FieldAttribute.Label](#), [FieldAttribute.LabelHidden](#), [FieldAttribute.Help](#),
[FieldAttribute.Visible](#), [FieldAttribute.Disabled](#), [FieldAttribute.RefreshOnModified](#),
[FieldAttribute.Preview](#), [FieldAttribute.AllowEnumType\(\)](#), [FieldAttribute.AllowClassType\(\)](#),
[FieldAttribute.IsArrayType\(\)](#), [Attribute.GetCustomAttributes\(MemberInfo, Type\)](#),
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(MemberInfo\)](#), [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#),
[Attribute.IsDefined\(MemberInfo, Type\)](#), [Attribute.IsDefined\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#),
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo\)](#), [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#), [Attribute.IsDefined\(ParameterInfo, Type\)](#),
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#), [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#),
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#),
[Attribute.GetCustomAttributes\(Module, Type\)](#), [Attribute.GetCustomAttributes\(Module\)](#),
[Attribute.GetCustomAttributes\(Module, bool\)](#), [Attribute.GetCustomAttributes\(Module, Type, bool\)](#),
[Attribute.IsDefined\(Module, Type\)](#), [Attribute.IsDefined\(Module, Type, bool\)](#),
[Attribute.GetCustomAttribute\(Module, Type\)](#), [Attribute.GetCustomAttribute\(Module, Type, bool\)](#),
[Attribute.GetCustomAttributes\(Assembly, Type\)](#),
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#), [Attribute.GetCustomAttributes\(Assembly\)](#),
[Attribute.GetCustomAttributes\(Assembly, bool\)](#), [Attribute.IsDefined\(Assembly, Type\)](#),
[Attribute.IsDefined\(Assembly, Type, bool\)](#), [Attribute.GetCustomAttribute\(Assembly, Type\)](#),
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#), [Attribute.Equals\(object\)](#),
[Attribute.GetHashCode\(\)](#), [Attribute.Match\(object\)](#), [Attribute.IsDefaultAttribute\(\)](#),

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

MaxLength

```
public int MaxLength { get; set; }
```

Property Value

[int](#)

MaxRows

```
public int MaxRows { get; set; }
```

Property Value

[int](#)

MinRows

```
public int MinRows { get; set; }
```

Property Value

[int](#)

Methods

GetBasicTypes()

Allowed basic types for PropertyType

```
public override Type[] GetBasicTypes()
```

Returns

Type[]

Class TimePickerAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class TimePickerAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← TimePickerAttribute

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#) , [FieldAttribute.Label](#) , [FieldAttribute.LabelHidden](#) , [FieldAttribute.Help](#) ,
[FieldAttribute.Visible](#) , [FieldAttribute.Disabled](#) , [FieldAttribute.RefreshOnModified](#) ,
[FieldAttribute.Preview](#) , [FieldAttribute.AllowEnumType\(\)](#) , [FieldAttribute.AllowClassType\(\)](#) ,
[FieldAttribute.IsArrayType\(\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Methods

GetBasicTypes()

Allowed basic types for PropertyType

```
public override Type[] GetBasicTypes()
```

Returns

[Type](#)[]

Class WeekPickerAttribute

Namespace: [MES.MQC.UtilityLibrary.Form.Field](#)

Assembly: MES.MQC.UtilityLibrary.dll

```
[AttributeUsage(AttributeTargets.Property)]
public class WeekPickerAttribute : FieldAttribute, _Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [FieldAttribute](#) ← WeekPickerAttribute

Implements

[Attribute](#)

Inherited Members

[FieldAttribute.Type](#) , [FieldAttribute.Label](#) , [FieldAttribute.LabelHidden](#) , [FieldAttribute.Help](#) ,
[FieldAttribute.Visible](#) , [FieldAttribute.Disabled](#) , [FieldAttribute.RefreshOnModified](#) ,
[FieldAttribute.Preview](#) , [FieldAttribute.AllowEnumType\(\)](#) , [FieldAttribute.AllowClassType\(\)](#) ,
[FieldAttribute.IsArrayType\(\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,

[Attribute.TypeId](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

Properties

MaxDateTime

```
public string MaxDateTime { get; set; }
```

Property Value

[string](#)

MinDateTime

```
public string MinDateTime { get; set; }
```

Property Value

[string](#)

Methods

GetBasicTypes()

Allowed basic types for PropertyType

```
public override Type[] GetBasicTypes()
```

Returns

[Type](#)[]