MarkDown-Module

Markdown - Module

- **Gadget**
- **Message Requester**
- **Help Window**

Overview

The gadget can display text formatted with the MarkDown Syntax. Markdown[1] is a lightweight MarkUp language that you can use to add formatting elements to plaintext text documents.

- Markdown Support
- Supported Emojis

[1] Created by John Gruber in 2004.

Markdown Support

Basic Syntax

- Headings
- Emphasis: Bold / Italic / Code
- Blockquotes
- ordered and unordered Lists
- Horizontal Rules
- Links and Autolinks
- Images

Extended Syntax

- Task Lists
- Definition Lists
- Tables
- Fenced code Blocks
- Footnotes
- Strikethrough
- Highlight
- Underline (or Insert)
- Subscript and Superscript
- Abbreviation
- Emoji (supported)

Supported Emojis

📞 telephone receiver
✉ envelope
📅 calendar
📝 memo
✏ pencil
🔖 bookmark
💡 bulb
🔍 magnifier
📎 paperclip
⚠ warning

😃 grinning face with big eyes
🙂 slightly smiling face
😏 smirking face
😎 smiling face with sunglasses
🙁 slightly frowning face
😠 angry face
😟 worried face
😉 winking face
🤣 rolling on the floor laughing
🙄 face with rolling eyes

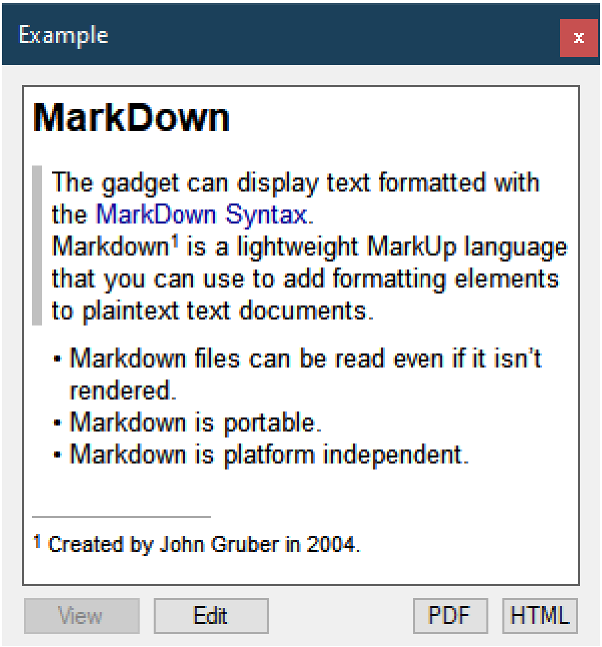
Markdown - Gadget

Syntax:

```
Gadget(GNum.i, X.i, Y.i, Width.i, Height.i,
Flags.i=#False, WindowNum.i=#PB_Default)
```

Flags:

`#AutoResize` - *automatic resizing of the gadget*
`#Borderless` - *draw no gadget frame*



MarkDown

> The gadget can display text formatted with the MarkDown Syntax.
> Markdown[1] is a lightweight MarkUp language that you can use to add formatting elements to plaintext text documents.

- Markdown files can be read even if it isn't rendered.
- Markdown is portable.
- Markdown is platform independent.

[1] Created by John Gruber in 2004.

Markdown Text

Show Markdown

Adding or changing markdown text.

Syntax:

```
SetText(GNum.i, Text.s)
```

Export Markdown

Export displayed text as HTML or PDF.

Syntax:

```
Export(GNum.i, Type.i, File.s="", Title.s="")
```

Type:

`#HTML` - *Export as HTML*
`#PDF` - *Export as PDF*

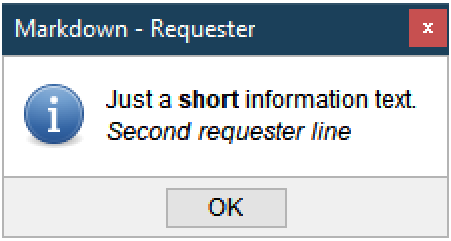
Markdown - Requester

Syntax:

```
Requester(Title.s, Text.s, Flags.i=#False,
Parent.i=#PB_Default)
```

Flags:

- `#YesNo` / `#YesNoCancel` - *use these buttons (otherwise OK)*
- `#Info` / `#Question` / `#Error`/ `#Warning` - *show this symbol*



Just a **short** information text.
Second requester line

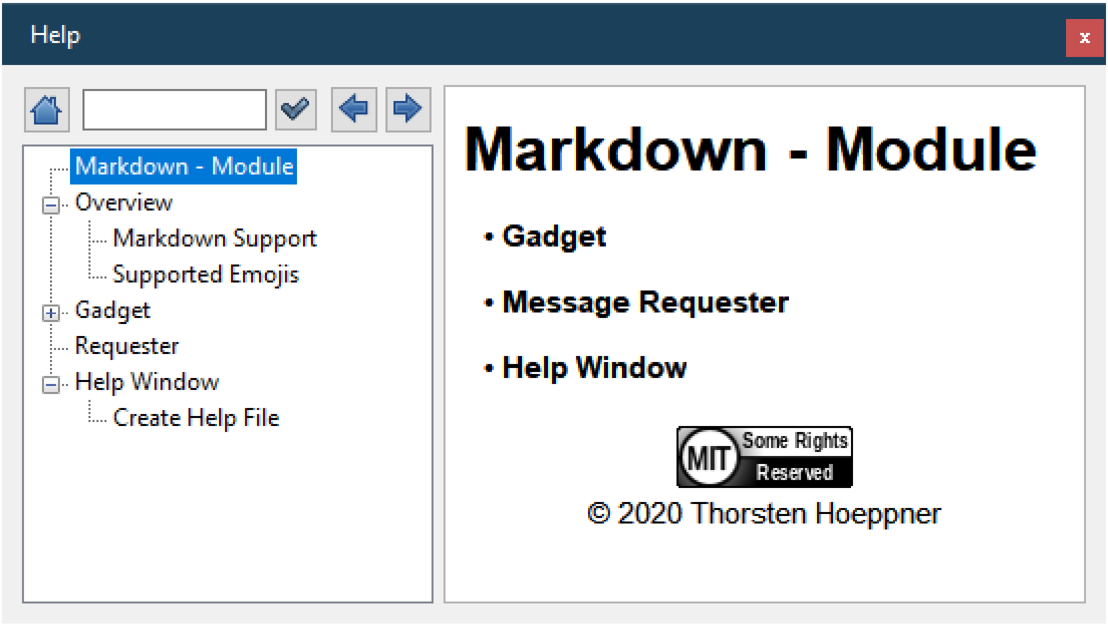
Markdown - Help

Syntax:

```
Help(Title.s, File.s, Label.s="", Flags.i=#False,
Parent.i=#PB_Default)
```

Flags:

`#AutoResize` - *resizing of the window and the gadget*



Markdown - Module

- **Gadget**
- **Message Requester**
- **Help Window**

MIT Some Rights Reserved

© 2020 Thorsten Hoeppner

Create Help File

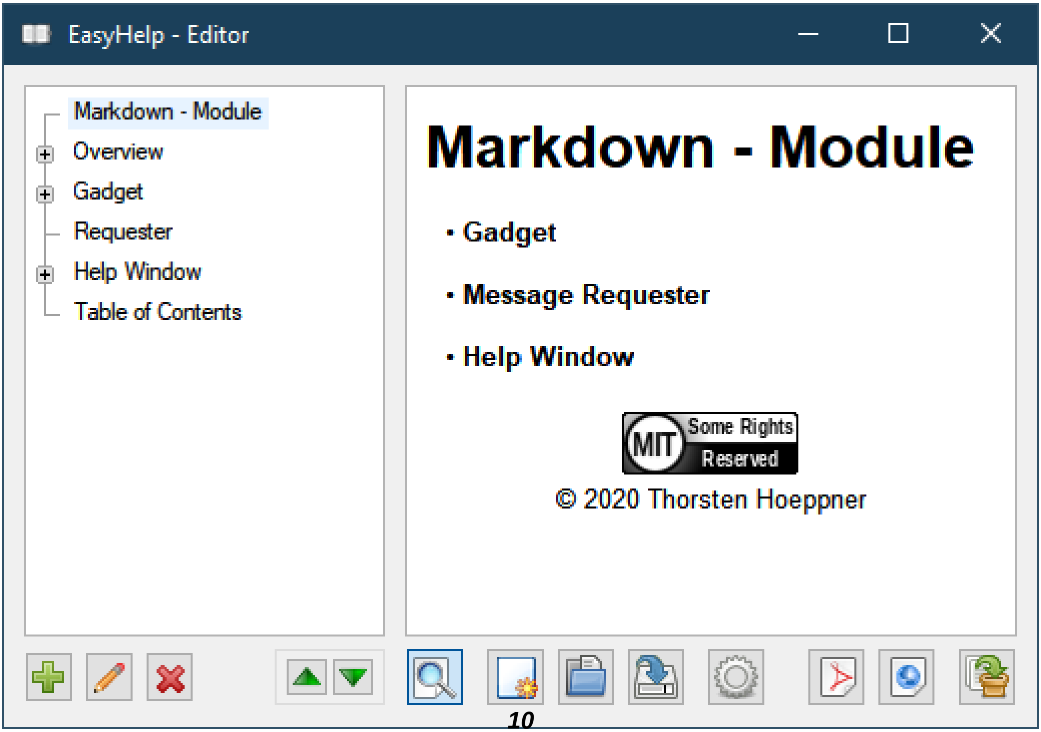
Module Commands

Man kann eine Hilfedatei mit den im Modul verfügbaren Befehlen erstellen:

```
If CreateHelp("Help.mdh")
  AddHelpItem("Title 1", "Text 1", "Label 1", 0)
  AddHelpItem("Title 2", "Text 2", "Label 2", 0)
  CloseHelp()
EndIf
```

EasyHelpEditor

It is easier and more comfortable with the EasyHelp-Editor program: Download



Markdown - Module

- Gadget
- Message Requester
- Help Window

MIT Some Rights Reserved

© 2020 Thorsten Hoeppner

Table of Contents