MDEdit

1.0.0

Generated by Doxygen 1.8.18

1 MDEdit - A Markdown source code editor with syntax highlighting and real-time preview. 1

1.1 Introduction 1

1.2 Getting started 1

2 Namespace Index 3

2.1 Packages 3

3 Hierarchical Index 5

3.1 Class Hierarchy 5

4 Class Index 7

4.1 Class List 7

5 Namespace Documentation 9

5.1 MDEdit Namespace Reference 9

6 Class Documentation 11

6.1 MDEdit.AsynchronousImageCache Class Reference 11

6.1.1 Detailed Description 11

6.1.2 Member Function Documentation 11

6.1.2.1 ImageUriResolverAsynchronous() 12

6.1.2.2 ImageUriResolverSynchronous() 12

6.1.2.3 SetExitEventHandler() 13

6.1.3 Event Documentation 13

6.1.3.1 CacheUpdated 13

6.2 MDEdit.CacheUpdatedEventArgs Class Reference 13

6.2.1 Detailed Description 14

6.2.2 Property Documentation 14

6.2.2.1 BaseImageUri 14

6.2.2.2 ImageUri 14

6.3 MDEdit.Editor Class Reference 14

6.3.1 Detailed Description 16

6.3.2 Member Enumeration Documentation 16

6.3.2.1 AccessTypes 16

6.3.3 Constructor & Destructor Documentation 17

6.3.3.1 Editor() 17

6.3.4 Member Function Documentation 17

6.3.4.1 Create() 17

6.3.4.2 Save() 17

6.3.4.3 SetText() [1/2] 18

6.3.4.4 SetText() [2/2] 18

6.3.5 Member Data Documentation 18

6.3.5.1 AutosaveInterval 18

6.3.5.2 PreviewTimeout 19

6.3.5.3 ShowLineChanges 19
6.3.5.4 ShowScrollbarOverview 19
6.3.5.5 SyntaxHighlighting 19
6.3.6 Property Documentation 19
6.3.6.1 AccessType 19
6.3.6.2 AutoSaveFile 20
6.3.6.3 Guid 20
6.3.6.4 KeepSaveHistory 20
6.3.6.5 MarkdownRenderer 20
6.3.6.6 SaveDirectory 20
6.3.6.7 Selection 21
6.3.6.8 SourceText 21
6.3.6.9 Text 21
6.3.6.10 TextChanged 21
6.3.7 Event Documentation 21
6.3.7.1 Autosave 21
6.3.7.2 PreviewRendered 22
6.3.7.3 SaveRequested 22
6.4 MDEdit.ImageRetrievalResult Struct Reference 22
6.4.1 Detailed Description 23
6.4.2 Constructor & Destructor Documentation 23
6.4.2.1 ImageRetrievalResult() 23
6.4.3 Member Function Documentation 23
6.4.3.1 operator() 23
6.4.4 Property Documentation 24
6.4.4.1 ImagePath 24
6.4.4.2 WasDownloaded 24
6.5 MDEdit.PreviewRenderedEventArgs Class Reference 24
6.5.1 Detailed Description 25
6.5.2 Property Documentation 25
6.5.2.1 Document 25
6.6 MDEdit.SaveEventArgs Class Reference 25
6.6.1 Detailed Description 26
6.6.2 Property Documentation 26
6.6.2.1 Text 26
6.7 MDEdit.Shortcut Class Reference 26
6.7.1 Detailed Description 27
6.7.2 Constructor & Destructor Documentation 27
6.7.2.1 Shortcut() 27
6.7.3 Property Documentation 27
6.7.3.1 Name 27
6.7.3.2 Shortcuts 27

Index **29**

Chapter 1

MDEdit - A Markdown source code editor with syntax highlighting and real-time preview.

1.1 Introduction

MDEdit is a Markdown source code editor control for Avalonia applications.

This library provides a control that can be added to Avalonia windows and integrates:

- A code editor with search/replace functions and syntax highlighting.
- A panel showing a preview of the document.
- A panel showing the save history of the file (that can persist across different sessions, if the application implements it properly).
- A panel with general settings.

It uses a modified version of CSharpEditor for the source code editor panel, and VectSharp.Markdown to render the Markdown document preview.

MDEdit is a .NET Standard 2.1 library, and should be usable in .NET Core 3.0+ and .NET 5.0+ projects. It is released under a GPLv3 licence. You can find the full documentation for this library at the documentation website. A PDF reference manual is also available.

1.2 Getting started

First of all, you need to install the NuGet package in your project.

The editor control cannot be added directly to the Window in XAML code, because it requires some non-trivial initialisation; you can create a new `Editor` control using the static method `Editor.Create` and then add it to the window:

```
using MDEdit;
// ...
    Editor editor = await Editor.Create();
    Grid grid = this.FindControl<Grid>("EditorContainer");
    grid.Children.Add(editor);
```

The first time an `Editor` control is added to your window may take some time to initialise; subsequent `Editor` controls will be created much faster.

The `Editor.Create` static method has multiple parameters, all of which are optional:

- `string initialText`: this is simply the initial source code that is shown in the control when it is created.
- `string guid`: this parameter provides an identifier for the control. This will be used, in particular, to store the save history of the file. If the control is initialised with the same `guid` across different sessions, the save history of the file will be restored.
- `Shortcut[] additionalShortcuts`: this makes it possible to display additional application-specific shortcuts in the shortcut section of the settings panel. Note that this does not actually implement the shortcut behaviour (which needs to be implemented separately by the developer) - it is simply provided so that users can open the settings panel and see all the shortcuts that can be used with the editor in the same place.

Take a look at the `MainWindow.xaml.cs file` in the MDEditDemo project to see how this works in practice.

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

MDEdit 9

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

MDEdit.AsynchronousImageCache 11
EventArgs
  MDEdit.CacheUpdatedEventArgs 13
  MDEdit.PreviewRenderedEventArgs 24
  MDEdit.SaveEventArgs 25
MDEdit.ImageRetrievalResult 22
MDEdit.Shortcut 26
UserControl
  MDEdit.Editor 14

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

MDEdit.AsynchronousImageCache
This class holds static methods for downloading image files from remote servers with an application-wide cache 11
MDEdit.CacheUpdatedEventArgs
Event data for the AsynchronousImageCache.CacheUpdated event 13
MDEdit.Editor
A C# source code editor for Avalonia 14
MDEdit.ImageRetrievalResult
Represents the result of an image retrieval request 22
MDEdit.PreviewRenderedEventArgs
A class to hold data for an event where the document preview has been rendered 24
MDEdit.SaveEventArgs
A class to hold data for an event where the user has requested to save the document 25
MDEdit.Shortcut
Represents a keyboard shortcut 26

Chapter 5

Namespace Documentation

5.1 MDEdit Namespace Reference

Classes

- class AsynchronousImageCache

  This class holds static methods for downloading image files from remote servers with an application-wide cache.
- class CacheUpdatedEventArgs

  Event data for the AsynchronousImageCache.CacheUpdated event.
- class Editor

  A C# source code editor for Avalonia.
- struct ImageRetrievalResult

  Represents the result of an image retrieval request.
- class PreviewRenderedEventArgs

  A class to hold data for an event where the document preview has been rendered.
- class SaveEventArgs

  A class to hold data for an event where the user has requested to save the document.
- class Shortcut

  Represents a keyboard shortcut.

Chapter 6

Class Documentation

6.1 MDEdit.AsynchronousImageCache Class Reference

This class holds static methods for downloading image files from remote servers with an application-wide cache.

Static Public Member Functions

- static void SetExitEventHandler ()

  This method should be invoked at some point before the application exits; it ensures that the image cache folder is cleared when the application is closed.
- static ImageRetrievalResult ImageUriResolverAsynchronous (string imageUri, string baseUriString)

  Resolves an image Uri asynchronously. If the image has already been downloaded previously and is available in the cache, its path on disk is returned immediately. Otherwise, it is queued for download and when it becomes available, the CacheUpdated event is invoked. If an image is requested again while it is being downloaded, a new download of the image is prevented.
- static ImageRetrievalResult ImageUriResolverSynchronous (string imageUri, string baseUriString)

  Resolves an image Uri synchronously. If the image has already been downloaded previously and is available in the cache, its path on disk is returned immediately. Otherwise, this method blocks until the image is downloaded. If an image is requested by this method while it is being downloaded as a result of a call to ImageUriResolverAsynchronous(string, string), it is downloaded a second time.

Events

- static EventHandler< CacheUpdatedEventArgs > CacheUpdated

  An event that is invoked when an image that was requested asynchronously becomes available.

6.1.1 Detailed Description

This class holds static methods for downloading image files from remote servers with an application-wide cache.

Definition at line 30 of file AsynchronousImageCache.cs.

6.1.2 Member Function Documentation

6.1.2.1 ImageUriResolverAsynchronous()

```
static ImageRetrievalResult MDEdit.AsynchronousImageCache.ImageUriResolverAsynchronous (
            string imageUri,
            string baseUriString )  [static]
```

Resolves an image Uri asynchronously. If the image has already been downloaded previously and is available in the cache, its path on disk is returned immediately. Otherwise, it is queued for download and when it becomes available, the CacheUpdated event is invoked. If an image is requested again while it is being downloaded, a new download of the image is prevented.

Parameters

imageUri	The Uri of the image to download, either absolute or relative to the *baseUriString* .
baseUriString	The base Uri for resolving the *imageUri* .

Returns

An ImageRetrievalResult containing the path to a copy of the image file on disk. If the ImageRetrievalResult.ImagePath property of the return value is `null`, the image was not available in the cache and has been queued for download.

Definition at line 103 of file AsynchronousImageCache.cs.

6.1.2.2 ImageUriResolverSynchronous()

```
static ImageRetrievalResult MDEdit.AsynchronousImageCache.ImageUriResolverSynchronous (
            string imageUri,
            string baseUriString )  [static]
```

Resolves an image Uri synchronously. If the image has already been downloaded previously and is available in the cache, its path on disk is returned immediately. Otherwise, this method blocks until the image is downloaded. If an image is requested by this method while it is being downloaded as a result of a call to ImageUriResolverAsynchronous(string, string), it is downloaded a second time.

Parameters

imageUri	The Uri of the image to download, either absolute or relative to the *baseUriString* .
baseUriString	The base Uri for resolving the *imageUri* .

Returns

An ImageRetrievalResult containing the path to a copy of the image file on disk. If the ImageRetrievalResult.ImagePath property of the return value is `null`, an error occurred while the image was being accessed (e.g. it was not found on the server).

Definition at line 186 of file AsynchronousImageCache.cs.

6.1.2.3 SetExitEventHandler()

```
static void MDEdit.AsynchronousImageCache.SetExitEventHandler ( )  [static]
```

This method should be invoked at some point before the application exits; it ensures that the image cache folder is cleared when the application is closed.

Definition at line 61 of file AsynchronousImageCache.cs.

6.1.3 Event Documentation

6.1.3.1 CacheUpdated

```
EventHandler<CacheUpdatedEventArgs> MDEdit.AsynchronousImageCache.CacheUpdated  [static]
```

An event that is invoked when an image that was requested asynchronously becomes available.

Definition at line 45 of file AsynchronousImageCache.cs.

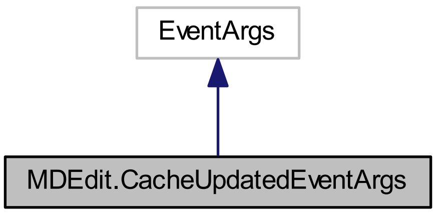
The documentation for this class was generated from the following file:

- MDEdit/AsynchronousImageCache.cs

6.2 MDEdit.CacheUpdatedEventArgs Class Reference

Event data for the AsynchronousImageCache.CacheUpdated event.

Inheritance diagram for MDEdit.CacheUpdatedEventArgs:



Properties

- string BaseImageUri `[get]`

  The base image Uri of the image that has been resolved.
- string ImageUri `[get]`

  The Uri of the image that has been resolved (either absolute, or relative to the BaseImageUri).

6.2.1 Detailed Description

Event data for the AsynchronousImageCache.CacheUpdated event.

Definition at line 270 of file AsynchronousImageCache.cs.

6.2.2 Property Documentation

6.2.2.1 BaseImageUri

```
string MDEdit.CacheUpdatedEventArgs.BaseImageUri  [get]
```

The base image Uri of the image that has been resolved.

Definition at line 275 of file AsynchronousImageCache.cs.

6.2.2.2 ImageUri

```
string MDEdit.CacheUpdatedEventArgs.ImageUri  [get]
```

The Uri of the image that has been resolved (either absolute, or relative to the BaseImageUri).

Definition at line 280 of file AsynchronousImageCache.cs.

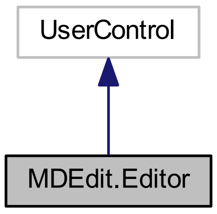
The documentation for this class was generated from the following file:

- MDEdit/AsynchronousImageCache.cs

6.3 MDEdit.Editor Class Reference

A C# source code editor for Avalonia.

Inheritance diagram for MDEdit.Editor:



Public Types

- enum AccessTypes { AccessTypes.ReadWrite, AccessTypes.ReadOnlyWithHistory, AccessTypes.ReadOnly }

  Describes the actions that the user can perform on the code.

Public Member Functions

- Editor ()

  Public constructor. This is only provided for compatibility with Avalonia (`see issue #2593`). Please use Editor.Create instead.
- async Task SetText (string text)

  Sets the text of the document.
- async Task SetText (SourceText text)

  Sets the text of the document.
- void Save ()

  Add the current text of the document to the save history (if enabled) and invoke the SaveRequested event.

Static Public Member Functions

- static async Task< Editor > Create (string initialText="", string guid=null, Shortcut[ ] additionalShortcuts=null)

  Create a new Editor instance.

Public Attributes

- bool SyntaxHighlighting => this.EditorControl.SyntaxHighlighting

  A boolean value indicating whether syntax highlighting is enabled.
- bool ShowLineChanges => this.EditorControl.ShowLineChanges

  A boolean value indicating whether changed lines are highlighted on the left side of the control.
- bool ShowScrollbarOverview => this.EditorControl.ShowScrollbarOverview

  A boolean value indicating whether a summary of the changed lines, errors/warning, search results, breakpoints and the position of the caret should be shown over the vertical scrollbar.
- int AutosaveInterval => this.AutoSaver.MillisecondsInterval

  The timeout between consecutive autosaves, in milliseconds.
- int PreviewTimeout => this.CompilationErrorChecker.MillisecondsInterval

  The timeout for updating the preview after the user stops typing, in milliseconds.

Properties

- EventHandler< EventArgs > TextChanged

  Event raised when the document text is changed.
- string Text `[get]`

  The source code of the document as a string.
- SourceText SourceText `[get]`

  The source code of the document as a SourceText.
- MarkdownRenderer MarkdownRenderer `[get]`

  The MarkdownRenderer used to display the document preview.
- AccessTypes AccessType `[get, set]`

  Determines whether the text of the document can be edited by the user.

- string Guid `[get]`

  A unique identifier for the document being edited.
- string SaveDirectory `[get]`

  The full path to the directory where the autosave file and the save history for the current document are kept.
- string AutoSaveFile `[get]`

  The full path to the autosave file.
- bool KeepSaveHistory = true `[get]`

  A boolean value indicating whether a history of the saved versions of the document is kept.
- TextSpan Selection `[get, set]`

  Gets or sets the selected text span.

Events

- EventHandler< SaveEventArgs > SaveRequested

  Event raised when the user uses the keyboard shortcut or presses the button to save the document.
- EventHandler< SaveEventArgs > Autosave

  Event raised when the document is automatically saved.
- EventHandler< PreviewRenderedEventArgs > PreviewRendered

  Event raised when the rendering of the document preview completes.

6.3.1 Detailed Description

A C# source code editor for Avalonia.

Definition at line 40 of file Editor.axaml.cs.

6.3.2 Member Enumeration Documentation

6.3.2.1 AccessTypes

`enum MDEdit.Editor.AccessTypes [strong]`

Describes the actions that the user can perform on the code.

Enumerator

ReadWrite	The code can be edited freely.
ReadOnlyWithHistory	The code cannot be edited, but the user can load previous versions of the file.
ReadOnly	The code can only be read. No advanced features are provided beyond syntax highlighting.

Definition at line 99 of file Editor.public.cs.

6.3.3 Constructor & Destructor Documentation

6.3.3.1 Editor()

```
MDEdit.Editor.Editor ( )
```

Public constructor. This is only provided for compatibility with Avalonia (`see issue #2593`). Please use Editor.Create instead.

Definition at line 97 of file Editor.axaml.cs.

6.3.4 Member Function Documentation

6.3.4.1 Create()

```
static async Task<Editor> MDEdit.Editor.Create (
            string initialText = "",
            string guid = null,
            Shortcut[] additionalShortcuts = null )  [static]
```

Create a new Editor instance.

Parameters

initialText	The initial text of the editor.
guid	A unique identifier for the document being edited. If this is `null`, a new System.Guid is generated. If the same identifier is used multiple times, the save history of the document will be available, even if the application has been closed between different sessions.
additionalShortcuts	Additional application-specific shortcuts (for display purposes only - you need to implement your own logic).

Returns

A fully initialised Editor instance.

Definition at line 220 of file Editor.public.cs.

6.3.4.2 Save()

```
void MDEdit.Editor.Save ( )
```

Add the current text of the document to the save history (if enabled) and invoke the SaveRequested event.

Definition at line 267 of file Editor.public.cs.

6.3.4.3 SetText() [1/2]

```
async Task MDEdit.Editor.SetText (
            SourceText text )
```

Sets the text of the document.

Parameters

text	The new text of the document.

Returns

A Task that completes when the text has been updated.

Definition at line 259 of file Editor.public.cs.

6.3.4.4 SetText() [2/2]

```
async Task MDEdit.Editor.SetText (
            string text )
```

Sets the text of the document.

Parameters

text	The new text of the document.

Returns

A Task that completes when the text has been updated.

Definition at line 249 of file Editor.public.cs.

6.3.5 Member Data Documentation

6.3.5.1 AutosaveInterval

```
int MDEdit.Editor.AutosaveInterval => this.AutoSaver.MillisecondsInterval
```

The timeout between consecutive autosaves, in milliseconds.

Definition at line 190 of file Editor.public.cs.

6.3.5.2 PreviewTimeout

```
int MDEdit.Editor.PreviewTimeout => this.CompilationErrorChecker.MillisecondsInterval
```

The timeout for updating the preview after the user stops typing, in milliseconds.

Definition at line 195 of file Editor.public.cs.

6.3.5.3 ShowLineChanges

```
bool MDEdit.Editor.ShowLineChanges => this.EditorControl.ShowLineChanges
```

A boolean value indicating whether changed lines are highlighted on the left side of the control.

Definition at line 180 of file Editor.public.cs.

6.3.5.4 ShowScrollbarOverview

```
bool MDEdit.Editor.ShowScrollbarOverview => this.EditorControl.ShowScrollbarOverview
```

A boolean value indicating whether a summary of the changed lines, errors/warning, search results, breakpoints and the position of the caret should be shown over the vertical scrollbar.

Definition at line 185 of file Editor.public.cs.

6.3.5.5 SyntaxHighlighting

```
bool MDEdit.Editor.SyntaxHighlighting => this.EditorControl.SyntaxHighlighting
```

A boolean value indicating whether syntax highlighting is enabled.

Definition at line 175 of file Editor.public.cs.

6.3.6 Property Documentation

6.3.6.1 AccessType

```
AccessTypes MDEdit.Editor.AccessType  [get], [set]
```

Determines whether the text of the document can be edited by the user.

Definition at line 122 of file Editor.public.cs.

6.3.6.2 AutoSaveFile

```
string MDEdit.Editor.AutoSaveFile  [get]
```

The full path to the autosave file.

Definition at line 165 of file Editor.public.cs.

6.3.6.3 Guid

```
string MDEdit.Editor.Guid  [get]
```

A unique identifier for the document being edited.

Definition at line 155 of file Editor.public.cs.

6.3.6.4 KeepSaveHistory

```
bool MDEdit.Editor.KeepSaveHistory = true  [get]
```

A boolean value indicating whether a history of the saved versions of the document is kept.

Definition at line 170 of file Editor.public.cs.

6.3.6.5 MarkdownRenderer

```
MarkdownRenderer MDEdit.Editor.MarkdownRenderer  [get]
```

The MarkdownRenderer used to display the document preview.

Definition at line 94 of file Editor.public.cs.

6.3.6.6 SaveDirectory

```
string MDEdit.Editor.SaveDirectory  [get]
```

The full path to the directory where the autosave file and the save history for the current document are kept.

Definition at line 160 of file Editor.public.cs.

6.3.6.7 Selection

```
TextSpan MDEdit.Editor.Selection  [get], [set]
```

Gets or sets the selected text span.

Definition at line 200 of file Editor.public.cs.

6.3.6.8 SourceText

```
SourceText MDEdit.Editor.SourceText  [get]
```

The source code of the document as a SourceText.

Definition at line 82 of file Editor.public.cs.

6.3.6.9 Text

```
string MDEdit.Editor.Text  [get]
```

The source code of the document as a string.

Definition at line 70 of file Editor.public.cs.

6.3.6.10 TextChanged

```
EventHandler<EventArgs> MDEdit.Editor.TextChanged  [add], [remove]
```

Event raised when the document text is changed.

Definition at line 54 of file Editor.public.cs.

6.3.7 Event Documentation

6.3.7.1 Autosave

```
EventHandler<SaveEventArgs> MDEdit.Editor.Autosave
```

Event raised when the document is automatically saved.

Definition at line 44 of file Editor.public.cs.

6.3.7.2 PreviewRendered

`EventHandler<PreviewRenderedEventArgs> MDEdit.Editor.PreviewRendered`

Event raised when the rendering of the document preview completes.

Definition at line 49 of file Editor.public.cs.

6.3.7.3 SaveRequested

`EventHandler<SaveEventArgs> MDEdit.Editor.SaveRequested`

Event raised when the user uses the keyboard shortcut or presses the button to save the document.

Definition at line 39 of file Editor.public.cs.

The documentation for this class was generated from the following files:

- MDEdit/Editor.axaml.cs
- MDEdit/Editor.public.cs

6.4 MDEdit.ImageRetrievalResult Struct Reference

Represents the result of an image retrieval request.

Public Member Functions

- ImageRetrievalResult (string imagePath, bool wasDownloaded)

  Creates a new ImageRetrievalResult object.

Static Public Member Functions

- static implicit operator (string, bool)(ImageRetrievalResult result)

  Converts a ImageRetrievalResult into a tuple.

Properties

- string ImagePath `[get]`

  The path to a file on disk containing the image. The file will have an appropriate extension based on the image file.
- bool WasDownloaded `[get]`

  This value is set to `true` if the image file was downloaded from a remote server and saved as a temporary file which may be deleted after the consuming code is done with it.

6.4.1 Detailed Description

Represents the result of an image retrieval request.

Definition at line 231 of file AsynchronousImageCache.cs.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 ImageRetrievalResult()

```
MDEdit.ImageRetrievalResult.ImageRetrievalResult (
            string imagePath,
            bool wasDownloaded )
```

Creates a new ImageRetrievalResult object.

Parameters

imagePath	The path to the file on disk containing the image.
wasDownloaded	Set to `true` if the image file was downloaded from a remote server and may be deleted after the consuming code is done with it.

Definition at line 248 of file AsynchronousImageCache.cs.

6.4.3 Member Function Documentation

6.4.3.1 operator()

```
static implicit MDEdit.ImageRetrievalResult.operator (
            string ,
            bool  )  [static]
```

Converts a ImageRetrievalResult into a tuple.

Parameters

result	The ImageRetrievalResult to convert.

Returns

A tuple containing the ImagePath and WasDownloaded properties of the *result* .

Definition at line 261 of file AsynchronousImageCache.cs.

6.4.4 Property Documentation

6.4.4.1 ImagePath

```
string MDEdit.ImageRetrievalResult.ImagePath  [get]
```

The path to a file on disk containing the image. The file will have an appropriate extension based on the image file.

Definition at line 236 of file AsynchronousImageCache.cs.

6.4.4.2 WasDownloaded

```
bool MDEdit.ImageRetrievalResult.WasDownloaded  [get]
```

This value is set to `true` if the image file was downloaded from a remote server and saved as a temporary file which may be deleted after the consuming code is done with it.

Definition at line 241 of file AsynchronousImageCache.cs.

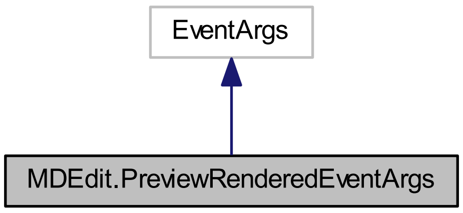
The documentation for this struct was generated from the following file:

- MDEdit/AsynchronousImageCache.cs

6.5 MDEdit.PreviewRenderedEventArgs Class Reference

A class to hold data for an event where the document preview has been rendered.

Inheritance diagram for MDEdit.PreviewRenderedEventArgs:



Properties

- Markdig.Syntax.MarkdownDocument Document `[get]`

  The Markdown document that has been rendered.

6.5.1 Detailed Description

A class to hold data for an event where the document preview has been rendered.

Definition at line 307 of file Editor.public.cs.

6.5.2 Property Documentation

6.5.2.1 Document

`Markdig.Syntax.MarkdownDocument MDEdit.PreviewRenderedEventArgs.Document  [get]`

The Markdown document that has been rendered.

Definition at line 312 of file Editor.public.cs.

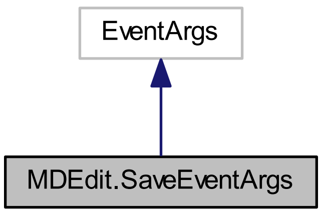
The documentation for this class was generated from the following file:

- MDEdit/Editor.public.cs

6.6 MDEdit.SaveEventArgs Class Reference

A class to hold data for an event where the user has requested to save the document.

Inheritance diagram for MDEdit.SaveEventArgs:



Properties

- string Text `[get]`

    The text of the document to save.

6.6.1 Detailed Description

A class to hold data for an event where the user has requested to save the document.

Definition at line 291 of file Editor.public.cs.

6.6.2 Property Documentation

6.6.2.1 Text

```
string MDEdit.SaveEventArgs.Text  [get]
```

The text of the document to save.

Definition at line 296 of file Editor.public.cs.

The documentation for this class was generated from the following file:

- MDEdit/Editor.public.cs

6.7 MDEdit.Shortcut Class Reference

Represents a keyboard shortcut.

Public Member Functions

- Shortcut (string name, string[ ][ ] shortcuts)

    Creates a new Shortcut instance.

Properties

- string Name `[get]`

    The name of the action performed by the shortcut.
- string[ ][ ] Shortcuts `[get]`

    The keys that have to be pressed together to perform the action.

6.7.1 Detailed Description

Represents a keyboard shortcut.

Definition at line 323 of file Editor.public.cs.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 Shortcut()

```
MDEdit.Shortcut.Shortcut (
             string name,
             string shortcuts[ ][ ] )
```

Creates a new Shortcut instance.

Parameters

name	The name of the action performed by the shortcut (e.g. "Copy").
shortcuts	The keys that have to be pressed together to perform the action (e.g. [["Ctrl", "C"], ["Ctrl", "Ins"]] to specify that either `Ctrl+C` or `Ctrl+Ins` can be used. "Ctrl" will automatically be converted to "Cmd" on macOS.

Definition at line 340 of file Editor.public.cs.

6.7.3 Property Documentation

6.7.3.1 Name

```
string MDEdit.Shortcut.Name  [get]
```

The name of the action performed by the shortcut.

Definition at line 328 of file Editor.public.cs.

6.7.3.2 Shortcuts

```
string [ ][ ] MDEdit.Shortcut.Shortcuts  [get]
```

The keys that have to be pressed together to perform the action.

Definition at line 333 of file Editor.public.cs.

The documentation for this class was generated from the following file:

- MDEdit/Editor.public.cs

Index

AccessType
    MDEdit.Editor, 19
AccessTypes
    MDEdit.Editor, 16
Autosave
    MDEdit.Editor, 21
AutoSaveFile
    MDEdit.Editor, 19
AutosaveInterval
    MDEdit.Editor, 18

BaseImageUri
    MDEdit.CacheUpdatedEventArgs, 14

CacheUpdated
    MDEdit.AsynchronousImageCache, 13
Create
    MDEdit.Editor, 17

Document
    MDEdit.PreviewRenderedEventArgs, 25

Editor
    MDEdit.Editor, 17

Guid
    MDEdit.Editor, 20

ImagePath
    MDEdit.ImageRetrievalResult, 24
ImageRetrievalResult
    MDEdit.ImageRetrievalResult, 23
ImageUri
    MDEdit.CacheUpdatedEventArgs, 14
ImageUriResolverAsynchronous
    MDEdit.AsynchronousImageCache, 11
ImageUriResolverSynchronous
    MDEdit.AsynchronousImageCache, 12

KeepSaveHistory
    MDEdit.Editor, 20

MarkdownRenderer
    MDEdit.Editor, 20
MDEdit, 9
MDEdit.AsynchronousImageCache, 11
    CacheUpdated, 13
    ImageUriResolverAsynchronous, 11
    ImageUriResolverSynchronous, 12
    SetExitEventHandler, 12
MDEdit.CacheUpdatedEventArgs, 13
    BaseImageUri, 14
    ImageUri, 14
MDEdit.Editor, 14
    AccessType, 19
    AccessTypes, 16
    Autosave, 21
    AutoSaveFile, 19
    AutosaveInterval, 18
    Create, 17
    Editor, 17
    Guid, 20
    KeepSaveHistory, 20
    MarkdownRenderer, 20
    PreviewRendered, 21
    PreviewTimeout, 18
    ReadOnly, 16
    ReadOnlyWithHistory, 16
    ReadWrite, 16
    Save, 17
    SaveDirectory, 20
    SaveRequested, 22
    Selection, 20
    SetText, 17, 18
    ShowLineChanges, 19
    ShowScrollbarOverview, 19
    SourceText, 21
    SyntaxHighlighting, 19
    Text, 21
    TextChanged, 21
MDEdit.ImageRetrievalResult, 22
    ImagePath, 24
    ImageRetrievalResult, 23
    operator, 23
    WasDownloaded, 24
MDEdit.PreviewRenderedEventArgs, 24
    Document, 25
MDEdit.SaveEventArgs, 25
    Text, 26
MDEdit.Shortcut, 26
    Name, 27
    Shortcut, 27
    Shortcuts, 27

Name
    MDEdit.Shortcut, 27

operator
    MDEdit.ImageRetrievalResult, 23

PreviewRendered

MDEdit.Editor, 21
PreviewTimeout
MDEdit.Editor, 18

ReadOnly
MDEdit.Editor, 16
ReadOnlyWithHistory
MDEdit.Editor, 16
ReadWrite
MDEdit.Editor, 16

Save
MDEdit.Editor, 17
SaveDirectory
MDEdit.Editor, 20
SaveRequested
MDEdit.Editor, 22
Selection
MDEdit.Editor, 20
SetExitEventHandler
MDEdit.AsynchronousImageCache, 12
SetText
MDEdit.Editor, 17, 18
Shortcut
MDEdit.Shortcut, 27
Shortcuts
MDEdit.Shortcut, 27
ShowLineChanges
MDEdit.Editor, 19
ShowScrollbarOverview
MDEdit.Editor, 19
SourceText
MDEdit.Editor, 21
SyntaxHighlighting
MDEdit.Editor, 19

Text
MDEdit.Editor, 21
MDEdit.SaveEventArgs, 26
TextChanged
MDEdit.Editor, 21

WasDownloaded
MDEdit.ImageRetrievalResult, 24