Blueprint Markdown

User Guide

Version 1.0

For Unreal Engine 5.5+

Introduction

Blueprint Markdown Documentation is a plugin for Unreal Engine that lets you create and manage documentation for your Blueprints and other assets directly inside the editor. With its easy-to-use visual editor, you can write rich documentation without needing to learn markdown syntax.
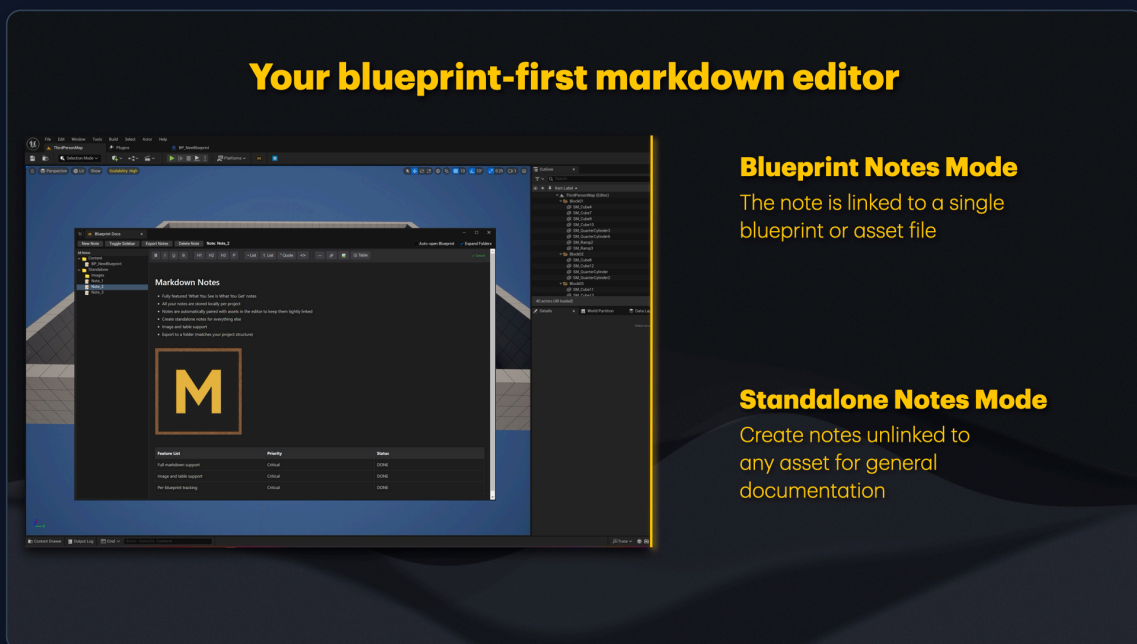

Figure 1: The Blueprint Markdown editor integrated into Unreal Engine

The editor provides a familiar word processor-style interface that automatically saves your work, tracks which Blueprint you're working on, and stores everything as portable markdown files that work with version control systems.

Key Features

🎨 Visual Editor

Write documentation using a familiar word processor-style interface. What you see is what you get!

📝 Rich Formatting

Headings, bold, italic, code blocks, tables, images, links, and more.

🔄 Auto-Save

🎯 Asset Linking

Your changes are saved automatically as you type. Never lose your work!

Documentation automatically associates with the Blueprint you're working on.

🌙 Dark Theme

Matches Unreal Engine's editor theme for a seamless experience.

💾 Local Storage

All notes stored in your project. Works offline, no cloud required.

📊 Table Editor

Create and edit tables visually with add/remove row and column controls.

🖼️ Image Support

Insert images from your computer and resize them by dragging.

Getting Started

Installation

1   **Download from Fab**

Purchase and download the Blueprint Markdown plugin from the Fab marketplace. The plugin will be installed automatically to your engine plugins folder.

2   **Launch Unreal Editor**

Open your project. The plugin will be available immediately.

3   **Enable the Plugin (if needed)**

Go to **Edit → Plugins**, search for "Blueprint Markdown", and ensure it's enabled.

4   **Restart the Editor**

If you enabled the plugin manually, restart the editor for changes to take effect.

> **That's it!** You're ready to start documenting your Blueprints.

Opening the Documentation Window

You can open the Blueprint Documentation window in three ways:

Method	Location
Window Menu	Window → Blueprint Documentation
Level Editor Toolbar	Click the "Blueprint Docs" button in the toolbar
Blueprint Editor Toolbar	Click "Blueprint Docs" when editing a Blueprint

The Editor Interface

The documentation window provides a complete editing environment with multiple panels and tools for managing your documentation.
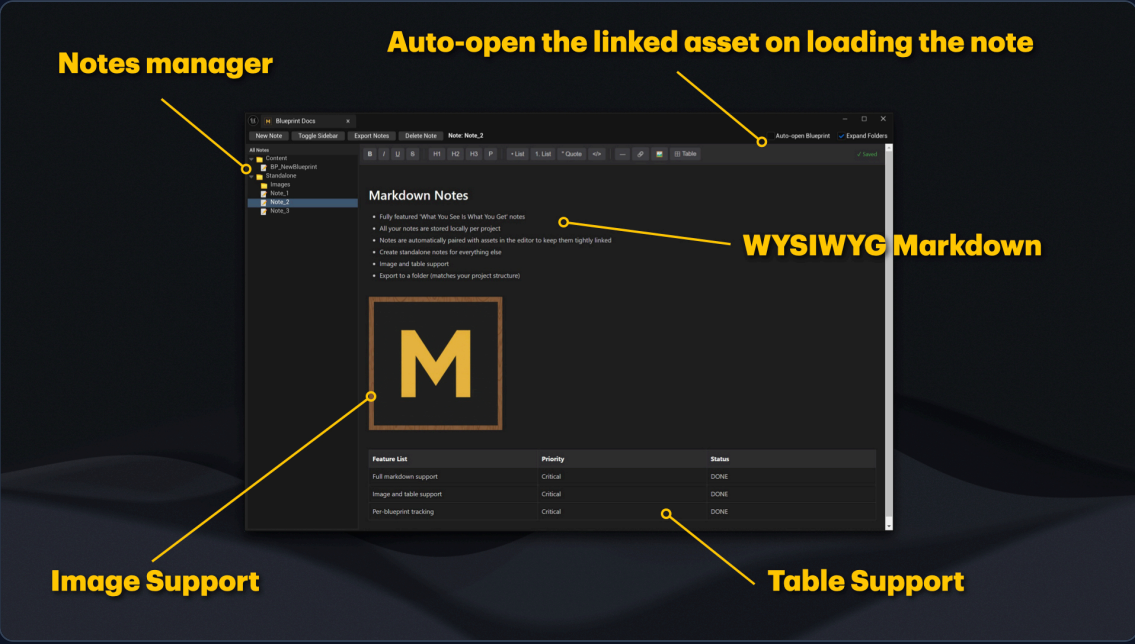

Figure 2: Overview of the editor interface showing the toolbar, sidebar, and editing area

Window Layout

The documentation window has three main areas:

1. Top Toolbar

Button	Function
New Note	Create a new standalone note
Toggle Sidebar	Show or hide the file browser sidebar
Export Notes	Export all documentation to an external folder
Delete Note	Delete the current note (with confirmation)
Auto-open Blueprint	Automatically open the Blueprint when selecting its note

Expand Folders	Expand all folders in the sidebar by default

2. Sidebar (File Browser)

The left sidebar shows all your notes organized by folder:

- **Content/** - Notes attached to assets, mirroring your project structure
- **Standalone/** - Notes not attached to any asset

Click any note to open it. Double-click to open the associated Blueprint.

3. Editor Area

The main editing area where you write and format your documentation.

Working Side by Side with Blueprints

One of the most powerful features of Blueprint Markdown is the ability to work on your documentation while viewing your Blueprint at the same time. This makes it easy to document exactly what you're building.
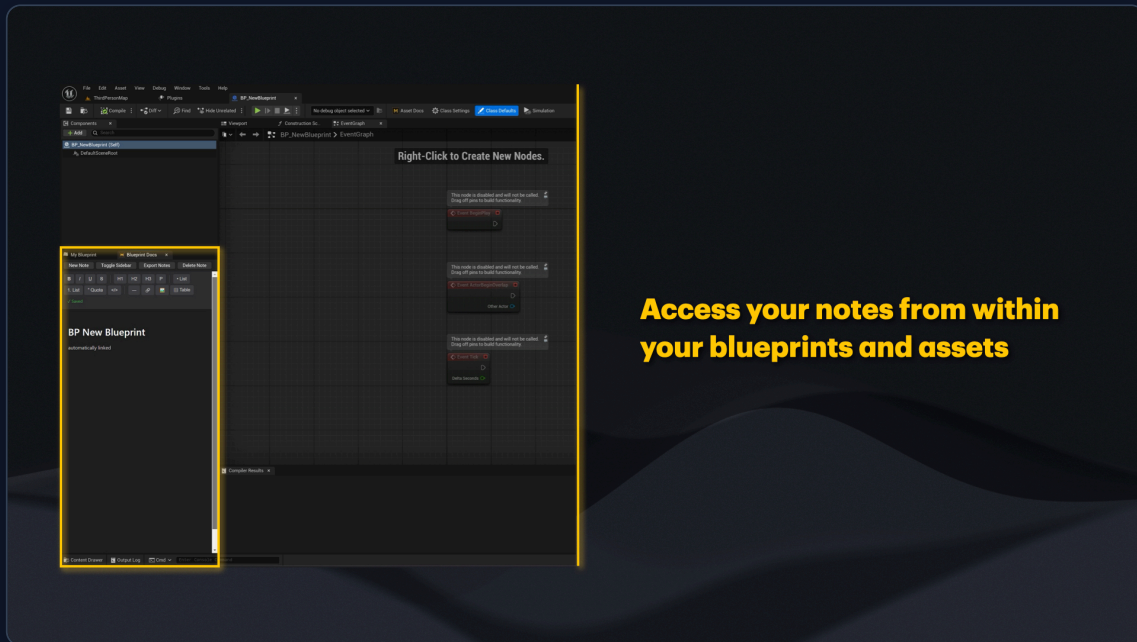

Figure 3: Working on documentation side by side with your Blueprint

Setting Up Side-by-Side View

1 **Open Your Blueprint**
Double-click any Blueprint in the Content Browser to open it in the Blueprint Editor.

2 **Open Documentation Window**
Click **Window → Blueprint Documentation** or use the toolbar button.

3 **Dock the Window**
Drag the documentation tab to dock it beside your Blueprint graph. You can also float it as a separate window.

Auto-Detection: When you switch between Blueprints, the documentation window automatically switches to show that Blueprint's documentation. This works seamlessly whether the windows are docked or floating.

Auto-Open Assets Feature

The Auto-open Assets feature provides a convenient way to navigate from your documentation directly to the associated Blueprint or asset.
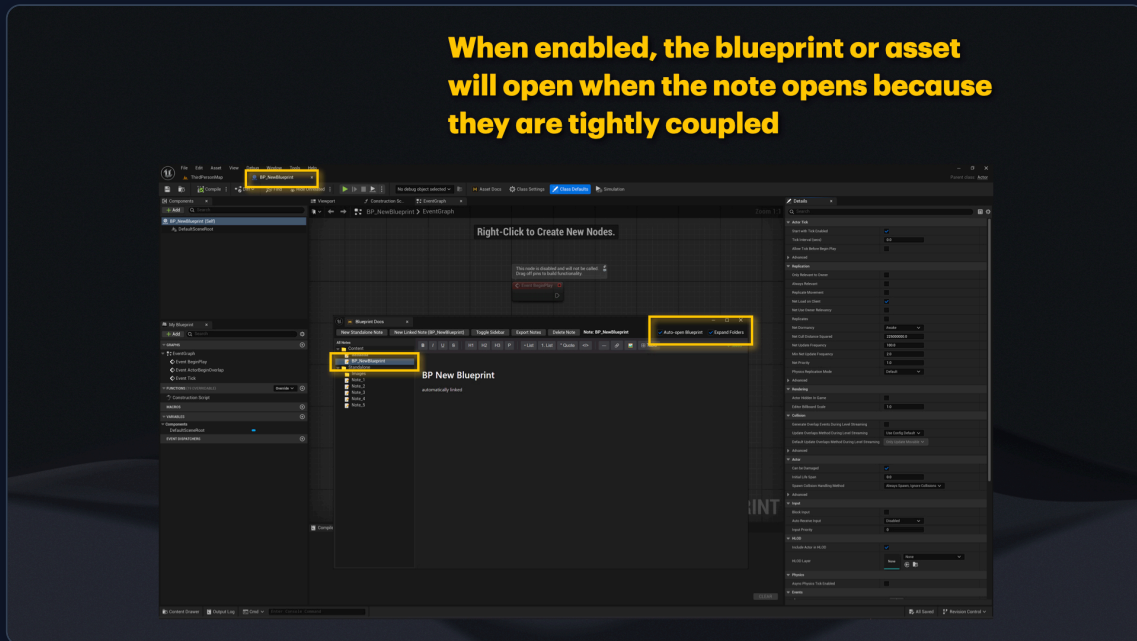

Figure 4: The Auto-open Assets toggle in the toolbar

How It Works

When enabled, clicking on a note in the sidebar will automatically open the associated Blueprint in a new editor tab. This is particularly useful when:

- Reviewing documentation for multiple Blueprints
- Navigating through your project's documentation
- Cross-referencing between notes and their associated assets

Enabling Auto-Open

Toggle the **Auto-open Assets** checkbox in the top toolbar to enable or disable this feature. When checked, clicking any asset-attached note in the sidebar will open that asset in the editor.

Note: This feature only works for notes that are attached to assets. Standalone notes (in the Standalone folder) don't have associated assets to open.

Formatting Toolbar

The editor toolbar appears at the top of the editing area and provides access to all formatting options.
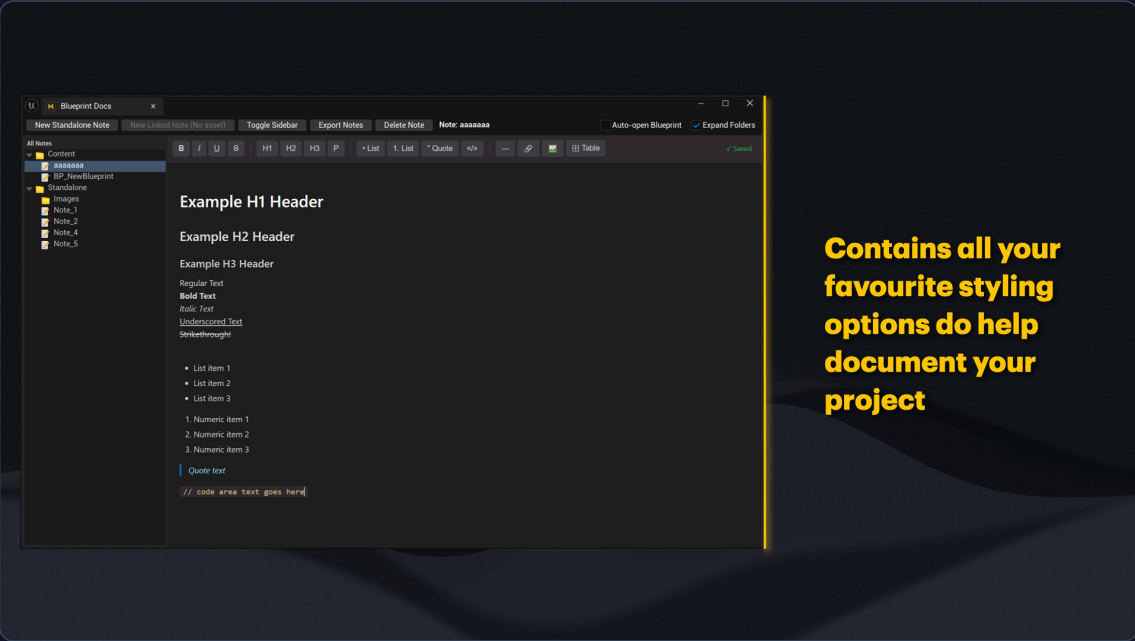

Figure 5: Rich formatting options including headings, text styles, lists, and more

Text Formatting

Button	Function	Shortcut
B	Bold text	Ctrl + B
I	Italic text	Ctrl + I
U	Underline text	Ctrl + U
S	Strikethrough	—

Headings & Paragraphs

Button	Function	Description
H1	Heading 1	Largest heading, for main titles
H2	Heading 2	Section headings
H3	Heading 3	Subsection headings
P	Paragraph	Normal text

Lists & Blocks

Button	Function
• List	Bullet list
1. List	Numbered list
" Quote	Block quote (for callouts or notes)
</>	Inline code (for variable names, etc.)

Other Elements

Button	Function
—	Horizontal rule (divider line)
🔗	Insert link
🖼	Insert image
⊞ Table	Insert table

Working with Tables

Inserting a Table

1   **Click the Table Button**
Click ⊞ Table in the toolbar to insert a 3-column table.

2   **Edit Headers**
Click on "Header 1", "Header 2", etc. to change the column names.

3   **Fill in Cells**
Click any cell and start typing to add content.

Table Editing Toolbar

When you click inside a table, a floating toolbar appears with these options:

Button	Function
+ Row Above	Add a new row above the current row
+ Row Below	Add a new row below the current row
+ Col Left	Add a new column to the left
+ Col Right	Add a new column to the right
– Row	Delete the current row
– Col	Delete the current column
Delete Table	Remove the entire table

> **Tip:** Columns automatically resize to equal widths. You don't need to worry about alignment!

Working with Images

Inserting an Image

1 **Click the Image Button**
Click 🖼️ in the toolbar.

2 **Select Your Image**
A file browser opens. Navigate to and select the image you want to insert.

3 **Image Appears in Editor**
The image is inserted at your cursor position.

> **Supported Formats:** PNG, JPG, JPEG, BMP, GIF, TGA

Resizing an Image

Images can be easily resized by dragging, allowing you to fit them perfectly into your documentation.
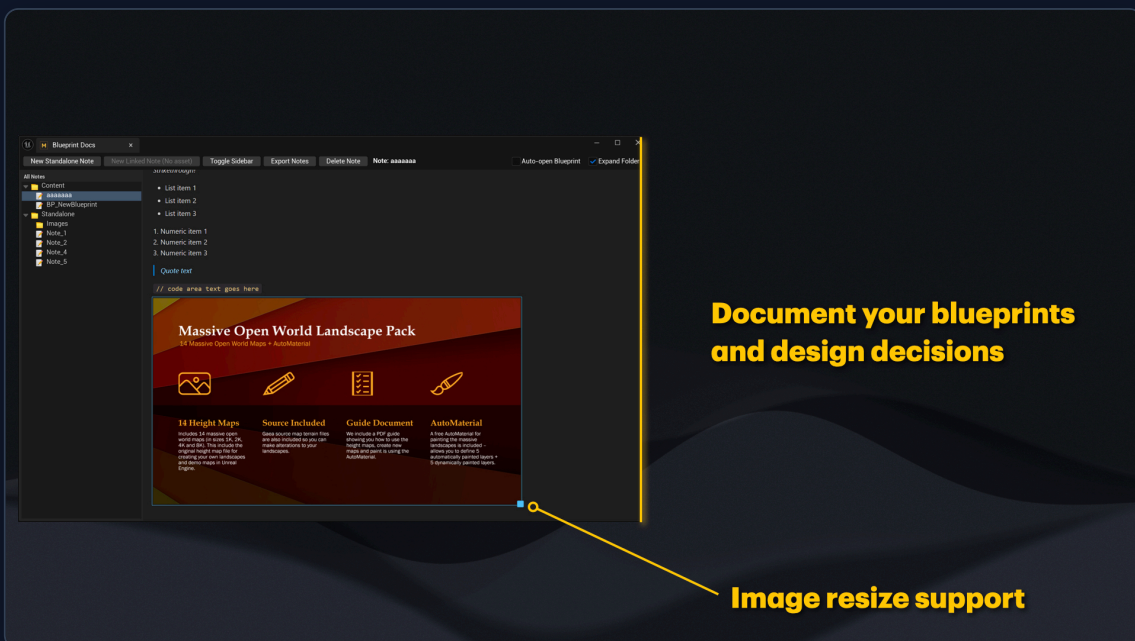

Figure 6: Resizing an image by dragging the corner handle

1 **Click the Image**

Click on the image to select it. A blue outline appears.

2 **Drag the Handle**

A small square appears at the bottom-right corner. Drag it to resize.

3 **Size is Saved**

The new size is saved automatically in the markdown file.

> **Image Paths:** Images are referenced by their file path on your computer. If you move the image file, it won't display in the editor. Use a dedicated folder for documentation images.

Best Practice for Images

- Create a folder like `Documentation/Images/` in your project
- Put all documentation images there
- This keeps them organized and ensures they work for your team

File Organization

Where Are Notes Stored?

All documentation is stored in your project at:

```
YourProject/Documentation/Blueprints/
```

Asset-Attached Notes

Notes for Blueprints and assets follow your project's folder structure:

Asset Location	Note Location
Content/Characters/BP_Player	Documentation/Blueprints/Content/Characters/BP_Player.md
Content/UI/W_MainMenu	Documentation/Blueprints/Content/UI/W_MainMenu.md
Content/Blueprints/BP_GameMode	Documentation/Blueprints/Content/Blueprints/BP_GameMode.md

Standalone Notes

Notes created with "New Note" go in:

```
Documentation/Blueprints/Standalone/
```

> **Version Control Friendly:** All notes are plain text .md files that work great with Git, Perforce, or any version control system.

Configuring the Storage Path

You can change where documentation is stored by clicking the Settings gear icon in the toolbar. This opens a dialog where you can browse to a new location or restore the default path.

Moving Notes: When changing the storage path, existing notes are copied to the new location. The destination folder must be empty to avoid overwriting existing files.

Exporting Documentation

You can export all your documentation to share with team members or for backup purposes.
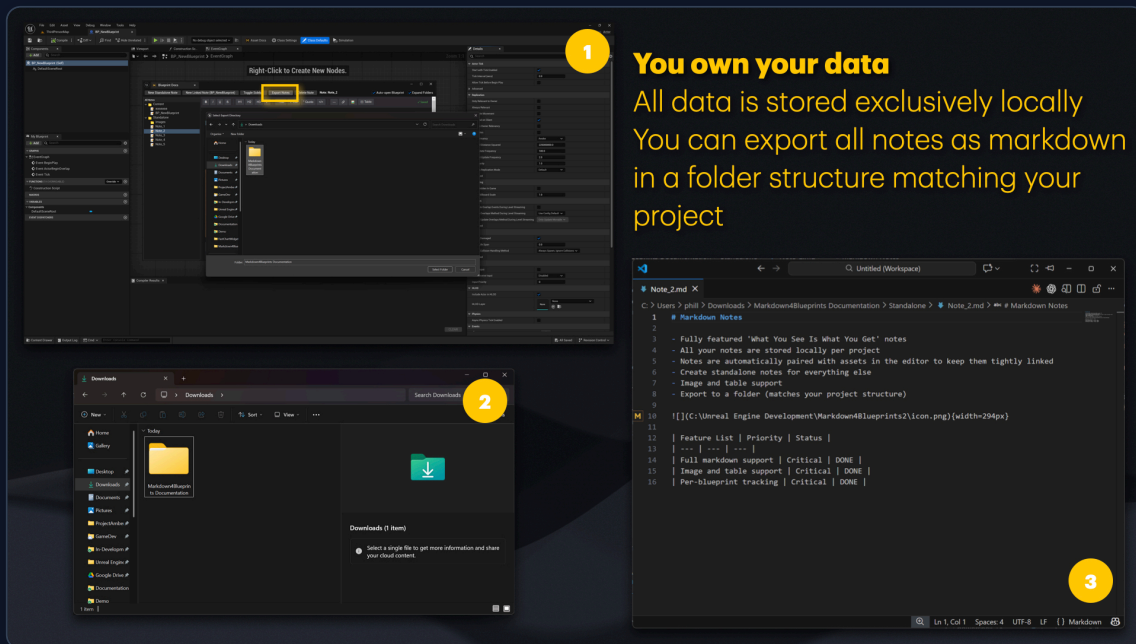

Figure 7: Exporting documentation to an external folder

Export All Notes

1   **Click "Export Notes"**
Click the "Export Notes" button in the top toolbar.

2   **Choose a Destination**
Select a folder where you want to export the documentation.

3   **Documentation is Copied**
A folder named "{ProjectName} Documentation" is created with all your notes.

Use Cases:

- Share documentation with team members who don't have the project
- Create backups of your documentation
- Generate documentation for clients or publishers

Markdown Compatibility

Exported files are standard markdown (.md) format that can be viewed with:

- Visual Studio Code (with markdown preview)
- GitHub / GitLab (automatically rendered)
- Any markdown editor or viewer
- Converted to HTML, PDF, or other formats with tools like Pandoc

Tips & Best Practices

Documenting Blueprints Effectively

What to Document

- **Purpose:** What does this Blueprint do? Why does it exist?
- **Usage:** How should other developers use this Blueprint?
- **Variables:** What are the important variables and what do they control?
- **Events:** What events does this Blueprint respond to?
- **Dependencies:** What other Blueprints or systems does this rely on?
- **Gotchas:** Are there any quirks or things to watch out for?

Organization Tips

- Use **Heading 1** for the main title (usually the Blueprint name)
- Use **Heading 2** for major sections (Overview, Usage, etc.)
- Use **Heading 3** for subsections
- Use **tables** to document variables, functions, and parameters
- Use **code formatting** for variable names, function names, and values

Keyboard Shortcuts

Shortcut	Action
`Ctrl` + `B`	Bold
`Ctrl` + `I`	Italic
`Ctrl` + `U`	Underline
`Ctrl` + `Z`	Undo
`Ctrl` + `Y`	Redo

Troubleshooting

Common Issues

Editor Shows Blank or Loading

- Wait a moment - the browser engine may still be initializing
- Try closing and reopening the documentation window
- Check the Output Log for error messages

Changes Not Saving

- Check the save status indicator in the top right
- If stuck on "Saving...", try clicking elsewhere then back
- Ensure your Documentation folder isn't read-only

Images Not Displaying

- The image file may have been moved or deleted
- Check if the file still exists at the original path
- Try re-inserting the image

Table Toolbar Not Appearing

- Click directly inside a table cell
- The toolbar appears above the table when a cell is selected

Note Not Appearing in Sidebar

- Make some changes to trigger auto-save
- New notes appear in the sidebar after first save
- Toggle the sidebar off and on to refresh

Cannot Move Notes to New Location

- The destination folder must be empty
- Check that you have write permissions to the destination
- Ensure no files in the destination have the same names as your notes

Getting Help

If you encounter issues not covered here, check the Output Log:

1. Open **Window → Developer Tools → Output Log**

2. Look for messages starting with "BlueprintMarkdown"

3. These messages can help identify what's going wrong