

Namespace Docfx

Namespaces

[Docfx.Build](#)

[Docfx.DataContracts](#)

[Docfx.Dotnet](#)

[Docfx.Exceptions](#)

[Docfx.Pdf](#)

Classes

[BuildOptions](#)

Provides options to be used with `Docfx.Docset.Build(System.String,Docfx.BuildOptions)`.

[Docset](#)

Provides access to a set of documentations and their associated configs, compilations and models.

Enums

[MemberLayout](#)

Specifies the layout of members.

Class BuildOptions

Namespace: [Docfx](#)

Assembly: Docfx.App.dll

Provides options to be used with `Docfx.Docset.Build(System.String, Docfx.BuildOptions)`.

```
public class BuildOptions
```

Inheritance

[object](#)  ← [BuildOptions](#)

Inherited Members

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

Properties

ConfigureMarkdig

Configures the markdig markdown pipeline.

```
public Func<MarkdownPipelineBuilder, MarkdownPipelineBuilder>? ConfigureMarkdig { get;
```

Property Value

[Func](#)  <MarkdownPipelineBuilder, MarkdownPipelineBuilder>?

Class Docset

Namespace: [Docfx](#)

Assembly: Docfx.App.dll

Provides access to a set of documentations and their associated configs, compilations and models.

```
public static class Docset
```

Inheritance

[object](#)  ← [Docset](#)

Inherited Members

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

Methods

Build(string)

Builds a docset specified by docfx.json config.

```
public static Task Build(string configPath)
```

Parameters

configPath [string](#) 

The path to docfx.json config file.

Returns

[Task](#) 

A task to await for build completion.

Build(string, BuildOptions)

Builds a docset specified by docfx.json config.

```
public static Task Build(string configPath, BuildOptions options)
```

Parameters

configPath [string](#)

The path to docfx.json config file.

options [BuildOptions](#)

The build options.

Returns

[Task](#)

A task to await for build completion.

Pdf(string)

Builds a pdf specified by docfx.json config.

```
public static Task Pdf(string configPath)
```

Parameters

configPath [string](#)

The path to docfx.json config file.

Returns

[Task](#)

A task to await for build completion.

Pdf(string, BuildOptions)

Builds a pdf specified by docfx.json config.

```
public static Task Pdf(string configPath, BuildOptions options)
```

Parameters

configPath [string](#)[↗]

The path to docfx.json config file.

options [BuildOptions](#)

The build options.

Returns

[Task](#)[↗]

A task to await for build completion.

Enum MemberLayout

Namespace: [Docfx](#)

Assembly: Docfx.Dotnet.dll

Specifies the layout of members.

```
public enum MemberLayout
```

Fields

SamePage = 0

Place members in the same page as their containing type

SeparatePages = 1

Place members in separate pages

Namespace Docfx.DataContracts.Managed Reference

Classes

[AdditionalNotes](#)

[ApiParameter](#)

[ArgumentInfo](#)

[AttributeInfo](#)

[ExceptionInfo](#)

[ItemViewModel](#)

[LinkInfo](#)

[NamedArgumentInfo](#)

[PageViewModel](#)

[SyntaxDetailViewModel](#)

Enums

[LinkType](#)

[MemberType](#)

[SyntaxLanguage](#)

Class AdditionalNotes

Namespace: [Docfx.DataContracts.ManagedReference](#)

Assembly: Docfx.Dotnet.dll

```
public class AdditionalNotes
```

Inheritance

[object](#)  ← [AdditionalNotes](#)

Inherited Members

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

Properties

Caller

```
[YamlMember(Alias = "caller")]  
[JsonProperty("caller")]  
[JsonPropertyName("caller")]  
public string Caller { get; set; }
```

Property Value

[string](#) 

Implementer

```
[YamlMember(Alias = "implementer")]  
[JsonProperty("implementer")]  
[JsonPropertyName("implementer")]  
public string Implementer { get; set; }
```


Property Value

[string](#)

Inheritor

```
[YamlMember(Alias = "inheritor")]  
[JsonProperty("inheritor")]  
[JsonPropertyName("inheritor")]  
public string Inheritor { get; set; }
```

Property Value

[string](#)

Class ApiParameter

Namespace: [Docfx.DataContracts.ManagedReference](#)








Assembly: Docfx.Dotnet.dll

```
public class ApiParameter
```

Inheritance

[object](#)  ← [ApiParameter](#)

Inherited Members

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

Properties

Attributes

```
[YamlMember(Alias = "attributes")]  
[JsonProperty("attributes")]  
[JsonPropertyName("attributes")]  
[MergeOption(MergeOption.Ignore)]  
public List<AttributeInfo> Attributes { get; set; }
```

Property Value

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

Description

```
[YamlMember(Alias = "description")]  
[JsonProperty("description")]
```

```
[JsonPropertyName("description")]  
public string Description { get; set; }
```

Property Value

[string](#)

Name

```
[YamlMember(Alias = "id")]  
[JsonProperty("id")]  
[JsonPropertyName("id")]  
[MergeOption(MergeOption.MergeKey)]  
public string Name { get; set; }
```

Property Value

[string](#)

Type

```
[YamlMember(Alias = "type")]  
[JsonProperty("type")]  
[JsonPropertyName("type")]  
public string Type { get; set; }
```

Property Value

[string](#)

Class ArgumentInfo

Namespace: [Docfx.DataContracts.ManagedReference](#)

Assembly: Docfx.Dotnet.dll

```
public class ArgumentInfo
```

Inheritance

[object](#)  ← [ArgumentInfo](#)

Inherited Members

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

Properties

Type

```
[YamlMember(Alias = "type")]  
[JsonProperty("type")]  
[JsonPropertyName("type")]  
public string Type { get; set; }
```

Property Value

[string](#) 

Value

```
[YamlMember(Alias = "value")]  
[JsonProperty("value")]  
[JsonPropertyName("value")]  
public object Value { get; set; }
```

Property Value

[object](#) 

Class AttributeInfo

Namespace: [Docfx.DataContracts.ManagedReference](#)








Assembly: Docfx.Dotnet.dll

```
public class AttributeInfo
```

Inheritance

[object](#)  ← [AttributeInfo](#)

Inherited Members

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

Properties

Arguments

```
[YamlMember(Alias = "arguments")]  
[JsonProperty("arguments")]  
[JsonPropertyName("arguments")]  
public List<ArgumentInfo> Arguments { get; set; }
```

Property Value

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

Constructor

```
[YamlMember(Alias = "ctor")]  
[JsonProperty("ctor")]  
[JsonPropertyName("ctor")]  
public string Constructor { get; set; }
```

Property Value

[string](#)

NamedArguments

```
[YamlMember(Alias = "namedArguments")]  
[JsonProperty("namedArguments")]  
[JsonPropertyName("namedArguments")]  
public List<NamedArgumentInfo> NamedArguments { get; set; }
```

Property Value

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

Type

```
[YamlMember(Alias = "type")]  
[JsonProperty("type")]  
[JsonPropertyName("type")]  
public string Type { get; set; }
```

Property Value

[string](#)

Class ExceptionInfo

Namespace: [Docfx.DataContracts.ManagedReference](#)

Assembly: Docfx.Dotnet.dll

```
public class ExceptionInfo
```

Inheritance

[object](#)  ← [ExceptionInfo](#)

Inherited Members

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

Properties

CommentId

```
[YamlMember(Alias = "commentId")]  
[JsonProperty("commentId")]  
[JsonPropertyName("commentId")]  
[MergeOption(MergeOption.Ignore)]  
public string CommentId { get; set; }
```

Property Value

[string](#) 

Description

```
[YamlMember(Alias = "description")]  
[JsonProperty("description")]
```



```
[JsonPropertyName("description")]  
public string Description { get; set; }
```

Property Value

[string](#) 

Type

```
[YamlMember(Alias = "type")]  
[MergeOption(MergeOption.MergeKey)]  
[JsonProperty("type")]  
[JsonPropertyName("type")]  
public string Type { get; set; }
```

Property Value

[string](#) 

Class ItemViewModel

Namespace: [Docfx.DataContracts.ManagedReference](#)

Assembly: Docfx.Dotnet.dll

```
public class ItemViewModel : IOverwriteDocumentViewModel
```

Inheritance

[object](#)  ← [ItemViewModel](#)

Implements

IOverwriteDocumentViewModel

Inherited Members

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

Properties

AdditionalNotes

```
[YamlMember(Alias = "additionalNotes")]  
[JsonProperty("additionalNotes")]  
[JsonPropertyName("additionalNotes")]  
public AdditionalNotes AdditionalNotes { get; set; }
```

Property Value

[AdditionalNotes](#)

AssemblyNameList

```
[YamlMember(Alias = "assemblies")]
[MergeOption(MergeOption.Ignore)]
[JsonProperty("assemblies")]
[JsonPropertyName("assemblies")]
public List<string> AssemblyNameList { get; set; }
```

Property Value

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

Attributes

```
[YamlMember(Alias = "attributes")]
[JsonProperty("attributes")]
[JsonPropertyName("attributes")]
[MergeOption(MergeOption.Ignore)]
public List<AttributeInfo> Attributes { get; set; }
```

Property Value

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

Children

```
[YamlMember(Alias = "children")]
[MergeOption(MergeOption.Ignore)]
[JsonProperty("children")]
[JsonPropertyName("children")]
public List<string> Children { get; set; }
```

Property Value

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

CommentId

```
[YamlMember(Alias = "commentId")]
[JsonProperty("commentId")]
[JsonPropertyName("commentId")]
public string CommentId { get; set; }
```

Property Value

[string](#)

Conceptual

The markdown content from the overwrite document

```
[YamlMember(Alias = "conceptual")]
[JsonProperty("conceptual")]
[JsonPropertyName("conceptual")]
public string Conceptual { get; set; }
```

Property Value

[string](#)

DerivedClasses

```
[YamlMember(Alias = "derivedClasses")]
[MergeOption(MergeOption.Ignore)]
[JsonProperty("derivedClasses")]
[JsonPropertyName("derivedClasses")]
public List<string> DerivedClasses { get; set; }
```

Property Value

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

Documentation

The details for current overwrite document, containing the start/end line numbers, file path, and git info.

```
[YamlMember(Alias = "documentation")]  
[JsonProperty("documentation")]  
[JsonPropertyName("documentation")]  
public SourceDetail Documentation { get; set; }
```

Property Value

SourceDetail

Examples

```
[YamlMember(Alias = "example")]  
[JsonProperty("example")]  
[JsonPropertyName("example")]  
[MergeOption(MergeOption.Replace)]  
public List<string> Examples { get; set; }
```

Property Value

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

Exceptions

```
[YamlMember(Alias = "exceptions")]  
[JsonProperty("exceptions")]  
[JsonPropertyName("exceptions")]  
public List<ExceptionInfo> Exceptions { get; set; }
```

Property Value

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

ExtensionMethods

```
[YamlMember(Alias = "extensionMethods")]
[MergeOption(MergeOption.Ignore)]
[JsonProperty("extensionMethods")]
[JsonPropertyName("extensionMethods")]
public List<string> ExtensionMethods { get; set; }
```

Property Value

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

FullName

```
[YamlMember(Alias = "fullName")]
[JsonProperty("fullName")]
[JsonPropertyName("fullName")]
public string FullName { get; set; }
```

Property Value

[string](#)

FullNameForCSharp

```
[YamlIgnore]
[JsonIgnore]
[JsonIgnore]
public string FullNameForCSharp { get; set; }
```

Property Value

[string](#)

FullNameForVB

```
[YamlIgnore]
[JsonIgnore]
```

```
[JsonIgnore]  
public string FullNameForVB { get; set; }
```

Property Value

[string](#)

FullNames

```
[ExtensibleMember("fullName.")]  
[JsonIgnore]  
[JsonIgnore]  
public SortedList<string, string> FullNames { get; set; }
```

Property Value

[SortedList](#) <[string](#), [string](#)>

Href

```
[YamlMember(Alias = "href")]  
[JsonProperty("href")]  
[JsonPropertyName("href")]  
public string Href { get; set; }
```

Property Value

[string](#)

Id

```
[YamlMember(Alias = "id")]  
[JsonProperty("id")]  
[JsonPropertyName("id")]  
public string Id { get; set; }
```

Property Value

[string](#)

Implements

```
[YamlMember(Alias = "implements")]
[MergeOption(MergeOption.Ignore)]
[JsonProperty("implements")]
[JsonPropertyName("implements")]
public List<string> Implements { get; set; }
```

Property Value

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

Inheritance

```
[YamlMember(Alias = "inheritance")]
[MergeOption(MergeOption.Ignore)]
[JsonProperty("inheritance")]
[JsonPropertyName("inheritance")]
public List<string> Inheritance { get; set; }
```

Property Value

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

InheritedMembers

```
[YamlMember(Alias = "inheritedMembers")]
[MergeOption(MergeOption.Ignore)]
[JsonProperty("inheritedMembers")]
[JsonPropertyName("inheritedMembers")]
public List<string> InheritedMembers { get; set; }
```

Property Value

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

IsExplicitInterfaceImplementation

```
[YamlMember(Alias = "isEii")]  
[JsonProperty("isEii")]  
[JsonPropertyName("isEii")]  
public bool IsExplicitInterfaceImplementation { get; set; }
```

Property Value

[bool](#)

IsExtensionMethod

```
[YamlMember(Alias = "isExtensionMethod")]  
[JsonProperty("isExtensionMethod")]  
[JsonPropertyName("isExtensionMethod")]  
public bool IsExtensionMethod { get; set; }
```

Property Value

[bool](#)

Metadata

```
[ExtensibleMember]  
[JsonIgnore]  
[JsonIgnore]  
public Dictionary<string, object> Metadata { get; set; }
```

Property Value

[Dictionary](#) <[string](#), [object](#)>

Name

```
[YamlMember(Alias = "name")]  
[JsonProperty("name")]  
[JsonPropertyName("name")]  
public string Name { get; set; }
```

Property Value

[string](#)↗

NameForCSharp

```
[YamlIgnore]  
[JsonIgnore]  
[JsonIgnore]  
public string NameForCSharp { get; set; }
```

Property Value

[string](#)↗

NameForVB

```
[YamlIgnore]  
[JsonIgnore]  
[JsonIgnore]  
public string NameForVB { get; set; }
```

Property Value

[string](#)↗

NameWithType

```
[YamlMember(Alias = "nameWithType")]  
[JsonProperty("nameWithType")]  
[JsonPropertyName("nameWithType")]  
public string NameWithType { get; set; }
```

Property Value

[string](#)

NameWithTypeForCSharp

```
[YamlIgnore]  
[JsonIgnore]  
[JsonIgnore]  
public string NameWithTypeForCSharp { get; set; }
```

Property Value

[string](#)

NameWithTypeForVB

```
[YamlIgnore]  
[JsonIgnore]  
[JsonIgnore]  
public string NameWithTypeForVB { get; set; }
```

Property Value

[string](#)

Names

```
[ExtensibleMember("name.")]  
[JsonIgnore]
```

```
[JsonIgnore]  
public SortedList<string, string> Names { get; set; }
```

Property Value

[SortedList](#) <[string](#), [string](#)>

NamesWithType

```
[ExtensibleMember("nameWithType.")]  
[JsonIgnore]  
[JsonIgnore]  
public SortedList<string, string> NamesWithType { get; set; }
```

Property Value

[SortedList](#) <[string](#), [string](#)>

NamespaceName

```
[YamlMember(Alias = "namespace")]  
[JsonProperty("namespace")]  
[JsonPropertyName("namespace")]  
public string NamespaceName { get; set; }
```

Property Value

[string](#)

Overload

```
[YamlMember(Alias = "overload")]  
[JsonProperty("overload")]  
[JsonPropertyName("overload")]  
public string Overload { get; set; }
```

Property Value

[string](#)

Overridden

```
[YamlMember(Alias = "overridden")]  
[JsonProperty("overridden")]  
[JsonPropertyName("overridden")]  
public string Overridden { get; set; }
```

Property Value

[string](#)

Parent

```
[YamlMember(Alias = "parent")]  
[JsonProperty("parent")]  
[JsonPropertyName("parent")]  
public string Parent { get; set; }
```

Property Value

[string](#)

Platform

```
[YamlMember(Alias = "platform")]  
[JsonProperty("platform")]  
[JsonPropertyName("platform")]  
[MergeOption(MergeOption.Replace)]  
public List<string> Platform { get; set; }
```

Property Value

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

Remarks

```
[YamlMember(Alias = "remarks")]  
[JsonProperty("remarks")]  
[JsonPropertyName("remarks")]  
public string Remarks { get; set; }
```

Property Value

[string](#)

SeeAlsos

```
[YamlMember(Alias = "seealso")]  
[JsonProperty("seealso")]  
[JsonPropertyName("seealso")]  
public List<LinkInfo> SeeAlsos { get; set; }
```

Property Value

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

SeeAlsosUidReference

```
[YamlIgnore]  
[JsonIgnore]  
[JsonIgnore]  
public List<string> SeeAlsosUidReference { get; }
```

Property Value

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

Source

```
[YamlMember(Alias = "source")]
[JsonProperty("source")]
[JsonPropertyName("source")]
public SourceDetail Source { get; set; }
```

Property Value

SourceDetail

Summary

```
[YamlMember(Alias = "summary")]
[JsonProperty("summary")]
[JsonPropertyName("summary")]
public string Summary { get; set; }
```

Property Value

[string](#)

SupportedLanguages

```
[YamlMember(Alias = "langs")]
[JsonProperty("langs")]
[JsonPropertyName("langs")]
public string[] SupportedLanguages { get; set; }
```

Property Value

[string](#)[]

Syntax

```
[YamlMember(Alias = "syntax")]
[JsonProperty("syntax")]
```

```
[JsonPropertyName("syntax")]
public SyntaxDetailViewModel Syntax { get; set; }
```

Property Value

[SyntaxDetailViewModel](#)

Type

```
[YamlMember(Alias = "type")]
[JsonProperty("type")]
[JsonPropertyName("type")]
public MemberType? Type { get; set; }
```

Property Value

[MemberType?](#)

Uid

The uid for this overwrite document, as defined in YAML header

```
[YamlMember(Alias = "uid")]
[JsonProperty("uid")]
[JsonPropertyName("uid")]
[MergeOption(MergeOption.MergeKey)]
public string Uid { get; set; }
```

Property Value

[string](#) 

Class LinkInfo

Namespace: [Docfx.DataContracts.ManagedReference](#)








Assembly: Docfx.Dotnet.dll

```
public class LinkInfo
```

Inheritance

[object](#)  ← [LinkInfo](#)

Inherited Members

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

Properties

AltText

```
[YamlMember(Alias = "altText")]  
[JsonProperty("altText")]  
[JsonPropertyName("altText")]  
public string AltText { get; set; }
```

Property Value

[string](#) 

CommentId

```
[YamlMember(Alias = "commentId")]  
[JsonProperty("commentId")]  
[JsonPropertyName("commentId")]
```

```
[MergeOption(MergeOption.Ignore)]  
public string CommentId { get; set; }
```

Property Value

[string](#)

LinkId

```
[YamlMember(Alias = "linkId")]  
[MergeOption(MergeOption.MergeKey)]  
[JsonProperty("linkId")]  
[JsonPropertyName("linkId")]  
public string LinkId { get; set; }
```

Property Value

[string](#)

LinkType

```
[YamlMember(Alias = "linkType")]  
[JsonProperty("linkType")]  
[JsonPropertyName("linkType")]  
[MergeOption(MergeOption.Ignore)]  
public LinkType LinkType { get; set; }
```

Property Value

[LinkType](#)

Class NamedArgumentInfo

Namespace: [Docfx.DataContracts.ManagedReference](#)

Assembly: Docfx.Dotnet.dll

```
public class NamedArgumentInfo
```

Inheritance

[object](#)  ← [NamedArgumentInfo](#)

Inherited Members

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

Properties

Name

```
[YamlMember(Alias = "name")]  
[JsonProperty("name")]  
[JsonPropertyName("name")]  
public string Name { get; set; }
```

Property Value

[string](#) 

Type

```
[YamlMember(Alias = "type")]  
[JsonProperty("type")]  
[JsonPropertyName("type")]  
public string Type { get; set; }
```

Property Value

[string](#)

Value

```
[YamlMember(Alias = "value")]  
[JsonProperty("value")]  
[JsonPropertyName("value")]  
public object Value { get; set; }
```

Property Value

[object](#)

Class PageViewModel

Namespace: [Docfx.DataContracts.ManagedReference](#)








Assembly: Docfx.Dotnet.dll

```
public class PageViewModel
```

Inheritance

[object](#)  ← [PageViewModel](#)

Inherited Members

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

Properties

Items

```
[YamlMember(Alias = "items")]  
[JsonProperty("items")]  
[JsonPropertyName("items")]  
public List<ItemViewModel> Items { get; set; }
```

Property Value

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

MemberLayout

```
[YamlMember(Alias = "memberLayout")]  
[JsonProperty("memberLayout")]  
[JsonPropertyName("memberLayout")]  
public MemberLayout MemberLayout { get; set; }
```

Property Value

[MemberLayout](#)

Metadata

```
[ExtensibleMember]  
[JsonExtensionData]  
[JsonExtensionData]  
public Dictionary<string, object> Metadata { get; set; }
```

Property Value

[Dictionary](#) <[string](#), [object](#)>

References

```
[YamlMember(Alias = "references")]  
[JsonProperty("references")]  
[JsonPropertyName("references")]  
public List<ReferenceViewModel> References { get; set; }
```

Property Value

[List](#) <ReferenceViewModel>

ShouldSkipMarkup

```
[YamlMember(Alias = "shouldSkipMarkup")]  
[JsonProperty("shouldSkipMarkup")]  
[JsonPropertyName("shouldSkipMarkup")]  
public bool ShouldSkipMarkup { get; set; }
```

Property Value

[bool](#)

Class SyntaxDetailViewModel

Namespace: [Docfx.DataContracts.ManagedReference](#)

Assembly: Docfx.Dotnet.dll

```
public class SyntaxDetailViewModel
```

Inheritance

[object](#)  ← [SyntaxDetailViewModel](#)

Inherited Members

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

Properties

Content

```
[YamlMember(Alias = "content")]  
[JsonProperty("content")]  
[JsonPropertyName("content")]  
public string Content { get; set; }
```

Property Value

[string](#) 

ContentForCSharp

```
[YamlIgnore]  
[JsonIgnore]  
[JsonIgnore]  
public string ContentForCSharp { get; set; }
```


Property Value

[string](#)

ContentForVB

```
[YamlIgnore]
[JsonIgnore]
[JsonIgnore]
public string ContentForVB { get; set; }
```

Property Value

[string](#)

Contents

```
[ExtensibleMember("content.")]
[JsonIgnore]
[JsonIgnore]
public SortedList<string, string> Contents { get; set; }
```

Property Value

[SortedList](#) <[string](#), [string](#)>

Parameters

```
[YamlMember(Alias = "parameters")]
[JsonProperty("parameters")]
[JsonPropertyName("parameters")]
public List<ApiParameter> Parameters { get; set; }
```

Property Value

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

Return

```
[YamlMember(Alias = "return")]  
[JsonProperty("return")]  
[JsonPropertyName("return")]  
public ApiParameter Return { get; set; }
```

Property Value

[ApiParameter](#)

TypeParameters

```
[YamlMember(Alias = "typeParameters")]  
[JsonProperty("typeParameters")]  
[JsonPropertyName("typeParameters")]  
public List<ApiParameter> TypeParameters { get; set; }
```

Property Value

[List](#)  [ApiParameter](#)

Enum LinkType

Namespace: [Docfx.DataContracts.ManagedReference](#)

Assembly: Docfx.Dotnet.dll

```
public enum LinkType
```

Fields

CRef = 0

HRef = 1

Enum MemberType

Namespace: [Docfx.DataContracts.ManagedReference](#)

Assembly: Docfx.Dotnet.dll

```
public enum MemberType
```

Fields

Assembly = 2

AttachedEvent = 16

AttachedProperty = 17

Class = 4

Constructor = 12

Container = 15

Default = 0

Delegate = 7

Enum = 8

Event = 11

Field = 9

Interface = 5

Method = 13

Namespace = 3

Operator = 14

Property = 10

Struct = 6

Toc = 1

Enum SyntaxLanguage

Namespace: [Docfx.DataContracts.ManagedReference](#)

Assembly: Docfx.Dotnet.dll

```
public enum SyntaxLanguage
```

Fields

```
CSharp = 1
```

```
Default = 0
```

```
VB = 2
```

Namespace Docfx.Dotnet

Classes

[DotnetApiCatalog](#)

Provides access to a .NET API definitions and their associated documentation.

[DotnetApiOptions](#)

Provides options to be used with `Docfx.Dotnet.DotnetApiCatalog.GenerateManagedReferenceYamlFiles(System.String,Docfx.Dotnet.DotnetApiOptions)`.

Enums

[SymbolIncludeState](#)

Return state of the `Docfx.Dotnet.DotnetApiOptions.IncludeApi` and `Docfx.Dotnet.DotnetApiOptions.IncludeAttribute` callbacks.

Class DotnetApiCatalog

Namespace: [Docfx.Dotnet](#)

Assembly: Docfx.Dotnet.dll

Provides access to a .NET API definitions and their associated documentation.

```
public static class DotnetApiCatalog
```

Inheritance

[object](#)  ← [DotnetApiCatalog](#)

Inherited Members

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

Methods

GenerateManagedReferenceYamlFiles(string)

Generates metadata reference YAML files using docfx.json config.

```
public static Task GenerateManagedReferenceYamlFiles(string configPath)
```

Parameters

configPath [string](#) 

The path to docfx.json config file.

Returns

[Task](#) 

A task to await for build completion.

GenerateManagedReferenceYamlFiles(string, DotnetApiOptions)

Generates metadata reference YAML files using docfx.json config.

```
public static Task GenerateManagedReferenceYamlFiles(string configPath, DotnetApiOption
```

Parameters

configPath [string](#) 

The path to docfx.json config file.

options [DotnetApiOptions](#)

Returns

[Task](#) 

A task to await for build completion.

Class DotnetApiOptions

Namespace: [Docfx.Dotnet](#)

Assembly: Docfx.Dotnet.dll








Provides options to be used with `Docfx.Dotnet.DotnetApiCatalog.GenerateManagedReferenceYamlFiles(System.String, Docfx.Dotnet.DotnetApiOptions)`.

```
public class DotnetApiOptions
```

Inheritance

[object](#)  ← [DotnetApiOptions](#)

Inherited Members

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

Properties

IncludeApi

Customizes the namespaces and types to include in the API catalog. Excluding a parent symbol exclude all child symbols underneath it.

```
public Func<ISymbol, SymbolIncludeState>? IncludeApi { get; init; }
```

Property Value

[Func](#)  <[ISymbol](#)  , [SymbolIncludeState](#)>?

IncludeAttribute

Customizes the attributes to include in the API catalog. Excluding a parent symbol exclude all child symbols underneath it.

```
public Func<ISymbol, SymbolIncludeState>? IncludeAttribute { get; init; }
```

Property Value

[Func](#) <[ISymbol](#), [SymbolIncludeState](#)>?

SourceUrl

Customizes the view source URL for files in a git repository. Returns `null` to use built-in support for GitHub, Azure Repos, etc.

```
public Func<GitSource, string?>? SourceUrl { get; init; }
```

Property Value

[Func](#) <[GitSource](#), [string](#)?>?

Enum SymbolIncludeState

Namespace: [Docfx.Dotnet](#)

Assembly: Docfx.Dotnet.dll

Return state of the Docfx.Dotnet.DotnetApiOptions.IncludeApi and Docfx.Dotnet.DotnetApiOptions.IncludeAttribute callbacks.

```
public enum SymbolIncludeState
```

Fields

Default = 0

Determines whether to include or not using the default configuration.

Exclude = 2

Ignores default rules and exclude the symbol from the API catalog.

Include = 1

Ignore default rules and include the symbol in the API catalog.