

# Namespace MB.Common.Aspects

## Classes

[LogAspect](#)

# Class LogAspect

Namespace: [MB.Common.Aspects](#)

Assembly: MB.Common.dll

```
public sealed class LogAspect : OnMethodBoundaryAspect, _Attribute
```

## Inheritance

[object](#) ↗ ← [Attribute](#) ↗ ← OnMethodBoundaryAspect ← LogAspect

## Implements

[Attribute](#) ↗

## Inherited Members

[OnMethodBoundaryAspect.CompileTimeValidate\(MethodBase\)](#) ↗ ,  
[OnMethodBoundaryAspect.AttributeTargetMemberAttributes](#) ,  
[OnMethodBoundaryAspect.NamespaceFilter](#) , [OnMethodBoundaryAspect.TypeNameFilter](#) ,  
[OnMethodBoundaryAspect.MethodNameFilter](#) , [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\(\)](#)

## Constructors

### LogAspect()

```
public LogAspect()
```

## Methods

### OnEntry(MethodExecutionArgs)

```
public override void OnEntry(MethodExecutionArgs args)
```

#### Parameters

**args** MethodExecutionArgs

### OnException(MethodExecutionArgs)

```
public override void OnException(MethodExecutionArgs args)
```

#### Parameters

**args** MethodExecutionArgs

### OnExit(MethodExecutionArgs)

```
public override void OnExit(MethodExecutionArgs args)
```

#### Parameters

**args** MethodExecutionArgs

# Namespace MB.Common.Attributes

## Classes

### [ErrorInformationAttribute](#)

### [GuidAttribute](#)

Represents a custom attribute that can be applied to fields to associate them with a specific GUID.

### [RequiredSharedParametersAttribute](#)

Represents an attribute to specify required shared parameters for a given target.

### [RevitCategoriesAttribute](#)

Represents an attribute to associate a target with a set of Revit built-in categories.

### [SharedParameterTypeAttribute](#)

Represents an attribute to specify the type of a shared parameter.

# Class ErrorInformationAttribute

Namespace: [MB.Common.Attributes](#)

Assembly: MB.Common.dll

```
[AttributeUsage(AttributeTargets.Field)]
public class ErrorInformationAttribute : Attribute, _Attribute
```

## Inheritance

[object](#) ↗ ← [Attribute](#) ↗ ← ErrorInformationAttribute

## 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\(\)](#) ↗

## Constructors

### ErrorInformationAttribute(string)

```
public ErrorInformationAttribute(string errorInformation)
```

#### Parameters

errorInformation [string](#)

## Properties

### ErrorInformation

```
public string ErrorInformation { get; }
```

#### Property Value

[string](#)

# Class GuidAttribute

Namespace: [MB.Common.Attributes](#)

Assembly: MB.Common.dll

Represents a custom attribute that can be applied to fields to associate them with a specific GUID.

```
[AttributeUsage(AttributeTargets.Field)]
public class GuidAttribute : Attribute, _Attribute
```

## Inheritance

[object](#) ← [Attribute](#) ← GuidAttribute

## 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\(\)](#)

## Remarks

This attribute allows developers to tag fields with a GUID value, which can then be used for identification, serialization, or other purposes where a globally unique identifier is required.

## Constructors

### GuidAttribute(string)

Initializes a new instance of the [GuidAttribute](#) class with the specified GUID string.

```
public GuidAttribute(string guid)
```

#### Parameters

**guid** [string](#)

The GUID string to associate with the field. Must be in a valid GUID format.

#### Exceptions

[FormatException](#)

Thrown when the provided string is not in a valid GUID format.

## Properties

### Guid

Gets the GUID associated with the field to which this attribute is applied.

```
public string Guid { get; }
```

#### Property Value

[string](#)

# Class RequiredSharedParametersAttribute

Namespace: [MB.Common.Attributes](#)

Assembly: MB.Common.dll

Represents an attribute to specify required shared parameters for a given target.

```
[AttributeUsage(AttributeTargets.Field)]
public class RequiredSharedParametersAttribute : Attribute, _Attribute
```

## Inheritance

[object](#) ← [Attribute](#) ← RequiredSharedParametersAttribute

## 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\(\)](#)

## Remarks

This attribute allows developers to declare a set of shared parameters that are required for the functionality or configuration of a specific target (e.g., class, method, or property).

## Constructors

### RequiredSharedParametersAttribute(params ToolsSharedParameter[])

Initializes a new instance of the [RequiredSharedParametersAttribute](#) class.

```
public RequiredSharedParametersAttribute(params ToolsSharedParameter[] sharedParameters)
```

#### Parameters

##### sharedParameters [ToolsSharedParameter\[\]](#)

A variable-length array of [ToolsSharedParameter](#) objects that represent the required shared parameters.

#### Exceptions

##### ArgumentNullException

Thrown if the provided `sharedParameters` array is null.

## Properties

### SharedParameters

Gets the array of shared parameters that are required.

```
public ToolsSharedParameter[] SharedParameters { get; }
```

#### Property Value

##### [ToolsSharedParameter\[\]](#)

# Class RevitCategoriesAttribute

Namespace: [MB.Common.Attributes](#)

Assembly: MB.Common.dll

Represents an attribute to associate a target with a set of Revit built-in categories.

```
[AttributeUsage(AttributeTargets.Field)]
public class RevitCategoriesAttribute : Attribute, _Attribute
```

## Inheritance

[object](#) ← [Attribute](#) ← RevitCategoriesAttribute

## 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\(\)](#)

## Remarks

This attribute can be used to declare which Revit built-in categories are relevant for a specific class, method, or property. It allows for filtering or categorization based on Revit's predefined categories.

## Constructors

### RevitCategoriesAttribute(params BuiltInCategory[])

Initializes a new instance of the [RevitCategoriesAttribute](#) class.

```
public RevitCategoriesAttribute(params BuiltInCategory[] categories)
```

#### Parameters

**categories** BuiltInCategory[]

A variable-length array of BuiltInCategory enums that represent the relevant Revit categories.

#### Exceptions

ArgumentNullException

Thrown if the provided **categories** array is null.

## Properties

### Categories

Gets the array of Revit built-in categories associated with the target.

```
public BuiltInCategory[] Categories { get; }
```

#### Property Value

BuiltInCategory[]

# Class SharedParameterTypeAttribute

Namespace: [MB.Common.Attributes](#)

Assembly: MB.Common.dll

Represents an attribute to specify the type of a shared parameter.

```
[AttributeUsage(AttributeTargets.Class|AttributeTargets.Property|AttributeTargets.Field)]
public class SharedParameterTypeAttribute : Attribute, _Attribute
```

## Inheritance

[object](#) ← [Attribute](#) ← SharedParameterTypeAttribute

## 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\(\)](#)

## Remarks

This attribute is used to associate a target (e.g., a class, method, or property) with a specific shared parameter type. It helps define the type of shared parameter in a structured and reusable way.

## Constructors

### SharedParameterTypeAttribute(SharedParameterType)

Initializes a new instance of the [SharedParameterTypeAttribute](#) class.

```
public SharedParameterTypeAttribute(SharedParameterType type)
```

#### Parameters

**type** [SharedParameterType](#)

The [SharedParameterType](#) to associate with the target.

#### Exceptions

ArgumentNullException

Thrown if the provided **type** is null (only if SharedParameterType is a reference type).

## Properties

### Type

Gets the type of the shared parameter associated with this attribute.

```
public SharedParameterType Type { get; }
```

### Property Value

[SharedParameterType](#)

# Namespace MB.Common.Comparers

## Classes

### [BoundingBoxComparer](#)

A utility class for comparing and resolving bounding box coordinates from multiple points.

# Class BoundingBoxComparer

Namespace: [MB.Common.Comparers](#)

Assembly: MB.Common.dll

A utility class for comparing and resolving bounding box coordinates from multiple points.

```
public class BoundingBoxComparer
```

## Inheritance

[object](#) ← BoundingBoxComparer

## Inherited Members

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

## Constructors

### BoundingBoxComparer()

Initializes a new instance of the [BoundingBoxComparer](#) class. Sets initial values for bounds: maximum values for minimum bounds and minimum values for maximum bounds.

```
public BoundingBoxComparer()
```

## Methods

### Compare(XYZ)

Compares the given point with the current bounds and updates the bounds accordingly.

```
public void Compare(XYZ point)
```

## Parameters

**point** XYZ

The 3D point to compare.

## Resolve()

Resolves the current bounds into a minimum and maximum point, extending them by a fixed tolerance.

```
public (XYZ min, XYZ max) Resolve()
```

Returns

(XYZ [min](#), XYZ [max](#))

A tuple containing:

- The minimum point (Autodesk.Revit.DB.XYZ) of the bounding box.
- The maximum point (Autodesk.Revit.DB.XYZ) of the bounding box.

# Namespace MB.Common.Constants

## Classes

[FilePaths](#)

[MbErrorMessages](#)

[MbfFiles](#)

Provides constants and utility properties for MBF-related file paths.

[MbfGroups](#)

Provides constant values for MBF parameter groups.

# Class FilePaths

Namespace: [MB.Common.Constants](#)

Assembly: MB.Common.dll

```
public static class FilePaths
```

## Inheritance

[object](#) ← FilePaths

## Inherited Members

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

# Fields

## Elbows

```
public const string Elbows = "Elbows"
```

### Field Value

[string](#)

## FlexPipes

```
public const string FlexPipes = "Flex Pipes"
```

### Field Value

[string](#)

## Others

```
public const string Others = "Others"
```

Field Value

[string](#) ↗

## Outlets

```
public const string Outlets = "Outlets"
```

Field Value

[string](#) ↗

## PipeAccessories

```
public const string PipeAccessories = "Pipe Accessories"
```

Field Value

[string](#) ↗

## Pipes

```
public const string Pipes = "Pipes"
```

Field Value

[string](#) ↗

## Sprinklers

```
public const string Sprinklers = "Sprinklers"
```

Field Value

[string](#)

Tees

```
public const string Tees = "Tees"
```

Field Value

[string](#)

## Properties

CoverageExcel

Path to the Coverage Excel file.

```
public static string CoverageExcel { get; }
```

Property Value

[string](#)

CoverageJson

Path to the Coverage JSON file.

```
public static string CoverageJson { get; }
```

Property Value

[string](#)

## MapperJsons

Path to the Mapper JSON file.

```
public static string MapperJsons { get; }
```

Property Value

[string](#)

## WorksetsExcel

Path to the Worksets Excel file.

```
public static string WorksetsExcel { get; }
```

Property Value

[string](#)

## WorksetsJson

Path to the Worksets JSON file.

```
public static string WorksetsJson { get; }
```

Property Value

[string](#)

# Class MbErrorMessages

Namespace: [MB.Common.Constants](#)

Assembly: MB.Common.dll

```
public static class MbErrorMessages
```

## Inheritance

[object](#) ← MbErrorMessages

## Inherited Members

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

## Fields

### MoreThanOneSpaceSelected

```
public const string MoreThanOneSpaceSelected = "Please select only one space."
```

#### Field Value

[string](#)

### NoSpaceSelected

```
public const string NoSpaceSelected = "Please select the space."
```

#### Field Value

[string](#)

### NoSprinklerSelected

```
public const string NoSprinklerSelected = "Please select atleast one sprinkler."
```

Field Value

[string](#) ↗

## SprinklerNotInsideSpace

```
public const string SprinklerNotInsideSpace = "Sprinkler is not inside the selected space."
```

Field Value

[string](#) ↗

# Class Mbffiles

Namespace: [MB.Common.Constants](#)

Assembly: MB.Common.dll

Provides constants and utility properties for MBF-related file paths.

```
public static class Mbffiles
```

## Inheritance

[object](#) ← Mbffiles

## Inherited Members

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

# Properties

## AddinDirectory

Gets the directory of the currently executing assembly (Add-in directory).

```
public static string AddinDirectory { get; }
```

## Property Value

[string](#)

## Remarks

This property dynamically resolves the directory path where the executing assembly is located.

## ToolsSharedParameter

Gets the full path to the shared parameter file (3dfirespf.txt) used by MBF.

```
public static string ToolsSharedParameter { get; }
```

## Property Value

[string](#) ↗

# Class MbGroups

Namespace: [MB.Common.Constants](#)

Assembly: MB.Common.dll

Provides constant values for MBF parameter groups.

```
public static class MbGroups
```

## Inheritance

[object](#) ← MbGroups

## Inherited Members

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

## Remarks

This class contains group names for organizing parameters used in the MBF suite. These constants can be used to ensure consistent naming throughout the application.

## Fields

### Fittings

Represents the group name for pipe accessory fittings parameters.

```
public const string Fittings = "MBFire-Pipes-Accessory-Fittings-Parameters"
```

### Field Value

[string](#)

# Namespace MB.Common.Errors

## Classes

[MbErrorExtensions](#)

## Enums

[MbError](#)

# Enum MbError

Namespace: [MB.Common.Errors](#)

Assembly: MB.Common.dll

```
public enum MbError
```

## Extension Methods

[MbErrorExtensions.GetErrorMessage\(MbError\)](#)

## Fields

[ErrorInformation("Please select only one space.")] MultipleSpacesSelected = 2

[ErrorInformation("Please select the space.")] NoSpaceSelected = 1

[ErrorInformation("Please select atleast one sprinkler.")] NoSprinklerSelected = 3

[ErrorInformation("Sprinkler is not inside the selected space.")]

SprinklersNotInsideSpace = 4

# Class MbErrorExtensions

Namespace: [MB.Common.Errors](#)

Assembly: MB.Common.dll

```
public static class MbErrorExtensions
```

## Inheritance

[object](#) ← MbErrorExtensions

## Inherited Members

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

# Methods

## GetErrorMessage(MbError)

Retrieves the error information associated with the specified [MbError](#).

```
public static string GetErrorMessage(this MbError error)
```

### Parameters

error [MbError](#)

The [MbError](#) instance for which the error message is retrieved.

### Returns

[string](#)

# Namespace MB.Common.Errors.Exceptions

## Classes

[MbException](#)

# Class MbException

Namespace: [MB.Common.Errors.Exceptions](#)

Assembly: MB.Common.dll

```
public class MbException : Exception, ISerializable, _Exception
```

## Inheritance

[object](#) ← [Exception](#) ← MbException

## Implements

[ISerializable](#), [Exception](#)

## Inherited Members

[Exception.GetBaseException\(\)](#), [Exception.ToString\(\)](#),  
[Exception.GetObjectData\(SerializationInfo, StreamingContext\)](#), [Exception.GetType\(\)](#),  
[Exception.Message](#), [Exception.Data](#), [Exception.InnerException](#), [Exception.TargetSite](#),  
[Exception.StackTrace](#), [Exception.HelpLink](#), [Exception.Source](#), [Exception.HResult](#),  
[Exception.SerializeObjectState](#), [object.Equals\(object\)](#), [object.Equals\(object, object\)](#),  
[object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.MemberwiseClone\(\)](#)

# Constructors

## MbException(MbError)

```
public MbException(MbError mbError)
```

## Parameters

mbError [MbError](#)

# Properties

## Description

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

Property Value

[string](#) ↗

## MbError

```
public MbError MbError { get; set; }
```

Property Value

[MbError](#)

# Namespace MB.Common.Parameters

## Classes

### [RevitParameterManager](#)

Manages shared parameters in a Revit document, ensuring required parameters exist and are properly configured.

# Class RevitParameterManager

Namespace: [MB.Common.Parameters](#)

Assembly: MB.Common.dll

Manages shared parameters in a Revit document, ensuring required parameters exist and are properly configured.

```
public class RevitParameterManager
```

## Inheritance

[object](#) ← RevitParameterManager

## Inherited Members

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

## Constructors

### RevitParameterManager()

```
public RevitParameterManager()
```

## Methods

### EnsureParameters(Document, ToolsSharedParameter[])

```
public static void EnsureParameters(Document doc, ToolsSharedParameter[] parameters)
```

## Parameters

**doc** Document

**parameters** [ToolsSharedParameter](#)[]

## EnsureParameters(Document, MbffTool)

Ensures that all required shared parameters for a specific tool exist in the Revit document.

```
public static void EnsureParameters(Document doc, MbffTool tool)
```

### Parameters

**doc** Document

The Revit document.

**tool** [MbffTool](#)

The tool requiring the shared parameters.

# Namespace MB.Common.Services

## Classes

[MailService](#)

# Class MailService

Namespace: [MB.Common.Services](#)

Assembly: MB.Common.dll

```
public static class MailService
```

## Inheritance

[object](#) ← MailService

## Inherited Members

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

## Methods

### Send(string, List<string>)

```
[LogAspect]  
public static void Send(string msg, List<string> attachmentPaths = null)
```

#### Parameters

msg [string](#)

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

### SendAsync(string, List<string>)

```
[LogAspect]  
public static Task SendAsync(string msg, List<string> attachmentPaths = null)
```

#### Parameters

msg [string](#)

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

Returns

[Task](#)

# Namespace MB.Common.SharedParameters

## Classes

### [ElementExtensions](#)

Provides extension methods for working with Revit Element objects.

### [SharedParameterExtensions](#)

Provides extension methods for the [ToolsSharedParameter](#) class.

## Enums

### [SharedParameterType](#)

Represents the types of shared parameters that can be used in Revit.

### [ToolsSharedParameter](#)

Defines shared parameters for use within Revit, including metadata such as GUIDs, descriptions, parameter types, and applicable categories.

# Class ElementExtensions

Namespace: [MB.Common.SharedParameters](#)

Assembly: MB.Common.dll

Provides extension methods for working with Revit Element objects.

```
public static class ElementExtensions
```

## Inheritance

[object](#) ← ElementExtensions

## Inherited Members

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

# Methods

## MbGetParameter(Element, ToolsSharedParameter)

Retrieves a Revit Autodesk.Revit.DB.Parameter associated with the specified shared parameter from the given Element.

```
public static Parameter MbGetParameter(this Element element, ToolsSharedParameter  
sharedParameter)
```

### Parameters

**element** Element

The Revit Element to retrieve the parameter from.

**sharedParameter** [ToolsSharedParameter](#)

The [ToolsSharedParameter](#) to retrieve.

### Returns

Parameter

The Autodesk.Revit.DB.Parameter object associated with the specified shared parameter. Returns `null` if the parameter is not found.

## MbSetParameter<T>(Element, ToolsSharedParameter, T)

Sets the value of a Revit Autodesk.Revit.DB.Parameter associated with a specified shared parameter.

```
public static void MbSetParameter<T>(this Element element, ToolsSharedParameter  
sharedParameter, T value)
```

### Parameters

**element** Element

The Revit Element to set the parameter for.

**sharedParameter** [ToolsSharedParameter](#)

The [ToolsSharedParameter](#) whose value is to be set.

**value** T

The value to assign to the parameter.

### Type Parameters

T

The type of the value to set (e.g., int, double, string, ElementId).

### Exceptions

**ArgumentException**

Thrown if the parameter value type is unsupported or the parameter cannot be set with the specified type.

# Class SharedParameterExtensions

Namespace: [MB.Common.SharedParameters](#)

Assembly: MB.Common.dll

Provides extension methods for the [ToolsSharedParameter](#) class.

```
public static class SharedParameterExtensions
```

## Inheritance

[object](#) ← SharedParameterExtensions

## Inherited Members

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

## Methods

### GetCreationOptions(ToolsSharedParameter)

Creates an instance of ExternalDefinitionCreationOptions for a [ToolsSharedParameter](#).

```
public static ExternalDefinitionCreationOptions GetCreationOptions(this  
ToolsSharedParameter sharedParameter)
```

#### Parameters

sharedParameter [ToolsSharedParameter](#)

The shared parameter.

#### Returns

ExternalDefinitionCreationOptions

An ExternalDefinitionCreationOptions instance.

## GetDisplayName(ToolsSharedParameter)

Retrieves the display name of a [ToolsSharedParameter](#).

```
public static string GetDisplayName(this ToolsSharedParameter sharedParameter)
```

Parameters

**sharedParameter** [ToolsSharedParameter](#)

The shared parameter.

Returns

[string](#)

The display name of the shared parameter.

## GetGuid(ToolsSharedParameter)

Retrieves the GUID associated with a [ToolsSharedParameter](#).

```
public static string GetGuid(this ToolsSharedParameter sharedParameter)
```

Parameters

**sharedParameter** [ToolsSharedParameter](#)

The shared parameter.

Returns

[string](#)

The GUID of the shared parameter.

## GetIncludedCategories(ToolsSharedParameter)

Retrieves the list of categories included in a [ToolsSharedParameter](#).

```
public static List<BuiltInCategory> GetIncludedCategories(this  
ToolsSharedParameter sharedParameter)
```

## Parameters

**sharedParameter** [ToolsSharedParameter](#)

The shared parameter.

## Returns

[List](#) <BuiltInCategory>

A list of BuiltInCategory values.

## GetSharedParameterType(ToolsSharedParameter)

Retrieves the type of [ToolsSharedParameter](#).

```
public static SharedParameterType GetSharedParameterType(this ToolsSharedParameter  
sharedParameter)
```

## Parameters

**sharedParameter** [ToolsSharedParameter](#)

The shared parameter.

## Returns

[SharedParameterType](#)

The [SharedParameterType](#) of the shared parameter.

# Enum SharedParameterType

Namespace: [MB.Common.SharedParameters](#)

Assembly: MB.Common.dll

Represents the types of shared parameters that can be used in Revit.

```
public enum SharedParameterType
```

## Fields

**Area = 3**

Represents an area parameter type, used for values that define area measurements.

**Integer = 2**

Represents an integer parameter type, used for whole number values.

**Text = 1**

Represents a text parameter type, used for string values.

**YesOrNo = 0**

Represents a Yes/No parameter type, typically used for boolean values.

# Enum ToolsSharedParameter

Namespace: [MB.Common.SharedParameters](#)

Assembly: MB.Common.dll

Defines shared parameters for use within Revit, including metadata such as GUIDs, descriptions, parameter types, and applicable categories.

```
public enum ToolsSharedParameter
```

## Extension Methods

[SharedParameterExtensions.GetCreationOptions\(ToolsSharedParameter\)](#) ,  
[SharedParameterExtensions.GetDisplayName\(ToolsSharedParameter\)](#) ,  
[SharedParameterExtensions.GetGuid\(ToolsSharedParameter\)](#) ,  
[SharedParameterExtensions.GetIncludedCategories\(ToolsSharedParameter\)](#) ,  
[SharedParameterExtensions.GetSharedParameterType\(ToolsSharedParameter\)](#).

## Fields

[Guid("9fdf6b2a-2772-4194-969c-f46d6e6ab866")]  
[SharedParameterType(SharedParameterType.Area)] [RevitCategories] ActualCoverage = 1

Represents the "MBF\_ActualCoverage" shared parameter. This shared parameter is applicable to pipe curves, pipe fittings, sprinklers, and pipe accessories. It is of type [Area](#).

[Guid("282d1a77-f651-4479-b6eb-85d84361c615")]  
[SharedParameterType(SharedParameterType.Text)] [RevitCategories] AddFitToId = 10

Represents the "MBF\_AddFitToID" shared parameter. This shared parameter is applicable to pipe curves, pipe fittings, pipe accessories, generic models, model groups, mechanical equipment, sprinklers, fire alarm devices, and flex pipe curves. It is of type [Text](#).

[Guid("40221BB0-FFFE-42C6-BFF1-F1048AFA0664")]  
[SharedParameterType(SharedParameterType.Integer)] [RevitCategories] Allowance = 18

Represents the "MBF\_Allowance" shared parameter. This shared parameter is applicable to flex pipe curves. It is of type [Integer](#).

[Guid("6BEC8CDC-378E-4389-A7D6-EAF2493CEB9A")]  
[SharedParameterType(SharedParameterType.YesNo)] [RevitCategories] CalculateSum =

Represents the "MBF\_IsFitting" shared parameter.This shared parameter is used for flex pipe curves and is of type [YesOrNo](#).

```
[Guid("14fca12a-1224-41e5-a8a6-d4c0af178974")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories] Color = 7
```

Represents the "MBF\_Color" shared parameter.This shared parameter is applicable to pipe curves, pipe fittings, pipe accessories, generic models, model groups, mechanical equipment, sprinklers, fire alarm devices, and flex pipe curves. It is of type [Text](#).

```
[Guid("27e7c7ba-40cc-4ca6-b0c3-07a8e537a3bb")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories]
DescriptionAndAbbreviationListNo = 12
```

Represents the "MBF\_Desc\_Abbrv\_ListNo" shared parameter.This shared parameter is applicable to pipe curves, pipe fittings, pipe accessories, generic models, model groups, mechanical equipment, sprinklers, fire alarm devices, and flex pipe curves. It is of type [Text](#).

```
[Guid("d94545bc-4956-411d-a6bc-9ec96101d350")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories]
ExportedPackageName = 13
```

Represents the "MBF\_Exported\_PakageName" shared parameter.This shared parameter is applicable to pipe curves, pipe fittings, pipe accessories, generic models, model groups, mechanical equipment, sprinklers, fire alarm devices, and flex pipe curves. It is of type [Text](#).

```
[Guid("83d4381a-bfe9-450e-9364-c44036e81756")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories] ExportedPdfName = 15
```

Represents the "MBF\_Exported\_PDFName" shared parameter.This shared parameter is applicable to pipe curves, pipe fittings, pipe accessories, generic models, model groups, mechanical equipment, sprinklers, fire alarm devices, and flex pipe curves. It is of type [Text](#).

```
[Guid("1bf7feba-7bca-40c9-8ca0-8365574c2648")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories]
FabricationGroupName = 17
```

Represents the "MBF\_FabGroupName" shared parameter.This shared parameter is applicable to pipe curves, pipe fittings, pipe accessories, generic models, model groups, mechanical equipment, sprinklers, fire alarm devices, and flex pipe curves. It is of type [Text](#).

```
[Guid("15b2c379-8df6-4f2d-8499-9883e0256663")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories] Finish = 8
```

Represents the "MBF\_Finish" shared parameter.This shared parameter is applicable to pipe curves, pipe fittings, pipe accessories, generic models, model groups, mechanical equipment, sprinklers, fire alarm devices, and flex pipe curves. It is of type [Text](#).

```
[Guid("72f35e3d-3bd7-4169-b3c0-c6485cdd39c4")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories] HandTight = 11
```

Represents the "MBF\_HandTight" shared parameter.This shared parameter is applicable to pipe curves, pipe fittings, pipe accessories, generic models, model groups, mechanical equipment, sprinklers, fire alarm devices, and flex pipe curves. It is of type [Text](#).

```
[Guid("8495409d-c110-4685-b548-2f1440fa7307")]
[SharedParameterType(SharedParameterType.YesNo)] [RevitCategories]
IgnoreInCalculation = 3
```

Represents the "MBF\_IgnoreInCalculation" shared parameter.This shared parameter is applicable to pipe fittings, sprinklers, pipe accessories, mechanical equipment, and fire alarm devices. It is of type [YesOrNo](#).

```
[Guid("e87615ee-c2b0-4f93-9194-8ffda612cdf6")]
[SharedParameterType(SharedParameterType.YesNo)] [RevitCategories]
IncludeInCalculation = 2
```

Represents the "MBF\_IncludeInCalcs" shared parameter.This shared parameter is applicable to pipe fittings, sprinklers, pipe accessories, mechanical equipment, and fire alarm devices. It is of type [YesOrNo](#).

```
[Guid("997C72F0-F375-439E-9AD1-BC2570651E42")]
[SharedParameterType(SharedParameterType.YesNo)] [RevitCategories] IsFitting = 20
```

Represents the "MBF\_IsFitting" shared parameter.This shared parameter is used for flex pipe curves and is of type [YesOrNo](#).

```
[Guid("b06ebb90-4f1e-4881-bfb5-fef6f7a64909")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories] MinimumFlow = 5
```

Represents the "MBF\_Min\_Flow" shared parameter.This shared parameter is applicable to pipe fittings, sprinklers, and pipe accessories. It is of type [Text](#).

```
[Guid("467a1334-8e46-4d4d-bf22-0bd377ff6be5")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories] MinimumPressure =
```

Represents the "MBF\_Min\_Pressure" shared parameter. This shared parameter is applicable to pipe fittings, sprinklers, and pipe accessories. It is of type [Text](#).

```
[Guid("c4347a82-7375-42aa-ad1a-96ba6212f122")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories] PipeCode = 9
```

Represents the "MBF\_PipeCode" shared parameter. This shared parameter is applicable to pipe curves, pipe fittings, pipe accessories, generic models, model groups, mechanical equipment, sprinklers, fire alarm devices, and flex pipe curves. It is of type [Text](#).

```
[Guid("2e83bba8-8d6e-4a90-a29f-108c705a7661")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories] ProtectionArea = 6
```

Represents the "MBF\_ProtectionArea" shared parameter. This shared parameter is applicable to pipe fittings, sprinklers, and pipe accessories. It is of type [Text](#).

```
[Guid("5f2c6451-0d24-4421-b3f8-02386c2cd60b")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories] RealGroupSize = 19
```

Represents the "MBF\_Allowance" shared parameter. This shared parameter is applicable to flex pipe curves. It is of type [Text](#).

```
[Guid("9d698cc7-3c6d-466e-9f89-ee1baf1326a2")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories] RealFittingSizes = 14
```

Represents the "MBF\_RealFittingSizes" shared parameter. This shared parameter is applicable to pipe curves, pipe fittings, pipe accessories, generic models, model groups, mechanical equipment, sprinklers, fire alarm devices, and flex pipe curves. It is of type [Text](#).

```
[Guid("9d698cc7-3c6d-466e-9f89-ee1baf1326a2")]
[SharedParameterType(SharedParameterType.Text)] [RevitCategories] SubType = 16
```

Represents the "MBF\_Subtype" shared parameter. This shared parameter is applicable to pipe curves, pipe fittings, pipe accessories, generic models, model groups, mechanical equipment, sprinklers, fire alarm devices, and flex pipe curves. It is of type [Text](#).

# Namespace MB.Common.Tools

## Classes

### [MbftoolExtensions](#)

Provides extension methods for the [Mbftool](#) class.

## Enums

### [Mbftool](#)

Enum representing various tools in the MBF suite.

# Enum Mbftool

Namespace: [MB.Common.Tools](#)

Assembly: MB.Common.dll

Enum representing various tools in the MBF suite.

```
public enum Mbftool
```

## Extension Methods

[MbftoolExtensions.GetRequiredSharedParameters\(Mbftool\)](#).

## Fields

ApplyParameters = 4

Represents the Apply Parameters tool.

AssemblyManager = 2

Represents the Assembly Manager tool.

CutLengthPipeOptimization = 3

Represents the Cut Length Pipe Optimization tool.

```
[RequiredSharedParameters(new ToolsSharedParameter[] { ToolsSharedParameter.Color,
ToolsSharedParameter.Finish, ToolsSharedParameter.PipeCode,
ToolsSharedParameter.AddFitToId, ToolsSharedParameter.HandTight,
ToolsSharedParameter.DescriptionAndAbbreviationListNo,
ToolsSharedParameter.ExportedPackageName, ToolsSharedParameter.RealFittingSizes,
ToolsSharedParameter.ExportedPdfName, ToolsSharedParameter.SubType,
ToolsSharedParameter.FabricationGroupName, ToolsSharedParameter.Allowance,
ToolsSharedParameter.RealGroupSize, ToolsSharedParameter.IsFitting,
ToolsSharedParameter.CalculateSum })] Mapper = 5
```

```
[RequiredSharedParameters(new ToolsSharedParameter[] {
ToolsSharedParameter.ActualCoverage, ToolsSharedParameter.ProtectionArea })]
SprinklerVoronoi = 1
```

Represents the Sprinkler Voronoi tool. This tool requires the [ActualCoverage](#) shared parameter.

## Remarks

Each tool is associated with a description and optionally required shared parameters. Use attributes like [DescriptionAttribute](#) and [RequiredSharedParametersAttribute](#) to provide metadata for each tool.

# Class MbftoolExtensions

Namespace: [MB.Common.Tools](#)

Assembly: MB.Common.dll

Provides extension methods for the [Mbftool](#) class.

```
public static class MbftoolExtensions
```

## Inheritance

[object](#) ← MbftoolExtensions

## Inherited Members

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

## Methods

### GetRequiredSharedParameters(Mbftool)

Retrieves the required shared parameters associated with the specified [Mbftool](#).

```
public static ToolsSharedParameter[] GetRequiredSharedParameters(this Mbftool tool)
```

#### Parameters

tool [Mbftool](#)

The [Mbftool](#) instance for which the required shared parameters are retrieved.

#### Returns

[ToolsSharedParameter\[\]](#)

An array of [ToolsSharedParameter](#) objects if the [RequiredSharedParametersAttribute](#) is found; otherwise, an empty array.

# Namespace MB.Common.Utils.Attributes

## Classes

### [AttributeUtils](#)

Provides utility methods for working with custom attributes in .NET.

# Class AttributeUtils

Namespace: [MB.Common.Utils.Attributes](#)

Assembly: MB.Common.dll

Provides utility methods for working with custom attributes in .NET.

```
public static class AttributeUtils
```

## Inheritance

[object](#) ← AttributeUtils

## Inherited Members

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

# Methods

## GetAttribute<T>(object)

Retrieves a custom attribute of the specified type **T** applied to a given enum value or field.

```
public static T GetAttribute<T>(object value) where T : Attribute
```

### Parameters

**value** [object](#)

The target enum value or field.

### Returns

**T**

The attribute of type **T** if it is applied; otherwise, throws an exception.

### Type Parameters

**T**

The type of the attribute to retrieve.

## GetFieldAttribute<T, TType>(string)

Retrieves a custom attribute of the specified type **T** applied to a given field in a class or struct.

```
public static T GetFieldAttribute<T, TType>(string fieldName) where T : Attribute
```

### Parameters

**fieldName** [string](#)

The name of the field in the class or struct.

### Returns

**T**

The attribute of type **T** if it is applied; otherwise, throws an exception.

### Type Parameters

**T**

The type of the attribute to retrieve.

**TType**

The type of the target class or struct.

## TryGetAttribute<T>(object, out T)

Attempts to retrieve a custom attribute of the specified type **T** applied to a given enum value or field without throwing an exception if it is not found.

```
public static bool TryGetAttribute<T>(object value, out T attribute) where T : Attribute
```

### Parameters

**value** [object](#)

The target enum value or field.

#### **attribute T**

When this method returns, contains the attribute of type **T** if found; otherwise, **null**.

Returns

#### [bool ↗](#)

True if the attribute is found; otherwise, false.

Type Parameters

#### **T**

The type of the attribute to retrieve.

# Namespace MB.Common.Utils.Handlers

## Classes

[MbHandlers](#)

# Class MbHandlers

Namespace: [MB.Common.Utils.Handlers](#)

Assembly: MB.Common.dll

```
public static class MbHandlers
```

## Inheritance

[object](#) ← MbHandlers

## Inherited Members

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

# Properties

## AsyncExternalEventHandler

```
public static AsyncEventHandler AsyncExternalEventHandler { get; set; }
```

### Property Value

AsyncEventHandler

## ExternalEventHandler

```
public static ActionEventHandler ExternalEventHandler { get; set; }
```

### Property Value

ActionEventHandler

# Methods

## RegisterHandlers()

```
public static void RegisterHandlers()
```

# Namespace MB.Common.Utils.Image

## Classes

[ImageUtils](#)

# Class ImageUtils

Namespace: [MB.Common.Utils.Image](#)

Assembly: MB.Common.dll

```
public static class ImageUtils
```

## Inheritance

[object](#) ← ImageUtils

## Inherited Members

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

# Methods

## ResizeImage(Image, int, int)

Resizes an image to the specified dimensions while maintaining quality.

```
public static Bitmap ResizeImage(Image image, int width, int height)
```

### Parameters

**image** [Image](#)

The image to resize.

**width** [int](#)

The target width of the resized image.

**height** [int](#)

The target height of the resized image.

### Returns

[Bitmap](#)

The resized [Bitmap](#).

# Namespace MB.Common.Utils.Result

## Classes

### [MbResult<T>](#)

Represents the result of an operation, encapsulating the success status, value, and error message.

# Class MbResult<T>

Namespace: [MB.Common.Utils.Result](#)

Assembly: MB.Common.dll

Represents the result of an operation, encapsulating the success status, value, and error message.

```
public class MbResult<T>
```

## Type Parameters

T

The type of the result value.

## Inheritance

[object](#) ← MbResult<T>

## Inherited Members

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

# Properties

## ErrorMessage

Gets the error message if the operation failed.

```
public string ErrorMessage { get; }
```

## Property Value

[string](#)

## IsSuccess

Gets a value indicating whether the operation was successful.

```
public bool IsSuccess { get; }
```

Property Value

[bool](#)

Value

Gets the value of the result if the operation was successful.

```
public T Value { get; }
```

Property Value

T

## Methods

### Failure(string)

Creates a failure result containing the specified error message.

```
public static MbResult<T> Failure(string errorMessage)
```

Parameters

errorMessage [string](#)

The error message describing the failure.

Returns

[MbResult](#) <T>

A failed [MbResult](#) instance containing the specified error message.

### Success(T)

Creates a successful result containing the specified value.

```
public static MbResult<T> Success(T value)
```

## Parameters

**value** T

The value of the result.

## Returns

[MbResult](#)<T>

A successful [MbResult<T>](#) instance containing the specified value.

# Namespace MB.Common.ViewLoader

## Classes

[IsolatedViewLoader](#)

# Class IsolatedViewLoader

Namespace: [MB.Common.ViewLoader](#)

Assembly: MB.Common.dll

```
public static class IsolatedViewLoader
```

## Inheritance

[object](#) ← IsolatedViewLoader

## Inherited Members

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

## Methods

### LoadViewFromUri(FrameworkElement, string)

```
public static void LoadViewFromUri(this FrameworkElement view, string baseUri)
```

#### Parameters

view [FrameworkElement](#)

baseUri [string](#)

### LoadView<T>(T)

```
public static void LoadView<T>(this T view) where T : FrameworkElement
```

#### Parameters

view T

## Type Parameters

T

# Namespace MBF.Revit.Data.SelectionFilters

## Classes

[DynamicCategorySelectionFilter](#)

# Class DynamicCategorySelectionFilter

Namespace: [MBF.Revit.Data.SelectionFilters](#)

Assembly: MBF.Revit.Data.dll

```
public class DynamicCategorySelectionFilter : ISelectionFilter
```

## Inheritance

[object](#) ← DynamicCategorySelectionFilter

## Implements

ISelectionFilter

## Inherited Members

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

## Constructors

DynamicCategorySelectionFilter(params BuiltInCategory[])

```
public DynamicCategorySelectionFilter(params BuiltInCategory[] categories)
```

## Parameters

categories BuiltInCategory[]

## Methods

AllowElement(Element)

```
public bool AllowElement(Element element)
```

## Parameters

**element** Element

Returns

bool ↗

## AllowReference(Reference, XYZ)

```
public bool AllowReference(Reference reference, XYZ position)
```

Parameters

**reference** Reference

**position** XYZ

Returns

bool ↗

# Namespace MBF.Revit.Data.Utils

## Classes

[MbFlexPipesUtils](#)

[MbParameterUtils](#)

[MbPipeAccessoriesUtils](#)

[MbPipeFittingsUtils](#)

Provides utility methods for working with pipe fittings in Revit.

[MbPipesUtils](#)

[MbSprinklersUtils](#)

[MbUnitUtils](#)

[MbViewUtils](#)

# Class MbFlexPipesUtils

Namespace: [MBF.Revit.Data.Utils](#)

Assembly: MBF.Revit.Data.dll

```
public static class MbFlexPipesUtils
```

## Inheritance

[object](#) ← MbFlexPipesUtils

## Inherited Members

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

## Methods

### GetFlexPipeInstances(IEnumerable<Element>)

Filters the provided collection of Revit elements and retrieves all flex pipes.

```
public static IEnumerable<FlexPipe> GetFlexPipeInstances(IEnumerable<Element>  
selectedElements)
```

#### Parameters

**selectedElements** [IEnumerable](#)<Element>

A collection of Revit elements to filter.

#### Returns

[IEnumerable](#)<FlexPipe>

An [IEnumerable](#)<T> of Autodesk.Revit.DB.Plumbing.FlexPipe objects.

#### Remarks

This method ensures that only elements categorized as flex pipes (BuiltInCategory.OST\_FlexPipeCurves) are returned.

## GetFlexPipeLength(FlexPipe)

```
public static double GetFlexPipeLength(FlexPipe flexPipe)
```

### Parameters

**flexPipe** FlexPipe

### Returns

[double](#)

# Class MbParameterUtils

Namespace: [MBF.Revit.Data.Utils](#)

Assembly: MBF.Revit.Data.dll

```
public static class MbParameterUtils
```

## Inheritance

[object](#) ← MbParameterUtils

## Inherited Members

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

## Methods

### GetParameterValue(StorageType, Parameter)

```
public static string GetParameterValue(StorageType storageType, Parameter instanceParameter)
```

#### Parameters

**storageType** StorageType

**instanceParameter** Parameter

#### Returns

[string](#)

# Class MbPipeAccessoriesUtils

Namespace: [MBF.Revit.Data.Utils](#)

Assembly: MBF.Revit.Data.dll

```
public static class MbPipeAccessoriesUtils
```

## Inheritance

[object](#) ← MbPipeAccessoriesUtils

## Inherited Members

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

## Methods

### GetPipeAccessoriesInstances(IEnumerable<Element>)

Filters the provided collection of Revit elements and retrieves all family instances that are categorized as "Pipe Accessories".

```
public static IEnumerable<FamilyInstance> GetPipeAccessoriesInstances(IEnumerable<Element>  
selectedElements)
```

#### Parameters

**selectedElements** [IEnumerable](#)<Element>

A collection of Revit elements to filter.

#### Returns

[IEnumerable](#)<FamilyInstance>

An [IEnumerable](#)<T> of FamilyInstance objects representing pipe accessories.

#### Remarks

This method ensures that only elements categorized as pipe accessories (BuiltInCategory.OST\_PipeAccessory) are returned.

# Class MbPipeFittingsUtils

Namespace: [MBF.Revit.Data.Utils](#)

Assembly: MBF.Revit.Data.dll

Provides utility methods for working with pipe fittings in Revit.

```
public static class MbPipeFittingsUtils
```

## Inheritance

[object](#) ← MbPipeFittingsUtils

## Inherited Members

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

## Methods

### FittingSizeIsEqual(FamilyInstance)

Checks if all connectors of a given fitting have equal sizes (radius).

```
public static MbResult<bool> FittingSizeIsEqual(FamilyInstance instance)
```

#### Parameters

**instance** FamilyInstance

The FamilyInstance of the fitting to check.

#### Returns

[MbResult](#)<[bool](#)>

Returns an [MbResult<T>](#) where T is [bool](#):

- If [IsSuccess](#) is [true](#), the operation completed successfully.
- If [Value](#) is [true](#), all connectors have equal sizes or there are fewer than two connectors.
- If [Value](#) is [false](#), connectors have different sizes.
- If [IsSuccess](#) is [false](#), an error occurred, and [ErrorMessage](#) provides details.

## GetAngleBetweenLargestRadiusConnectors(FamilyInstance)

Calculates the angle between the two connectors with the largest radius in the given FamilyInstance.

```
public static MbResult<double> GetAngleBetweenLargestRadiusConnectors(FamilyInstance instance)
```

### Parameters

**instance** FamilyInstance

The FamilyInstance containing connectors to analyze.

### Returns

[MbResult](#) <[double](#)>

Returns an [MbResult<T>](#) where T is [double](#):

- If [IsSuccess](#) is [true](#), the operation completed successfully.
- If [Value](#) is a valid angle, it represents the computed angle between the two connectors with the largest radius.
- If [IsSuccess](#) is [false](#), an error occurred, and [ErrorMessage](#) provides details.

## GetAngleBetweenLargestRadiusConnectors(List<Connector>)

Calculates the angle between the two connectors with the largest radius in the given list of connectors.

```
public static MbResult<double> GetAngleBetweenLargestRadiusConnectors(List<Connector> connectors)
```

### Parameters

**connectors** [List](#)<Connector>

A list of connectors to analyze.

### Returns

[MbResult](#) <[double](#)>

Returns an [MbResult<T>](#) where T is [double](#):

- If [IsSuccess](#) is [true](#), the operation completed successfully.

- If `Value` is a valid angle, it represents the computed angle between the two connectors with the largest radius.
- If `IsSuccess` is `false`, an error occurred, and `ErrorMessage` provides details.

## GetElbowsFamilyInstances(IEnumerable<Element>)

Filters the provided collection of Revit elements and retrieves all pipe fitting family instances that are categorized as "Elbow" fittings.

```
public static IEnumerable<FamilyInstance> GetElbowsFamilyInstances(IEnumerable<Element>
selectedElements)
```

### Parameters

`selectedElements` [IEnumerable](#)<Element>

A collection of Revit elements to filter.

### Returns

[IEnumerable](#)<FamilyInstance>

An [IEnumerable](#)<T> of FamilyInstance objects representing elbow fittings.

### Remarks

This method ensures that only elements categorized as pipe fittings (`BuiltInCategory.OST_PipeFitting`) and specifically of type `PartType.Elbow` are returned.

## GetOrderedConnectorSizesInInch(Element)

Retrieves a sorted list of connector sizes (diameters) in inches for a given Element that is an Autodesk.Revit.DB.MEPCurve.

```
public static MbResult<List<double>> GetOrderedConnectorSizesInInch(Element element)
```

### Parameters

`element` Element

The Element representing an Autodesk.Revit.DB.MEPCurve.

Returns

[MbResult](#) <List <double>>>

Returns an [MbResult<T>](#) where T is List<double>:

- If `IsSuccess` is `true`, the operation completed successfully.
- The list contains the diameters of all connectors in inches, sorted in ascending order.
- If `IsSuccess` is `false`, an error occurred, and `ErrorMessage` provides details.

## GetOrderedConnectorSizesInInch(FamilyInstance)

```
public static MbResult<List<double>> GetOrderedConnectorSizesInInch(FamilyInstance instance)
```

Parameters

`instance` FamilyInstance

Returns

[MbResult](#) <List <double>>>

## GetOtherFittingWhichIsNotElbowsOrTeesOrOutletsFamilyInstances(IEnumerable<Element>)

Filters the provided collection of Revit elements and retrieves all pipe fitting family instances that are not categorized as "Tee", "Elbow", or "Outlets" fittings.

```
public static IEnumerable<FamilyInstance>
GetOtherFittingWhichIsNotElbowsOrTeesOrOutletsFamilyInstances(IEnumerable<Element>
selectedElements)
```

Parameters

`selectedElements` [IEnumerable](#) <Element>

A collection of Revit elements to filter.

Returns

## [IEnumerable](#) <FamilyInstance>

An [IEnumerable<T>](#) of FamilyInstance objects representing fittings that are not tee, elbow, or outlet fittings.

### Remarks

This method ensures that only elements categorized as pipe fittings (BuiltInCategory.OST\_PipeFitting ) and excluding types PartType.Tee, PartType.Elbow, or PartType.SpudAdjustable are returned.

## GetOutletsFamilyInstances(IEnumerable<Element>)

Filters the provided collection of Revit elements and retrieves all pipe fitting family instances that are categorized as "Outlets" (Spud Adjustable) fittings.

```
public static IEnumerable<FamilyInstance> GetOutletsFamilyInstances(IEnumerable<Element> selectedElements)
```

### Parameters

#### `selectedElements` [IEnumerable](#) <Element>

A collection of Revit elements to filter.

### Returns

## [IEnumerable](#) <FamilyInstance>

An [IEnumerable<T>](#) of FamilyInstance objects representing outlet fittings.

### Remarks

This method ensures that only elements categorized as pipe fittings (BuiltInCategory.OST\_PipeFitting ) and specifically of type PartType.SpudAdjustable (representing outlets) are returned.

## GetTeeFamilyInstances(IEnumerable<Element>)

Filters the provided collection of Revit elements and retrieves all pipe fitting family instances that are categorized as "Tee" fittings.

```
public static IEnumerable<FamilyInstance> GetTeeFamilyInstances(IEnumerable<Element>
```

```
selectedElements)
```

## Parameters

**selectedElements** [IEnumerable](#)<Element>

A collection of Revit elements to filter.

## Returns

[IEnumerable](#)<FamilyInstance>

An [IEnumerable](#)<T> of FamilyInstance objects representing tee fittings.

## Remarks

This method ensures that only elements categorized as pipe fittings (BuiltInCategory.OST\_PipeFitting ) and specifically of type PartType.Tee are returned.

# Class MbPipesUtils

Namespace: [MBF.Revit.Data.Utils](#)

Assembly: MBF.Revit.Data.dll

```
public static class MbPipesUtils
```

## Inheritance

[object](#) ← MbPipesUtils

## Inherited Members

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

## Methods

### GetPipeInstances(IEnumerable<Element>)

Filters the provided collection of Revit elements and retrieves all rigid pipes.

```
public static IEnumerable<Pipe> GetPipeInstances(IEnumerable<Element> selectedElements)
```

#### Parameters

**selectedElements** [IEnumerable](#)<Element>

A collection of Revit elements to filter.

#### Returns

[IEnumerable](#)<Pipe>

An [IEnumerable](#)<T> of Autodesk.Revit.DB.Plumbing.Pipe objects.

#### Remarks

This method ensures that only elements categorized as pipes (BuiltInCategory.OST\_PipeCurves) are returned.

# Class MbSprinklersUtils

Namespace: [MBF.Revit.Data.Utils](#)

Assembly: MBF.Revit.Data.dll

```
public static class MbSprinklersUtils
```

## Inheritance

[object](#) ← MbSprinklersUtils

## Inherited Members

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

## Methods

### GetSprinklerInstances(IEnumerable<Element>)

Filters the provided collection of Revit elements and retrieves all family instances that are categorized as "Sprinklers".

```
public static IEnumerable<FamilyInstance> GetSprinklerInstances(IEnumerable<Element>  
selectedElements)
```

#### Parameters

**selectedElements** [IEnumerable](#)<Element>

A collection of Revit elements to filter.

#### Returns

[IEnumerable](#)<FamilyInstance>

An [IEnumerable](#)<T> of FamilyInstance objects representing sprinklers.

#### Remarks

This method ensures that only elements categorized as sprinklers (BuiltInCategory.OST\_Sprinklers) are returned.

# Class MbUnitUtils

Namespace: [MBF.Revit.Data.Utils](#)

Assembly: MBF.Revit.Data.dll

```
public static class MbUnitUtils
```

## Inheritance

[object](#) ← MbUnitUtils

## Inherited Members

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

## Methods

### GetLengthUnitSymbol()

```
public static string GetLengthUnitSymbol()
```

#### Returns

[string](#)

### GetStandardUnitSymbol(object)

Maps Revit unit types to standardized short symbols

```
public static string GetStandardUnitSymbol(object unitTypeId)
```

#### Parameters

[unitTypeId](#) [object](#)

#### Returns

[string](#) ↗

# Class MbViewUtils

Namespace: [MBF.Revit.Data.Utils](#)

Assembly: MBF.Revit.Data.dll

```
public class MbViewUtils
```

## Inheritance

[object](#) ← MbViewUtils

## Inherited Members

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

## Constructors

### MbViewUtils()

```
public MbViewUtils()
```

## Methods

### IsGraphicalView(View)

```
public static bool IsGraphicalView(View view)
```

#### Parameters

**view** View

#### Returns

[bool](#)

## ZoomToElement(Element)

```
public static void ZoomToElement(Element element)
```

### Parameters

**element** Element

## ZoomToElement(FamilyInstance)

```
public static void ZoomToElement(FamilyInstance instnace)
```

### Parameters

**instnace** FamilyInstance

# Namespace MBFWpfToolkit

## Classes

### [MbWindow](#)

Represents a custom window with additional properties and functionalities.

# Class MbWindow

Namespace: [MBFWpfToolkit](#)

Assembly: MBFWpfToolkit.dll

Represents a custom window with additional properties and functionalities.

```
public class MbWindow : Window
```

## Inheritance

[object](#) ← MbWindow

## Derived

[HelpWizard](#), [MbMessageBox](#)

## Constructors

### MbWindow()

Initializes a new instance of the [MbWindow](#) class. Subscribes to [Loaded](#) and Window.Closed events to handle resource initialization and cleanup.

```
public MbWindow()
```

## Fields

### IsThemeToggleVisibleProperty

Identifies the [IsThemeToggleVisible](#) dependency property.

```
public static readonly DependencyProperty IsThemeToggleVisibleProperty
```

## Field Value

[DependencyProperty](#)

## PluginDescriptionProperty

Identifies the PluginDescription dependency property.

```
public static readonly DependencyProperty PluginDescriptionProperty
```

Field Value

[DependencyProperty](#)

## PluginNameProperty

Identifies the PluginName dependency property.

```
public static readonly DependencyProperty PluginNameProperty
```

Field Value

[DependencyProperty](#)

## PluginVersionProperty

Identifies the PluginVersion dependency property.

```
public static readonly DependencyProperty PluginVersionProperty
```

Field Value

[DependencyProperty](#)

## PrimaryBrushProperty

Identifies the PrimaryBrush dependency property.

```
public static readonly DependencyProperty PrimaryBrushProperty
```

Field Value

[DependencyProperty](#)

## ShowLanguagesComboBoxProperty

Identifies the ShowLanguagesComboBox dependency property.

```
public static readonly DependencyProperty ShowLanguagesComboBoxProperty
```

Field Value

[DependencyProperty](#)

## ShowLogoProperty

Identifies the [ShowLogo](#) dependency property.

```
public static readonly DependencyProperty ShowLogoProperty
```

Field Value

[DependencyProperty](#)

## ShowPluginNameProperty

Identifies the [ShowPluginName](#) dependency property.

```
public static readonly DependencyProperty ShowPluginNameProperty
```

Field Value

[DependencyProperty](#)

## ShowPluginVersionProperty

Identifies the [ShowPluginVersion](#) dependency property.

```
public static readonly DependencyProperty ShowPluginVersionProperty
```

Field Value

[DependencyProperty](#)

# Properties

## IsThemeToggleVisible

Gets or sets a value indicating whether the theme toggle button (light/dark mode switch) is visible in the window's title bar.

```
public bool IsThemeToggleVisible { get; set; }
```

Property Value

[bool](#)

## MainResources

Gets or sets the main shared resource dictionary used across all [MbWindow](#) instances. This allows consistent styling and resource reuse throughout the application.

```
public static ResourceDictionary? MainResources { get; set; }
```

Property Value

[ResourceDictionary](#)

## PluginDescription

Gets or sets the description of the plugin.

```
public string PluginDescription { get; set; }
```

Property Value

[string](#)

## PluginName

Gets or sets the name of the plugin.

```
public string PluginName { get; set; }
```

Property Value

[string](#) ↗

## PluginVersion

Gets or sets the version of the plugin.

```
public string PluginVersion { get; set; }
```

Property Value

[string](#) ↗

## PrimaryBrush

Gets or sets the primary brush as a Color.

```
public Color PrimaryBrush { get; set; }
```

Property Value

[Color](#) ↗

## ShowLanguagesComboBox

Gets or sets a value indicating whether to show the ComboBox for languages.

```
public bool ShowLanguagesComboBox { get; set; }
```

Property Value

[bool](#) ↗

## ShowLogo

Gets or sets a value indicating whether the logo should be displayed in the window's header.

```
public bool ShowLogo { get; set; }
```

Property Value

[bool](#)

## ShowPluginName

Gets or sets a value indicating whether the plugin name should be displayed in the window's header.

```
public bool ShowPluginName { get; set; }
```

Property Value

[bool](#)

## ShowPluginVersion

Gets or sets a value indicating whether the plugin version should be displayed in the window's header.

```
public bool ShowPluginVersion { get; set; }
```

Property Value

[bool](#)

## Methods

### GetPrimaryBrush()

Gets the primary brush color.

```
protected static Color GetPrimaryBrush()
```

Returns

[Color](#) ↗

The primary brush color.

## InitializeWindow()

Performs static initialization for the [MbWindow](#). Applies the primary brush color, sets window properties, configures non-client UI content, and ensures shared resources are loaded into [MainResources](#).

```
protected void InitializeWindow()
```

## ShowSprite()

Displays the [MicroBimSprite](#) overlay window (e.g., a floating mascot or animation). Stores the instance internally and calls its [Show\(\)](#) method to render it on screen.

```
public void ShowSprite()
```

# Namespace MBFWpfToolkit.AttachedProperties

## Classes

### [ComboBoxSearchBehavior](#)

Provides an attached behavior to enable live search filtering on a [ComboBox](#).

### [HelpAssistant](#)

Provides attached properties to define step-based help or onboarding instructions for UI elements. Elements can be dynamically registered and retrieved in a specific order based on their step number.

### [LocalizationHelper](#)

Provides localization support for WPF controls using attached properties. When [LangKeyProperty](#) is set on a control, the corresponding localized value will be applied automatically.

# Class ComboBoxSearchBehavior

Namespace: [MBFWpfToolkit.AttachedProperties](#)

Assembly: MBFWpfToolkit.dll

Provides an attached behavior to enable live search filtering on a [ComboBox](#).

```
public static class ComboBoxSearchBehavior
```

## Inheritance

[object](#) ← ComboBoxSearchBehavior

## Inherited Members

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

## Fields

### IsSearchEnabledProperty

Identifies the IsSearchEnabled attached property. When set to true, enables live search/filtering within the ComboBox.

```
public static readonly DependencyProperty IsSearchEnabledProperty
```

## Field Value

[DependencyProperty](#)

## Methods

### GetIsSearchEnabled(DependencyObject)

Gets the value of the IsSearchEnabled attached property.

```
public static bool GetIsSearchEnabled(DependencyObject obj)
```

Parameters

**obj** [DependencyObject](#)

Returns

[bool](#)

## GetOriginalItemsSource(DependencyObject)

Gets the original unfiltered [IEnumerable](#) item source.

```
public static IEnumerable GetOriginalItemsSource(DependencyObject obj)
```

Parameters

**obj** [DependencyObject](#)

Returns

[IEnumerable](#)

## SetIsSearchEnabled(DependencyObject, bool)

Sets the value of the IsSearchEnabled attached property.

```
public static void SetIsSearchEnabled(DependencyObject obj, bool value)
```

Parameters

**obj** [DependencyObject](#)

**value** [bool](#)

# Class HelpAssistant

Namespace: [MBFWpfToolkit.AttachedProperties](#)

Assembly: MBFWpfToolkit.dll

Provides attached properties to define step-based help or onboarding instructions for UI elements. Elements can be dynamically registered and retrieved in a specific order based on their step number.

```
public class HelpAssistant
```

## Inheritance

[object](#) ← HelpAssistant

## Inherited Members

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

## Constructors

### HelpAssistant()

```
public HelpAssistant()
```

## Fields

### AutoRegisterProperty

Identifies the AutoRegister attached property. This property is used only to trigger element registration dynamically.

```
public static readonly DependencyProperty AutoRegisterProperty
```

## Field Value

[DependencyProperty](#)

## StepNumberProperty

Identifies the StepNumber attached property. Represents the sequence order of the help step.

```
public static readonly DependencyProperty StepNumberProperty
```

Field Value

[DependencyProperty](#)

## StepTextProperty

Identifies the StepText attached property. Describes the instructional text for the help step.

```
public static readonly DependencyProperty StepTextProperty
```

Field Value

[DependencyProperty](#)

## Methods

### GetAutoRegister(DependencyObject)

Gets the AutoRegister value of the element.

```
public static bool GetAutoRegister(DependencyObject obj)
```

Parameters

**obj** [DependencyObject](#)

Returns

[bool](#)

### GetOrderedSteps()

Retrieves all registered UI elements with valid step numbers, sorted in ascending order.

```
public static List<FrameworkElement> GetOrderedSteps()
```

Returns

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

A sorted list of registered [FrameworkElement](#)s for the help assistant.

## GetStepNumber(DependencyObject)

Gets the step number of the element.

```
public static int GetStepNumber(DependencyObject obj)
```

Parameters

**obj** [DependencyObject](#)

Returns

[int](#)

## GetStepText(DependencyObject)

Gets the step text of the element.

```
public static string GetStepText(DependencyObject obj)
```

Parameters

**obj** [DependencyObject](#)

Returns

[string](#)

## SetAutoRegister(DependencyObject, bool)

Sets the AutoRegister value of the element.

```
public static void SetAutoRegister(DependencyObject obj, bool value)
```

Parameters

obj [DependencyObject](#)

value [bool](#)

## SetStepNumber(DependencyObject, int)

Sets the step number of the element.

```
public static void SetStepNumber(DependencyObject obj, int value)
```

Parameters

obj [DependencyObject](#)

value [int](#)

## SetStepText(DependencyObject, string)

Sets the step text of the element.

```
public static void SetStepText(DependencyObject obj, string value)
```

Parameters

obj [DependencyObject](#)

value [string](#)

# Class LocalizationHelper

Namespace: [MBFWpfToolkit.AttachedProperties](#)

Assembly: MBFWpfToolkit.dll

Provides localization support for WPF controls using attached properties. When [LangKeyProperty](#) is set on a control, the corresponding localized value will be applied automatically.

```
public static class LocalizationHelper
```

## Inheritance

[object](#) ← LocalizationHelper

## Inherited Members

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

# Fields

## LangKeyProperty

Identifies the [LangKey](#) attached property. This property is used to bind a localization key to a control.

```
public static readonly DependencyProperty LangKeyProperty
```

## Field Value

[DependencyProperty](#)

# Methods

## GetLangKey(DependencyObject)

Gets the value of the [LangKey](#) attached property.

```
public static string GetLangKey(DependencyObject obj)
```

Parameters

**obj** [DependencyObject](#)

Returns

[string](#)

## SetLangKey(DependencyObject, string)

Sets the value of the [LangKey](#) attached property.

```
public static void SetLangKey(DependencyObject obj, string value)
```

Parameters

**obj** [DependencyObject](#)

**value** [string](#)

# Namespace MBFWpfToolkit.Constants

## Enums

### [MbMessageIcon](#)

Defines icons that can be displayed in a custom message box. These icons represent different statuses or types of feedback for the user.

### [SkinType](#)

Defines available theme skins for the application UI.

# Enum MbMessageIcon

Namespace: [MBFWpfToolkit.Constants](#)

Assembly: MBFWpfToolkit.dll

Defines icons that can be displayed in a custom message box. These icons represent different statuses or types of feedback for the user.

```
public enum MbMessageIcon
```

## Fields

**Error = 16**

A white X in a circle with a red background, indicating an error or failure.

**Information = 64**

An exclamation point in a triangle with a yellow background, providing informational feedback.

**New = 512**

A trash bin or cross symbol, indicating the creation of a new item.

**None = 0**

No icon is displayed.

**Question = 48**

A question mark in a circle, prompting the user for confirmation or input.

**Remove = 1024**

A trash bin or cross symbol, indicating the removal or deletion of an item.

**Sound = 32**

A bold white X in a circle with a darker red background, indicating a critical error.

**Success = 256**

A lowercase letter 'i' in a circle with a green background, indicating success or completion.

**Warning = 128**

An exclamation point in a triangle with an orange background, warning about a potential issue.

# Enum SkinType

Namespace: [MBFWpfToolkit.Constants](#)

Assembly: MBFWpfToolkit.dll

Defines available theme skins for the application UI.

```
public enum SkinType
```

## Fields

**Dark = 1**

A dark theme with a darker background and lighter foreground elements.

**Default = 0**

The default (light or system-defined) application theme.

**Violet = 2**

A violet-colored theme for a more vibrant visual experience.

# Namespace MBFWpfToolkit.Controls

## Classes

### [MbMessageBox](#)

A custom message box window that supports HandyControl theming, icon display, and multiple button configurations.

### [MicroBimSprite](#)

Represents the MicroBIM Sprite window used for displaying custom interactive UI elements.

# Class MbMessageBox

Namespace: [MBFWpfToolkit.Controls](#)

Assembly: MBFWpfToolkit.dll

A custom message box window that supports HandyControl theming, icon display, and multiple button configurations.

```
public class MbMessageBox : MbWindow
```

## Inheritance

[object](#) ← [MbWindow](#) ← MbMessageBox

## Inherited Members

[MbWindow.IsThemeToggleVisibleProperty](#), [MbWindow.ShowLanguagesComboBoxProperty](#),  
[MbWindow.PluginNameProperty](#), [MbWindow.PluginDescriptionProperty](#),  
[MbWindow.PluginVersionProperty](#), [MbWindow.PrimaryBrushProperty](#), [MbWindow.ShowLogoProperty](#),  
[MbWindow.ShowPluginNameProperty](#), [MbWindow.ShowPluginVersionProperty](#),  
[MbWindow.InitializeWindow\(\)](#), [MbWindow.ShowSprite\(\)](#), [MbWindow.GetPrimaryBrush\(\)](#),  
[MbWindow.IsThemeToggleVisible](#), [MbWindow.ShowLanguagesComboBox](#), [MbWindow.PluginName](#),  
[MbWindow.PluginDescription](#), [MbWindow.PluginVersion](#), [MbWindow.PrimaryBrush](#),  
[MbWindow.ShowLogo](#), [MbWindow.ShowPluginName](#), [MbWindow.ShowPluginVersion](#),  
[MbWindow.MainResources](#)

## Methods

### Show(string, MessageBoxButton, MbMessageIcon)

Displays the message box with the specified message and button configuration.

```
public static MessageBoxResult Show(string message, MessageBoxButton button =  
MessageBoxButton.OK, MbMessageIcon icon = MbMessageIcon.None)
```

## Parameters

**message** [string](#)

The message to display.

**button** [MessageBoxButton](#)

The button configuration.

**icon** [MbMessageIcon](#)

The icon to display.

Returns

[MessageBoxResult](#)

The result of the button clicked.

# Class MicroBimSprite

Namespace: [MBFWpfToolkit.Controls](#)

Assembly: MBFWpfToolkit.dll

Represents the MicroBIM Sprite window used for displaying custom interactive UI elements.

```
public class MicroBimSprite : Window, IComponentConnector
```

## Inheritance

[object](#) ← MicroBimSprite

## Implements

[IComponentConnector](#)

## Remarks

This window is styled using HandyControl and can be used to provide compact, reusable, or themed dialogs/popups.

## Constructors

### MicroBimSprite()

Initializes a new instance of the [MicroBimSprite](#) window. Loads the associated XAML and prepares the window for display.

```
public MicroBimSprite()
```

## Methods

### InitializeComponent()

InitializeComponent

```
public void InitializeComponent()
```

# Namespace MBFWpfToolkit.Controls.Help

## Classes

### [HelpWizard](#)

Represents an interactive in-app help wizard that guides users through UI elements step-by-step.

# Class HelpWizard

Namespace: [MBFWpfToolkit.Controls.Help](#)

Assembly: MBFWpfToolkit.dll

Represents an interactive in-app help wizard that guides users through UI elements step-by-step.

```
public class HelpWizard : MbWindow, IComponentConnector
```

## Inheritance

[object](#) ← [MbWindow](#) ← HelpWizard

## Implements

[IComponentConnector](#)

## Inherited Members

[MbWindow.IsThemeToggleVisibleProperty](#), [MbWindow.ShowLanguagesComboBoxProperty](#),  
[MbWindow.PluginNameProperty](#), [MbWindow.PluginDescriptionProperty](#),  
[MbWindow.PluginVersionProperty](#), [MbWindow.PrimaryBrushProperty](#), [MbWindow.ShowLogoProperty](#),  
[MbWindow.ShowPluginNameProperty](#), [MbWindow.ShowPluginVersionProperty](#),  
[MbWindow.InitializeWindow\(\)](#), [MbWindow.ShowSprite\(\)](#), [MbWindow.GetPrimaryBrush\(\)](#),  
[MbWindow.IsThemeToggleVisible](#), [MbWindow.ShowLanguagesComboBox](#), [MbWindow.PluginName](#),  
[MbWindow.PluginDescription](#), [MbWindow.PluginVersion](#), [MbWindow.PrimaryBrush](#),  
[MbWindow.ShowLogo](#), [MbWindow.ShowPluginName](#), [MbWindow.ShowPluginVersion](#),  
[MbWindow.MainResources](#)

# Constructors

## HelpWizard(MbWindow)

Initializes a new instance of the [HelpWizard](#) class.

```
public HelpWizard(MbWindow mainWindow)
```

## Parameters

**mainWindow** [MbWindow](#)

The main application window hosting the UI elements to be highlighted in the help wizard.

# Properties

## CurrentStep

Gets or sets the index of the currently active help step.

```
public int CurrentStep { get; set; }
```

Property Value

[int](#)

## Steps

Gets or sets the list of all help steps.

```
public List<HelpStep> Steps { get; set; }
```

Property Value

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

# Methods

## InitializeComponent()

InitializeComponent

```
public void InitializeComponent()
```

# Namespace MBFWpfToolkit.Convertisers

## Classes

### [BooleanToStringConverter](#)

Converts a [bool](#) value to a [string](#) based on a semicolon-delimited parameter string.

### [BooleanToVisibilityWithInverseConverter](#)

Converts a [bool](#) value to a [Visibility](#) value. Supports optional inversion using a converter parameter.

### [EnumDisplayNameConverter](#)

A value converter for converting between enum values and their display names specified by the [DisplayAttribute](#).

### [ValidationMultiConverter](#)

A multi-value converter that validates a group of input values (typically bound from multiple [TextBox](#) elements). It checks for empty input and validation errors to determine whether a button (or any control) should be enabled.

# Class Boolean2StringConverter

Namespace: [MBFWpfToolkit.Converters](#)

Assembly: MBFWpfToolkit.dll

Converts a [bool](#) value to a [string](#) based on a semicolon-delimited parameter string.

```
public class Boolean2StringConverter : IValueConverter
```

## Inheritance

[object](#) ← Boolean2StringConverter

## Implements

[IValueConverter](#)

## Inherited Members

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

## Remarks

Used to convert a boolean to two possible string values (e.g., "Off;On") for UI display.

## Constructors

### Boolean2StringConverter()

```
public Boolean2StringConverter()
```

## Methods

### Convert(object, Type, object, CultureInfo)

Converts a boolean value to a string.

```
public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
```

## Parameters

**value** [object](#)

The boolean value to convert.

**targetType** [Type](#)

The target type (should be [string](#)).

**parameter** [object](#)

A semicolon-separated string representing the two output values (e.g., "FalseValue;TrueValue").

**culture** [CultureInfo](#)

The current culture (not used).

## Returns

[object](#)

Returns the second string (after semicolon) if **value** is **true**, or the first string if **false**. Returns an empty string if input is invalid.

## ConvertBack(object, Type, object, CultureInfo)

Not implemented. Conversion back is not supported.

```
public object ConvertBack(object value, Type targetType, object parameter,  
CultureInfo culture)
```

## Parameters

**value** [object](#)

**targetType** [Type](#)

**parameter** [object](#)

**culture** [CultureInfo](#)

## Returns

[object](#)

[DoNothing](#)

# Class BooleanToVisibilityWithInverseConverter

Namespace: [MBFWpfToolkit.Converters](#)

Assembly: MBFWpfToolkit.dll

Converts a [bool](#) value to a [Visibility](#) value. Supports optional inversion using a converter parameter.

```
public class BooleanToVisibilityWithInverseConverter : IValueConverter
```

## Inheritance

[object](#) ← BooleanToVisibilityWithInverseConverter

## Implements

[IValueConverter](#)

## Inherited Members

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

## Remarks

If the parameter is set to "Inverse" (case-insensitive), the visibility result will be inverted. Useful for showing or hiding UI elements based on boolean flags with optional inversion.

## Constructors

### BooleanToVisibilityWithInverseConverter()

```
public BooleanToVisibilityWithInverseConverter()
```

## Methods

### Convert(object, Type, object, CultureInfo)

Converts a [bool](#) to [Visibility](#).

```
public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
```

## Parameters

**value** [object](#)

The boolean value to convert.

**targetType** [Type](#)

The target type (expected to be [Visibility](#)).

**parameter** [object](#)

Optional parameter. If set to "Inverse", the result will be inverted.

**culture** [CultureInfo](#)

Culture info (not used).

## Returns

[object](#)

Returns [Visible](#) if **value** is true, or [Collapsed](#) if false. If "Inverse" is passed as a parameter, the logic is flipped.

## ConvertBack(object, Type, object, CultureInfo)

Converts a [Visibility](#) value back to [bool](#).

```
public object ConvertBack(object value, Type targetType, object parameter,  
CultureInfo culture)
```

## Parameters

**value** [object](#)

The visibility value to convert.

**targetType** [Type](#)

The target type (expected to be [bool](#)).

**parameter** [object](#)

Optional parameter (not used in ConvertBack).

#### **culture** [CultureInfo](#)

Culture info (not used).

Returns

#### **object**

Returns **true** if **value** is [Visible](#), otherwise **false**.

# Class EnumDisplayNameConverter

Namespace: [MBFWpfToolkit.Converters](#)

Assembly: MBFWpfToolkit.dll

A value converter for converting between enum values and their display names specified by the [Display Attribute](#).

```
public class EnumDisplayNameConverter : IValueConverter
```

## Inheritance

[object](#) ← EnumDisplayNameConverter

## Implements

[IValueConverter](#)

## Inherited Members

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

## Constructors

### EnumDisplayNameConverter()

```
public EnumDisplayNameConverter()
```

## Methods

### Convert(object, Type, object, CultureInfo)

Converts an [Enum](#) value to its display name specified in the [DisplayAttribute](#).

```
public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
```

## Parameters

### **value** [object](#)

The enum value to be converted.

### **targetType** [Type](#)

The target type of the binding (not used).

### **parameter** [object](#)

An optional parameter for the converter (not used).

### **culture** [CultureInfo](#)

The culture to be used in the converter (not used).

Returns

### [object](#)

The display name of the enum value, or its string representation if no [DisplayAttribute](#) is found.

## ConvertBack(object, Type, object, CultureInfo)

Converts a display name back to the corresponding [Enum](#) value.

```
public object ConvertBack(object value, Type targetType, object parameter,  
CultureInfo culture)
```

Parameters

### **value** [object](#)

The display name to be converted back to the enum value.

### **targetType** [Type](#)

The target enum type.

### **parameter** [object](#)

An optional parameter for the converter (not used).

### **culture** [CultureInfo](#)

The culture to be used in the converter (not used).

Returns

[object](#)

The corresponding enum value that matches the display name or field name.

Exceptions

[ArgumentException](#)

Thrown if no matching enum value is found for the display name.

# Class ValidationMultiConverter

Namespace: [MBFWpfToolkit.Converters](#)

Assembly: MBFWpfToolkit.dll

A multi-value converter that validates a group of input values (typically bound from multiple [TextBox](#) elements). It checks for empty input and validation errors to determine whether a button (or any control) should be enabled.

```
public class ValidationMultiConverter : IMultiValueConverter
```

## Inheritance

[object](#) ← ValidationMultiConverter

## Implements

[IMultiValueConverter](#)

## Inherited Members

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

## Remarks

This converter expects alternating values: [Text, HasError, Text, HasError, ...]. If any Text is null/empty or any HasError is true, the result is false.

## Constructors

### ValidationMultiConverter()

```
public ValidationMultiConverter()
```

## Methods

### Convert(object[], Type, object, CultureInfo)

Validates a group of values to determine if a control (e.g., a button) should be enabled.

```
public object Convert(object[] values, Type targetType, object parameter,
CultureInfo culture)
```

## Parameters

**values** [object](#)[]

An array of values from multiple bindings. Alternating pattern: Text, HasError, Text, HasError...

**targetType** [Type](#)

The target type of the binding (usually [bool](#)).

**parameter** [object](#)

Optional parameter (not used).

**culture** [CultureInfo](#)

The culture info (not used).

## Returns

[object](#)

Returns [true](#) if all inputs are non-empty and have no validation errors; otherwise [false](#).

## ConvertBack(object, Type[], object, CultureInfo)

ConvertBack is not implemented. Returns [DoNothing](#).

```
public object[] ConvertBack(object value, Type[] targetTypes, object parameter,
CultureInfo culture)
```

## Parameters

**value** [object](#)

**targetTypes** [Type](#)[]

**parameter** [object](#)

culture [CultureInfo](#)

Returns

[object](#)[]

An array of [DoNothing](#).

# Namespace MBFWpfToolkit.Extensions

## Classes

### [ListExtensions](#)

Provides extension methods for [List<T>](#) to support additional operations.

# Class ListExtensions

Namespace: [MBFWpfToolkit.Extensions](#)

Assembly: MBFWpfToolkit.dll

Provides extension methods for [List<T>](#) to support additional operations.

```
public static class ListExtensions
```

## Inheritance

[object](#) ← ListExtensions

## Inherited Members

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

# Methods

## Move<T>(List<T>, int, int)

Moves an item within a [List<T>](#) from one index to another.

```
public static void Move<T>(this List<T> list, int oldIndex, int newIndex)
```

### Parameters

**list** [List](#)<T>

The list in which the item will be moved.

**oldIndex** [int](#)

The current index of the item.

**newIndex** [int](#)

The target index to move the item to.

### Type Parameters

T

The type of elements in the list.

## Remarks

If `oldIndex` and `newIndex` are equal, or if either index is out of range, the method does nothing.

# Namespace MBFWpfToolkit.Helpers

## Classes

### [MbNotification](#)

Provides utility methods to display styled notifications using HandyControl's HandyControl.Controls.Growl system.

### [MbResourceHelper](#)

Provides helper methods for retrieving theme and skin resources in a WPF application.

### [PasswordHelper](#)

Provides attached properties and helper methods to enable binding for the Password property of PasswordBox controls.

# Class MbNotification

Namespace: [MBFWpfToolkit.Helpers](#)

Assembly: MBFWpfToolkit.dll

Provides utility methods to display styled notifications using HandyControl's HandyControl.Controls.Growl system.

```
public static class MbNotification
```

## Inheritance

[object](#) ← MbNotification

## Inherited Members

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

## Methods

### ShowError(string, int)

Displays an error notification with a red cross icon.

```
public static void ShowError(string message, int waitTime = 2)
```

#### Parameters

**message** [string](#)

The message to be shown in the notification.

**waitTime** [int](#)

The duration (in seconds) the notification remains visible. Default is 2 seconds.

### ShowInfo(string, int)

Displays an informational notification with a blue info icon.

```
public static void ShowInfo(string message, int waitTime = 2)
```

## Parameters

**message** [string](#)

The message to be shown in the notification.

**waitTime** [int](#)

The duration (in seconds) the notification remains visible. Default is 2 seconds.

## ShowSuccess(string, int)

Displays a success notification with a green check icon.

```
public static void ShowSuccess(string message, int waitTime = 2)
```

## Parameters

**message** [string](#)

The message to be shown in the notification.

**waitTime** [int](#)

The duration (in seconds) the notification remains visible. Default is 2 seconds.

# Class MbResourceHelper

Namespace: [MBFWpfToolkit.Helpers](#)

Assembly: MBFWpfToolkit.dll

Provides helper methods for retrieving theme and skin resources in a WPF application.

```
public class MbResourceHelper
```

## Inheritance

[object](#) ← MbResourceHelper

## Inherited Members

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

## Constructors

### MbResourceHelper()

```
public MbResourceHelper()
```

## Methods

### GetResource<T>(string)

Retrieves a resource from the application's merged resource dictionaries by key.

```
public static T GetResource<T>(string key)
```

#### Parameters

**key** [string](#)

The key of the resource to retrieve.

Returns

T

The resource cast to type T if found; otherwise, the default value of T.

Type Parameters

T

The expected type of the resource.

## GetSkin(SkinType)

Loads a built-in HandyControl skin resource dictionary by skin type.

```
public static ResourceDictionary GetSkin(SkinType skin)
```

Parameters

skin [SkinType](#)

The [SkinType](#) indicating which skin to load.

Returns

[ResourceDictionary](#)

The loaded [ResourceDictionary](#).

## GetSkin(Assembly, string, SkinType)

Loads a skin resource dictionary from a specified assembly and path.

```
public static ResourceDictionary GetSkin(Assembly assembly, string themePath, SkinType skin)
```

Parameters

assembly [Assembly](#)

The assembly containing the skin resources.

`themePath` [string](#)

The path to the skin directory inside the assembly.

`skin` [SkinType](#)

The [SkinType](#) indicating which skin to load.

Returns

[ResourceDictionary](#)

The loaded [ResourceDictionary](#) for the specified skin.

## GetStandaloneTheme()

Loads the standalone HandyControl theme resource dictionary.

```
public static ResourceDictionary GetStandaloneTheme()
```

Returns

[ResourceDictionary](#)

The loaded [ResourceDictionary](#).

## GetTheme()

Retrieves the current theme resource dictionary. If not already loaded, it loads the standalone HandyControl theme.

```
public static ResourceDictionary GetTheme()
```

Returns

[ResourceDictionary](#)

The [ResourceDictionary](#) representing the current theme.

# Class PasswordHelper

Namespace: [MBFWpfToolkit.Helpers](#)

Assembly: MBFWpfToolkit.dll

Provides attached properties and helper methods to enable binding for the Password property of PasswordBox controls.

```
public static class PasswordHelper
```

## Inheritance

[object](#) ← PasswordHelper

## Inherited Members

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

## Fields

### AttachProperty

Attached property to enable or disable binding functionality for PasswordBox.

```
public static readonly DependencyProperty AttachProperty
```

### Field Value

[DependencyProperty](#)

### PasswordProperty

Attached property to bind the password value.

```
public static readonly DependencyProperty PasswordProperty
```

### Field Value

## Methods

### GetAttach(DependencyObject)

Gets the value of the Attach property.

```
public static bool GetAttach(DependencyObject dp)
```

Parameters

dp [DependencyObject](#)

Returns

[bool](#)

### GetPassword(DependencyObject)

Gets the bound password string from the attached property.

```
public static string GetPassword(DependencyObject dp)
```

Parameters

dp [DependencyObject](#)

Returns

[string](#)

### SetAttach(DependencyObject, bool)

Sets the Attach property to enable password binding on the given object.

```
public static void SetAttach(DependencyObject dp, bool value)
```

## Parameters

dp [DependencyObject](#)

value [bool](#)

## SetPassword(DependencyObject, string)

Sets the bound password string on the attached property.

```
public static void SetPassword(DependencyObject dp, string value)
```

## Parameters

dp [DependencyObject](#)

value [string](#)

# Namespace MBFWpfToolkit.MbViewModel

## Classes

### [MbBaseViewModel](#)

Base ViewModel class providing theme and language management functionality.

# Class MbBaseViewModel

Namespace: [MBFWpfToolkit.MbViewModel](#)

Assembly: MBFWpfToolkit.dll

Base ViewModel class providing theme and language management functionality.

```
public class MbBaseViewModel : ObservableValidator
```

## Inheritance

[object](#) ← MbBaseViewModel

## Constructors

### MbBaseViewModel()

Initializes a new instance of the [MbBaseViewModel](#) class.

```
public MbBaseViewModel()
```

## Properties

### Culture

Gets or sets the current culture.

```
public CultureInfo Culture { get; set; }
```

### Property Value

[CultureInfo](#)

### IsDark

```
public bool IsDark { get; set; }
```

## Property Value

[bool](#) ↗

## this[string]

Indexer to access localized values dynamically.

```
public string this[string key] { get; }
```

## Parameters

**key** [string](#) ↗

The key of the localized value.

## Property Value

[string](#) ↗

The localized value or a placeholder if not found.

## MbWindow

Gets or sets the associated MbWindow.

```
public MbWindow? MbWindow { get; set; }
```

## Property Value

[MbWindow](#)

## OpenLinkCommand

Gets an CommunityToolkit.Mvvm.Input.IRelayCommand<> instance wrapping MBFWpfToolkit.MbView Model.MbBaseViewModel.OpenLink(System.Object).

```
public IRelayCommand<object> OpenLinkCommand { get; }
```

## Property Value

IRelayCommand<[object](#)>

## SelectedLanguage

Gets or sets the selected language for the application.

```
public CultureInfo SelectedLanguage { get; set; }
```

## Property Value

[CultureInfo](#)

## SupportedLanguages

Supported languages for the application.

```
public ObservableCollection<CultureInfo> SupportedLanguages { get; }
```

## Property Value

[ObservableCollection](#)<[CultureInfo](#)>

## SwitchThemeCommand

Gets an CommunityToolkit.Mvvm.Input.I RelayCommand<> instance wrapping MBFWpfToolkit.MbView Model.MbBaseViewModel.SwitchTheme(System.Windows.Controls.Primitives.ToggleButton).

```
public I RelayCommand<ToggleButton> SwitchThemeCommand { get; }
```

## Property Value

IRelayCommand<[ToggleButton](#)>

## Methods

## GetLang(string)

Retrieves a localized string by its key.

```
public string GetLang(string key)
```

Parameters

**key** [string](#)

The key of the string to retrieve.

Returns

[string](#)

The localized string or a placeholder if not found.

## InitializeViewModel(MbWindow)

Initializes the ViewModel with the provided window and sets default values.

```
public void InitializeViewModel(MbWindow mbWindow)
```

Parameters

**mbWindow** [MbWindow](#)

The window associated with this ViewModel.

## SetLang(DependencyObject, DependencyProperty, string)

Binds a dependency property to a language resource key.

```
public void SetLang(DependencyObject dependencyObject, DependencyProperty dependencyProperty, string key)
```

Parameters

**dependencyObject** [DependencyObject](#)

The target dependency object.

**dependencyProperty** [DependencyProperty](#)↗

The target dependency property.

**key** [string](#)↗

The key of the language resource.

# Namespace MBFWpfToolkit.Models

## Classes

### [HelpStep](#)

Represents a help step in the UI help wizard with an associated UI element, descriptive text, and order.

# Class HelpStep

Namespace: [MBFWpfToolkit.Models](#)

Assembly: MBFWpfToolkit.dll

Represents a help step in the UI help wizard with an associated UI element, descriptive text, and order.

```
public class HelpStep
```

## Inheritance

[object](#) ← HelpStep

## Inherited Members

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

# Constructors

## HelpStep()

```
public HelpStep()
```

# Properties

## Element

Gets or sets the UI element associated with this help step.

```
public UIElement Element { get; set; }
```

## Property Value

[UIElement](#)

## Remarks

This UI element is used to anchor tooltips or visual guidance during the help walkthrough.

## StepNumber

Gets or sets the order number of the step in the help sequence.

```
public int StepNumber { get; set; }
```

Property Value

[int](#)

## StepText

Gets or sets the textual description of the step to be shown to the user.

```
public string StepText { get; set; }
```

Property Value

[string](#)

# Namespace MBFWpfToolkit.Properties.Langs

## Classes

### [Lang](#)

A strongly-typed resource class, for looking up localized strings, etc.

# Class Lang

Namespace: [MBFWpfToolkit.Properties.Langs](#)

Assembly: MBFWpfToolkit.dll

A strongly-typed resource class, for looking up localized strings, etc.

```
public class Lang
```

## Inheritance

[object](#) ← Lang

## Inherited Members

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

# Properties

## About

Looks up a localized string similar to About.

```
public static string About { get; }
```

## Property Value

[string](#)

## Culture

Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.

```
public static CultureInfo Culture { get; set; }
```

## Property Value

## ResourceManager

Returns the cached ResourceManager instance used by this class.

```
public static ResourceManager ResourceManager { get; }
```

### Property Value

[ResourceManager](#)

# Namespace MBFWpfToolkit.Utils

## Classes

### [WindowUtils](#)

Provides utility functions for managing windows and processes in a WPF application.

# Class WindowUtils

Namespace: [MBFWpfToolkit.Utils](#)

Assembly: MBFWpfToolkit.dll

Provides utility functions for managing windows and processes in a WPF application.

```
public static class WindowUtils
```

## Inheritance

[object](#) ← WindowUtils

## Inherited Members

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

# Methods

## OpenLink(object)

Opens a link in the default web browser or associated application.

```
public static void OpenLink(object url)
```

## Parameters

`url` [object](#)

The URL or file path to open. This should be a valid string representation of the resource.

## Exceptions

[ArgumentNullException](#)

Thrown when the `url` is null.

[InvalidOperationException](#)

Thrown when the `url` is not a valid string.

# Namespace MBFWpfToolkit.ValidationRules

## Classes

### [DecimalValidationRule](#)

Validation rule for decimal input with localization support.

### [DoubleValidationRule](#)

Validation rule for double input with localization support.

### [IntegerValidationRule](#)

Validation rule for integer input with localization support.

### [StringValidationRule](#)

Validation rule for string input with localization support.

# Class DecimalValidationRule

Namespace: [MBFWpfToolkit.ValidationRules](#)

Assembly: MBFWpfToolkit.dll

Validation rule for decimal input with localization support.

```
public class DecimalValidationRule : ValidationRule
```

## Inheritance

[object](#) ← [ValidationRule](#) ← DecimalValidationRule

## Inherited Members

[ValidationRule.Validate\(object, CultureInfo, BindingExpressionBase\)](#) ,  
[ValidationRule.Validate\(object, CultureInfo, BindingGroup\)](#) , [ValidationRule.ValidationStep](#) ,  
[ValidationRule.ValidatesOnTargetUpdated](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### DecimalValidationRule()

```
public DecimalValidationRule()
```

## Properties

### IsRequired

Gets or sets a value indicating whether the input is required.

```
public bool IsRequired { get; set; }
```

### Property Value

[bool](#)

## Maximum

Gets or sets the maximum allowable value.

```
public decimal Maximum { get; set; }
```

Property Value

[decimal](#)

## Minimum

Gets or sets the minimum allowable value.

```
public decimal Minimum { get; set; }
```

Property Value

[decimal](#)

## Methods

### Validate(object, CultureInfo)

Validates the input value.

```
public override ValidationResult Validate(object value, CultureInfo cultureInfo)
```

Parameters

**value** [object](#)

The value to validate.

**cultureInfo** [CultureInfo](#)

The culture information.

Returns

## ValidationResult

A [ValidationResult](#) indicating whether the value is valid.

# Class DoubleValidationRule

Namespace: [MBFWpfToolkit.ValidationRules](#)

Assembly: MBFWpfToolkit.dll

Validation rule for double input with localization support.

```
public class DoubleValidationRule : ValidationRule
```

## Inheritance

[object](#) ← [ValidationRule](#) ← DoubleValidationRule

## Inherited Members

[ValidationRule.Validate\(object, CultureInfo, BindingExpressionBase\)](#) ,  
[ValidationRule.Validate\(object, CultureInfo, BindingGroup\)](#) , [ValidationRule.ValidationStep](#) ,  
[ValidationRule.ValidatesOnTargetUpdated](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### DoubleValidationRule()

```
public DoubleValidationRule()
```

## Properties

### IsRequired

Gets or sets a value indicating whether the input is required.

```
public bool IsRequired { get; set; }
```

### Property Value

[bool](#)

## Maximum

Gets or sets the maximum allowable value.

```
public double Maximum { get; set; }
```

Property Value

[double](#)

## Minimum

Gets or sets the minimum allowable value.

```
public double Minimum { get; set; }
```

Property Value

[double](#)

## Methods

### Validate(object, CultureInfo)

Validates the input value.

```
public override ValidationResult Validate(object value, CultureInfo cultureInfo)
```

Parameters

**value** [object](#)

The value to validate.

**cultureInfo** [CultureInfo](#)

The culture information.

Returns

## ValidationResult

A [ValidationResult](#) indicating whether the value is valid.

# Class IntegerValidationRule

Namespace: [MBFWpfToolkit.ValidationRules](#)

Assembly: MBFWpfToolkit.dll

Validation rule for integer input with localization support.

```
public class IntegerValidationRule : ValidationRule
```

## Inheritance

[object](#) ← [ValidationRule](#) ← IntegerValidationRule

## Inherited Members

[ValidationRule.Validate\(object, CultureInfo, BindingExpressionBase\)](#) ,  
[ValidationRule.Validate\(object, CultureInfo, BindingGroup\)](#) , [ValidationRule.ValidationStep](#) ,  
[ValidationRule.ValidatesOnTargetUpdated](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### IntegerValidationRule()

```
public IntegerValidationRule()
```

## Properties

### IsRequired

Gets or sets a value indicating whether the input is required.

```
public bool IsRequired { get; set; }
```

### Property Value

[bool](#)

## Maximum

Gets or sets the maximum allowable value.

```
public int Maximum { get; set; }
```

### Property Value

[int](#)

## Minimum

Gets or sets the minimum allowable value.

```
public int Minimum { get; set; }
```

### Property Value

[int](#)

## Methods

### Validate(object, CultureInfo)

Validates the input value.

```
public override ValidationResult Validate(object value, CultureInfo cultureInfo)
```

#### Parameters

**value** [object](#)

The value to validate.

**cultureInfo** [CultureInfo](#)

The culture information.

#### Returns

## ValidationResult

A [ValidationResult](#) indicating whether the value is valid.

# Class StringValidationRule

Namespace: [MBFWpfToolkit.ValidationRules](#)

Assembly: MBFWpfToolkit.dll

Validation rule for string input with localization support.

```
public class StringValidationRule : ValidationRule
```

## Inheritance

[object](#) ← [ValidationRule](#) ← StringValidationRule

## Inherited Members

[ValidationRule.Validate\(object, CultureInfo, BindingExpressionBase\)](#) ,  
[ValidationRule.Validate\(object, CultureInfo, BindingGroup\)](#) , [ValidationRule.ValidationStep](#) ,  
[ValidationRule.ValidatesOnTargetUpdated](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#)

## Constructors

### StringValidationRule()

```
public StringValidationRule()
```

## Properties

### IsRequired

Gets or sets a value indicating whether the input is required.

```
public bool IsRequired { get; set; }
```

### Property Value

[bool](#)

## MaximumLength

Gets or sets the maximum allowable length.

```
public int MaximumLength { get; set; }
```

Property Value

[int](#)

## MinimumLength

Gets or sets the minimum allowable length.

```
public int MinimumLength { get; set; }
```

Property Value

[int](#)

## Pattern

Gets or sets the regular expression pattern for validation.

```
public string Pattern { get; set; }
```

Property Value

[string](#)

## Methods

### Validate(object, CultureInfo)

Validates the input value.

```
public override ValidationResult Validate(object value, CultureInfo cultureInfo)
```

## Parameters

**value** [object](#)

The value to validate.

**cultureInfo** [CultureInfo](#)

The culture information.

## Returns

[ValidationResult](#)

A [ValidationResult](#) indicating whether the value is valid.

# Namespace OnMethodBoundaryAspect

## Compile

### Classes

[MethodInfos](#)

# Class MethodInfos

Namespace: [OnMethodBoundaryAspectCompile](#)

Assembly: MB.Common.dll

```
public static class MethodInfo
```

## Inheritance

[object](#) ← MethodInfo

## Inherited Members

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

## Fields

\_methodInfo\_

A71B48C493D29E2B8313E53FD0672C4DBA5A075AF264536B3A  
67D76FBCE1A69E

```
public static readonly MethodBase  
_methodInfo_A71B48C493D29E2B8313E53FD0672C4DBA5A075AF264536B3A67D76FBCE1A69E
```

Field Value

[MethodBase](#)

\_methodInfo\_

C4071C8024CEA028B080B887AE95502EE04771BA7A3F0711DA  
C2B879E57DD7E6

```
public static readonly MethodBase  
_methodInfo_C4071C8024CEA028B080B887AE95502EE04771BA7A3F0711DAC2B879E57DD7E6
```

## Field Value

[MethodBase](#) ↗

# Namespace Revit.Extensions.Elements

## Classes

### [PipeExtensions](#)

Provides extension methods for Pipe elements in Revit.

### [SpaceExtensions](#)

Provides extension methods for Space elements in Revit.

# Class PipeExtensions

Namespace: [Revit.Extensions.Elements](#)

Assembly: Revit.Extensions.dll

Provides extension methods for Pipe elements in Revit.

```
public static class PipeExtensions
```

## Inheritance

[object](#) ← PipeExtensions

## Inherited Members

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

# Methods

## GetEndPoints(Pipe)

Retrieves the endpoints of a pipe as a list of XYZ points.

```
public static MbResult<List<XYZ>> GetEndPoints(this Pipe pipe)
```

### Parameters

**pipe** Pipe

The pipe to get the endpoints for.

### Returns

[MbResult](#)<[List](#)<XYZ>>

An [MbResult<T>](#) containing the list of endpoints or an error message. Possible error messages:

- "The pipe cannot be null." - Returned if the input pipe is null.
- "The pipe does not have a valid LocationCurve." - Returned if the pipe's location is not a valid curve.

## GetLineOfPipe(Pipe)

Gets the line representing the pipe's geometry.

```
public static MbResult<Line> GetLineOfPipe(this Pipe pipe)
```

### Parameters

**pipe** Pipe

The pipe to retrieve the line from.

### Returns

[MbResult](#)<Line>

An [MbResult](#)<T> containing the line or an error message. Possible error messages:

- "The pipe cannot be null." - Returned if the input pipe is null.
- "The pipe does not have a valid LocationCurve." - Returned if the pipe's location is not a valid curve.

## GetNearestEndPoints(Pipe, XYZ)

Finds the nearest and farthest endpoints of a pipe relative to a specified point.

```
public static MbResult<List<XYZ>> GetNearestEndPoints(this Pipe pipe, XYZ point)
```

### Parameters

**pipe** Pipe

The pipe to find endpoints for.

**point** XYZ

The reference point for distance comparison.

### Returns

[MbResult](#)<[List](#)<XYZ>>

An [MbResult<T>](#) containing a list of nearest and farthest endpoints or an error message. Possible error messages:

- "The pipe cannot be null." - Returned if the input pipe is null.
- "The reference point cannot be null." - Returned if the input point is null.
- "The pipe does not have a valid LocationCurve." - Returned if the pipe's location is not a valid curve.

## GetNearestOneSideConnector(Pipe, XYZ)

Finds the nearest one-side connector on a pipe to a specified point.

```
public static MbResult<Connector> GetNearestOneSideConnector(this Pipe pipe, XYZ point)
```

### Parameters

**pipe** Pipe

The pipe to search for connectors.

**point** XYZ

The point to measure the distance from.

### Returns

[MbResult<Connector>](#)

An [MbResult<T>](#) containing the nearest connector or an error message. Possible error messages:

- "The pipe cannot be null." - Returned if the input pipe is null.
- "The reference point cannot be null." - Returned if the input point is null.
- "No connectors found on the pipe." - Returned if the pipe does not have any valid connectors.
- "An error occurred while finding the nearest connector: {ErrorMessage}" - Returned if an unexpected exception occurs.

## GetPipeCurve(Pipe)

Gets the curve representing the pipe's geometry.

```
public static MbResult<Curve> GetPipeCurve(this Pipe pipe)
```

## Parameters

### **pipe** Pipe

The pipe to retrieve the curve from.

## Returns

### [MbResult](#) <Curve>

An [MbResult<T>](#) containing the curve or an error message. Possible error messages:

- "The pipe cannot be null." - Returned if the input pipe is null.
- "The pipe does not have a valid LocationCurve." - Returned if the pipe's location is not a valid curve.

## GetPipingSystemType(Pipe, Document)

Gets the piping system type associated with a pipe.

```
public static MbResult<PipingSystemType> GetPipingSystemType(this Pipe pipe,  
Document document)
```

## Parameters

### **pipe** Pipe

The pipe to retrieve the piping system type for.

### **document** Document

The Revit document containing the pipe.

## Returns

### [MbResult](#) <PipingSystemType>

An [MbResult<T>](#) containing the piping system type or an error message. Possible error messages:

- "The pipe cannot be null." - Returned if the input pipe is null.
- "The document cannot be null." - Returned if the input document is null.
- "The pipe is not associated with a piping system." - Returned if the pipe is not part of a piping system.

- "Failed to retrieve the piping system type." - Returned if the piping system type cannot be retrieved.
- "An error occurred while retrieving the piping system type: {ErrorMessage}" - Returned if an unexpected exception occurs.

# Class SpaceExtensions

Namespace: [Revit.Extensions.Elements](#)

Assembly: Revit.Extensions.dll

Provides extension methods for Space elements in Revit.

```
public static class SpaceExtensions
```

## Inheritance

[object](#) ← SpaceExtensions

## Inherited Members

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

# Methods

## GetBoundarySegment(Space)

Retrieves the boundary segments of a space as a list of curves.

```
public static MbResult<List<Curve>> GetBoundarySegment(this Space space)
```

### Parameters

**space** Space

The space to get the boundary segments for.

### Returns

[MbResult](#)<[List](#)<Curve>>

An [MbResult<T>](#) containing:

- A list of boundary curves if the operation is successful.
- An error message if the operation fails. Possible error messages include:

# Validate(Space)

Validates the space object to ensure it is valid and has a positive area.

```
public static MbResult<bool> Validate(this Space space)
```

## Parameters

**space** Space

The space to validate.

## Returns

[MbResult](#) <bool>

An [MbResult<T>](#) containing:

- true if the space is valid.
- An error message if the validation fails. Possible error messages include:

# Namespace Revit.Extensions.Extensions

## Classes

### [CollectionExtensions](#)

Provides extension methods for working with collections.

### [DocumentExtensions](#)

Provides extension methods for the Autodesk Revit API with the result pattern for better error handling.

### [ViewExtensions](#)

Provides extension methods for Revit views.

# Class CollectionExtensions

Namespace: [Revit.Extensions.Extensions](#)

Assembly: Revit.Extensions.dll

Provides extension methods for working with collections.

```
public static class CollectionExtensions
```

## Inheritance

[object](#) ← CollectionExtensions

## Inherited Members

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

## Methods

### ToObservableCollection<T>(IEnumerable<T>)

Converts an enumerable collection to an ObservableCollection.

```
public static ObservableCollection<T> ToObservableCollection<T>(this IEnumerable<T> source)
```

#### Parameters

[source](#) [IEnumerable](#)<T>

The source enumerable collection to convert.

#### Returns

[ObservableCollection](#)<T>

An ObservableCollection containing the elements from the source.

#### Type Parameters

T

The type of elements in the collection.

# Class DocumentExtensions

Namespace: [Revit.Extensions.Extensions](#)

Assembly: Revit.Extensions.dll

Provides extension methods for the Autodesk Revit API with the result pattern for better error handling.

```
public static class DocumentExtensions
```

## Inheritance

[object](#) ← DocumentExtensions

## Inherited Members

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

## Methods

### CreateAndPlaceColumnAtPoint(Document, XYZ, FamilySymbol, Level, Level)

Creates and places a column at a specified point in the Revit document.

```
public static MbResult<FamilyInstance> CreateAndPlaceColumnAtPoint(this Document doc, XYZ  
point, FamilySymbol columnType, Level baseLevel, Level topLevel)
```

#### Parameters

**doc** Document

The Revit document where the column will be placed.

**point** XYZ

The XYZ point where the column will be placed.

**columnType** FamilySymbol

The FamilySymbol representing the column type.

### **baseLevel** Level

The base level of the column.

### **topLevel** Level

The top level of the column.

Returns

[MbResult](#) <FamilyInstance>

An [MbResult<T>](#) containing:

- The created FamilyInstance if successful.
- An error message if any input is invalid or column creation fails.

## GetFamilyTypesByCategoryOrderedByName(Document, BuiltInCategory)

Retrieves all family types of the specified category from the given Revit document as an ObservableCollection.

```
public static MbResult<ObservableCollection<FamilySymbol>>
GetFamilyTypesByCategoryOrderedByName(this Document doc, BuiltInCategory category)
```

Parameters

### **doc** Document

The Revit document to search for family types.

### **category** BuiltInCategory

The BuiltInCategory to filter family types.

Returns

[MbResult](#) <[ObservableCollection](#)> <FamilySymbol>

An [MbResult<T>](#) containing:

- An ObservableCollection of FamilySymbol objects if successful.
- An error message if the document or category is invalid.

## GetLevelsOrderedByProperty<TKey>(Document, Func<Level, TKey>, bool)

Retrieves all levels from the specified Revit document as an ObservableCollection, ordered dynamically by a specified property.

```
public static MbResult<ObservableCollection<Level>> GetLevelsOrderedByProperty<TKey>(this  
Document doc, Func<Level, TKey> keySelector, bool ascending = true)
```

### Parameters

**doc** Document

The Revit document to search for levels.

**keySelector** [Func](#)<Level, TKey>

A function to extract a key from a Level for ordering.

**ascending** [bool](#)

If true, orders levels in ascending order; otherwise, descending.

### Returns

[MbResult](#)<[ObservableCollection](#)<Level>>

An [MbResult](#)<T> containing:

- An ObservableCollection of Level objects if successful.
- An error message if the document or keySelector is null.

### Type Parameters

**TKey**

The type of the property used for ordering.

## GetRebarBarTypes(Document)

Retrieves all rebar bar types from the specified Revit document as an ObservableCollection.

```
public static MbResult<ObservableCollection<RebarBarType>> GetRebarBarTypes(this
```

```
Document doc)
```

## Parameters

**doc** Document

The Revit document to retrieve rebar bar types from.

## Returns

[MbResult](#) <[ObservableCollection](#) <RebarBarType>>

An [MbResult<T>](#) containing:

- An ObservableCollection of RebarBarType objects if successful.
- An error message if the document is null.

## GetRebarShapes(Document)

Retrieves all rebar shapes from the specified Revit document.

```
public static MbResult<List<RebarShape>> GetRebarShapes(this Document doc)
```

## Parameters

**doc** Document

The Revit document to retrieve rebar shapes from.

## Returns

[MbResult](#) <[List](#) <RebarShape>>

An [MbResult<T>](#) containing:

- A list of RebarShape objects if successful.
- An error message if the document is null.

## SearchOrCreate3DView(Document, string, bool)

Searches for a 3D view with the specified name. If it exists, returns it. Optionally, creates a new 3D view if specified.

```
public static MbResult<View3D> SearchOrCreate3DView(this Document doc, string viewName, bool  
createView = false)
```

## Parameters

**doc** Document

The Revit document to search or create the 3D view in.

**viewName** [string](#)

The name of the 3D view to search for or create.

**createView** [bool](#)

If [true](#), creates a new 3D view if one does not exist.

## Returns

[MbResult](#)<View3D>

An [MbResult<T>](#) containing:

- The existing or newly created Autodesk.Revit.DB.View3D object if successful.
- An error message if the document is null, the view name is invalid, or creation fails.

# Class ViewExtensions

Namespace: [Revit.Extensions.Extensions](#)

Assembly: Revit.Extensions.dll

Provides extension methods for Revit views.

```
public static class ViewExtensions
```

## Inheritance

[object](#) ← ViewExtensions

## Inherited Members

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

# Methods

## GetVisibleElements(View, Document)

Retrieves all elements that are visible in the specified Revit view.

```
public static MbResult<List<Element>> GetVisibleElements(this View view, Document document)
```

### Parameters

**view** View

The Revit view to search for visible elements.

**document** Document

The Revit document containing the view.

### Returns

[MbResult](#)<[List](#)<Element>>

An [MbResult<T>](#) object containing:

- A list of elements visible in the specified view if the operation is successful.
- A failure result with the following error messages for invalid parameters:
  - "The view parameter cannot be null."
  - "The document parameter cannot be null."

# Namespace Revit.Extensions.Selections

## Classes

[SelectionExtensions](#)

# Class SelectionExtensions

Namespace: [Revit.Extensions.Selections](#)

Assembly: Revit.Extensions.dll

```
public static class SelectionExtensions
```

## Inheritance

[object](#) ← SelectionExtensions

## Inherited Members

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

# Methods

## GetSelectedElements<T>(UIDocument)

Gets the selected elements of the specified type from the document.

```
public static List<T> GetSelectedElements<T>(this UIDocument uiDocument) where T : Element
```

### Parameters

**uiDocument** UIDocument

The uiDocument to get the selected elements from.

### Returns

[List](#)<T>

A list of the selected elements of the specified type.

### Type Parameters

**T**

The type of elements to get.

## GetSelectedFamilyInstances(UIDocument)

Gets the selected family instances from the document.

```
public static List<FamilyInstance> GetSelectedFamilyInstances(this UIDocument uiDocument)
```

### Parameters

**uiDocument** UIDocument

The UI document to get the selected family instances from.

### Returns

[List](#) <FamilyInstance>

A list of the selected family instances.

## GetSelectedPipes(UIDocument)

Gets the selected pipes from the document.

```
public static List<Pipe> GetSelectedPipes(this UIDocument uiDocument)
```

### Parameters

**uiDocument** UIDocument

The UI document to get the selected pipes from.

### Returns

[List](#) <Pipe>

A list of the selected pipes.

## GetSelectedSpaces(UIDocument)

Gets the selected spaces from the document.

```
public static List<Space> GetSelectedSpaces(this UIDocument uiDocument)
```

## Parameters

**uiDocument** UIDocument

The UI document to get the selected spaces from.

## Returns

[List](#)<Space>

A list of the selected spaces.

## GetSelectedSprinklers(UIDocument)

Gets the selected sprinklers from the document.

```
public static List<FamilyInstance> GetSelectedSprinklers(this UIDocument uiDocument)
```

## Parameters

**uiDocument** UIDocument

The UI document to get the selected sprinklers from.

## Returns

[List](#)<FamilyInstance>

A list of the selected sprinklers.

# Namespace Revit.Extensions.Unit

## Classes

[MbUnitExtensions](#)

# Class MbUnitExtensions

Namespace: [Revit.Extensions.Unit](#)

Assembly: Revit.Extensions.dll

```
public static class MbUnitExtensions
```

## Inheritance

[object](#) ← MbUnitExtensions

## Inherited Members

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

# Fields

## Accuracy

```
public static double Accuracy
```

## Field Value

[double](#)

## IsImperial

```
public static bool IsImperial
```

## Field Value

[bool](#)

# Methods

## ConvertFromInternalUnits(double)

```
public static double ConvertFromInternalUnits(this double value)
```

Parameters

**value** [double](#)

Returns

[double](#)

## ConvertToInternalLength(double)

```
public static double ConvertToInternalLength(this double value)
```

Parameters

**value** [double](#)

Returns

[double](#)

## IdentifyDocument(Document)

```
public static void IdentifyDocument(Document document)
```

Parameters

**document** Document