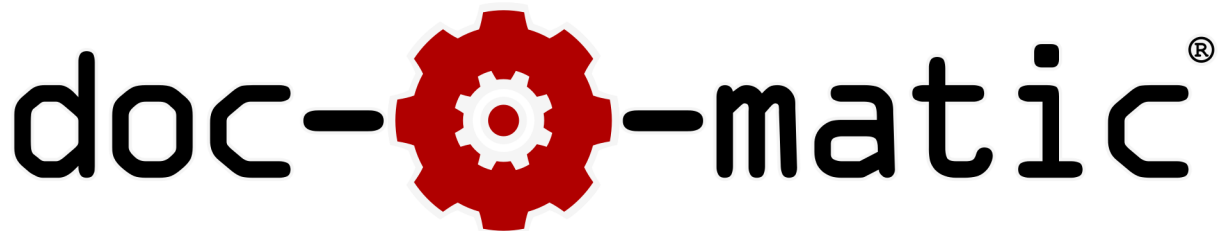# Doc-O-Matic 10 Release Notes

Doc-O-Matic 10 is a major new release of our industry-leading source code documentation and help authoring software. This new version comes with many exciting new features and improvements, there is virtually no area in the application that hasn't dramatically improved over its predecessor.

Doc-O-Matic 10 comes in a completely new, light-weight user interface. The new ribbon interface brings many advantages, first and foremost a centralized, easily discoverable place where you do and find things. Doc-O-Matic 10 enables you to manage and write documentation faster and more efficiently then ever before.

Another very important improvement is that Doc-O-Matic 10 is now a native 64-bit application, enabling users to build virtually limitless projects.

## Release Notes

The first part of this document consists of the release notes for all incremental updates to the initial 10.0 release. After that you can find the a summary over the most important new features, improvements as well as a couple of noteworthy bug fixes in the initial 10.0 release.

# Updates since 10.0.0

## New in Version 10.1.6 – 2021-05

### Server Connectivity

- Fixes a problem connecting to Doc-O-Matic servers for updated versions and license activation.

# New in Version 10.1.5 – 2021-04

## Trial Functionality

- In trial mode, Doc-O-Matic no longer scrambles the output text of longer paragraphs. Except for a disclaimer printed on all pages, all output content is now completely identical to the licensed version's output.

## Bug Fixes

- Fixes a crash in OpenSSL on some Windows 10 systems when requesting a trial license, registering a license key or during other actions that access Doc-O-Matic servers. Doc-O-Matic now ships with a newer version of the OpenSSL libraries that fix this problem.

# New in Version 10.1.4 – 2020-09

## Bug Fixes

- Fixes a problem when generating **Help 3 for Visual Studio 2019**. Newer versions of Visual Studio 2019 require the help contents not to contain DTDs. If HTML pages contain DTDs, the generated help system will not be installed, instead an security error message will be shown during integration into the Visual Studio catalog.
- Fixes a problem when using **C/C++ macro expansion** in connection with the **Topic Editor** where the editor may allow editing of a topic even though the original source code is not UTF-8 encoded.

# New in Version 10.1.3 – 2020-04

## Bug Fixes

- Fixes a problem when processing **XMLDoc** SEEALSO tags. In previous versions, cref attributes containing topic IDs with spaces worked. This update restores the same behavior. Note that cref attributes in SEE tags will still be processed in a different fashion to allow for usage of link targets that spell out parameters and type information.
- Fixes a long standing issue when generating **PDF** output that uses Russian, Greek, Turkish and Eastern European character sets. Previous releases may have generated unreadable or incorrectly layouted PDF files if one of these character sets had been used.

- Fixes a problem generating **PDF** files when embedding or referencing certain types of True Type fonts in PDF.

# New in Version 10.1.2 – 2020-01

## Improvements and Bug Fixes

- **Dramatically improves** the speed of **PDF** output generation when using watermark images and title page backgrounds or other large images: Depending on the size of the project and the number of images used, the new version generates PDF files in less than *1/10th of the time* used previously.
- **Fixes** a long standing issue with **PDF** output generation involving oversized graphics: If a graphic is taller then the maximum available space on a page, the new version now prints the oversized graphic (it may be clipped). Previous versions would suppress the output entirely also suppressing text that followed it.
- **Improves** the behavior of in the **HTML** sidebar implementation: When focusing the table of contents entry of the current page, the sidebar avoids unnecessary scrolling to the right, resulting in a calmer viewing experience.
- **Fixes** a small settings issue with the default **HTML** output configuration which would hide the X-axis scrollbar in the sidebar (now sets `overflow-x: auto;` in the additional CSS field by default).
- **Fixes** a problem in the **Topic Editor** when writing topics: If a topic consists of 2-3 lines with the last two lines being a bulleted lists, such topics were not written properly. Upon re-reading the text would loose the list.
- **Fixes** an issue with the **Topic Editor**'s `Insert Section Heading` feature: When the cursor is located inside a bulleted or numbered list, the new section heading will now be inserted above the entire list, no matter which list item the cursor is located in.
- **Fixes** a problem with automatic Text-to-DTX conversion when adding plain text files to a project: Doc-O-Matic now ensures the file encoding is correctly identified in order to preserve its content. Previous versions may have failed to identify UTF-8 encoded files if they lacked a byte order mark.

# New in Version 10.1.1 – 2019-04

## New Support for Visual Studio 2019

- **Adds** a new Project Template for Visual Studio 2019, which comes with 3 configurations, all of which integrate visually into the corresponding Visual Studio style:
  - Visual Studio 2019 for integrating into the Help Viewer help system.
  - Web Help for stand along browser based help.
  - Print for printable PDF output.

- **Adds** new Project Databases for automatic linking into the Microsoft Visual Studio 2019 .NET 4.7 help systems, suitable for both WebHelp/HTML and in Help Viewer output.

## New HTML Output Features

- **Adds** a sidebar splitter bar: The dynamic TOC sidebar now provides a splitter bar along its right edge which allows for resizing the area. The user-adjusted size of the sidebar is persisted across pages and sessions.
- The sidebar now uses **responsive layout** and disappears from the page automatically if the space becomes too small (adjustable via the additional CSS setting).
- **Adds** a new setting on settings page `[Web Help] > Files & Format > Output Options > Contents Page` named "`Sidebar splitter moves topic title`" which influences how topics pages behave when the sidebar splitter bar is moved. When this option is turned on, the topic title will be moved along such that it aligns with the page content.
- Slightly updates the HTML default output formatting.

## Bug Fixes

- The TOC sidebar now syncs its tree to the displayed page much more quickly (now after 50ms, was 500ms).
- HTML Help configurations in new projects created with version 10.1.0 had a background color missing for the welcome page title and sub-title which was displaying text with white color on white background, giving the impression of invisibility. This problem also appears in regular HTML output when viewing the welcome page using old versions of Internet Explorer.

# New in Version 10.1.0 – 2019-01

## New HTML Output Features

- **Adds** a new option for a **Dynamic Table of Contents Tree** which is embedded in the page content, much like the table of contents tree in help systems.
  - The tree automatically synchronizes with the topic or folder that is currently displayed.
  - This option is available on configuration settings page `[Configuration] > Files & Format > Output Options > Contents Page`.
  - This option is turned on by default in new projects.
  - When using this feature, Doc-O-Matic automatically enables the fixed header and footer settings (even if the corresponding options are not turned on in the project configuration).

- **Adds** a new option for a fixed footer on configuration settings page `[Configuration] > Files & Format > Layout`. Similar to the fixed header option, this setting causes the page footer to stay visible at all times.

- **Adds** new index search options for case sensitive search and whole word search. These options are stored across sessions and restored automatically when visiting the index page. These options also work in the embedded index search.

## New Comment Processing Features

- **Adds** a new editor comment writing option named *Minimum width of comment* to help create a more uniform looking source code comment structure, where all comments have the same minimum width, regardless of how wide their respective contents are.

- **Adds better support for Doxygen**-style comments:

  - **Adds** processing for the following Doxygen-specific one-word styling tags: `@a`, `@e`, `@em`, `@b`, `@c`, `@p` and `@n`.

  - **Adds** support for `@code` and `@endcode` code block markup.

  - **Adds** support for `@li` list item tags.

  - **Adds** a new option on general settings page `[General Settings] > Sections > Javadoc` named *Support Doxygen Tags* which turns on the new features above.

  - The default configuration now includes *@brief* and @note section mappings.

  - All tags above can also be used in the backslash version if the setting "Support backslash tags" is turned on as well.

- **Adds** a new section option via general settings page `[General Settings] > Sections` in the section editor, named *End automatically after paragraph*. When this option is turned on, a section ends with its first paragraph. As an example, this enables support for more compressed comments, where a summary section starts the comment, which automatically terminates and the general description section can follow without its own starter. For example this can be used for Doxygen-style comments like this:

```
/* @brief Example showing how to document a
 * function with Doxygen.
 *
 * Description of what the function does. This
 * part may refer to the parameters of the ...
```

- In the example above, the second paragraph will be put in the project's default section (usually "Description").

- **Improves** the comment wall remover.

doc-⚙-matic®

## New XMLDoc-related Features

- **Adds** new **`paramref`** tag auto-label processing:
  - If a project contains a text label called "auto.xml.paramref", this label will be used to format the paramref tag's contents automatically.
  - The label can be freely configured, including formatting, images. However, the exclude option has no effect.
  - Doc-O-Matic automatically upgrades old projects by adding this label. If the label is removed from the project, Doc-O-Matic will process **`paramref`** as it currently does, converting the tag's content to plain text.

- **Adds** support for non-standard use of XMLDoc SEE tags used to set the link's text to something other than the link target. The first variant gives a link with the text *LinkTarget*, the second with a link target and the third gives *a new link target*:

```
// 1. Link to <see cref="LinkTarget"/>
// 2. Link to <see cref="LinkTarget" text="a link target" />
// 3. Link to <see cref="LinkTarget">a new link target</see> (NEW)
```

- **Adds** support for non-standard use if XMLDoc SEEALSO tags. Doc-O-Matic will now correctly ignores the content of the tag "Foo Class" when it encounters a tag like the following:

```
// <seealso cref="Foo">Foo Class</seealso>.
```

## New C#-related Features

- **Adds** support for **C#** lambda expression-bodied properties:

```csharp
public bool IsValid => name != null && id != -1;
```

- **Adds** support for **C#** lambda expression-bodied methods:

```csharp
public int GetHashCode() => worker.GetHashCode();
```

- **Fixes** a problem processing **C#** source code that caused symbols declared on namespace-level without access specifier to be `public`. Namespace-level symbols without explicit access specifier are now `internal`.

- **Fixes** a problem with resetting the access specifier in **C#** partial classes when one or more partial implementations were not declaring an access specifier.

## New in the Topic Editor

- **Adds** new customizable background colors for text labels in the Topic Editor:
  - Adds a new option in the text label configuration to customize a label's background color in the editor.
  - When using many labels, it's much easier to distinguish them with customized background colors, instead of

having all of them displayed using the same default color.

- This setting is available via `[Configuration] > AutoDoc > Text Labels` from the text label edit dialog.
- As a visual aid during authoring, this setting is not used outside the Topic Editor.

## General Features and Changes

- **Adds** a new variable named %CLASS% for the *Symbol Titles* and *AutoDoc Topic Titles* on settings pages `[Configuration] > AutoDoc > AutoDoc Content > Titles` and `[Configuration] > AutoDoc > AutoDoc Content > AutoDoc Titles` to enable the display of a member's class name without its namespace. This variable can be useful for disambiguation of index titles. This variable is defined only for class members.
- **Changes** the way the default export status is determined for `structs` contained in namespaces: If the source code language has a concept of visibility on namespace-level, namespace-level `structs` will get exported if both of the following conditions are met:

  1. the `global struct` export setting is turned on (see page `[Configuration] > AutoDoc > Export`) **AND**

  2. the export setting for `nested classes` with the matching visibility is turned on as well (see page `[Configuration] > AutoDoc > Export > Members`).

## Online Help Language

- **Adds** a new option under the new `Basics > Language` application settings page that allows for selecting the language of the online help. Doc-O-Matic ships with online help in **English** and **German**.

# New in Version 10.0.4 – 2017-09

- **Adds** parameter type lists in the output of navigation tables for overloaded global functions the same way that overloaded methods have parameter type lists.
- **Adds** a new topic properties editor option that allows to switch between column mode and list mode. This option is available in the Application Options on the "Editor" section.
- **Better** C/C++ standard compliance by distinguishing overloaded functions with const parameters from those that have no const specifiers. The example below illustrates the difference. In previous versions, this would have resulted in one function declaration being visible, from now on, these will be distinguished. Note that this change may affect existing documentation that uses external documentation as the topic IDs for such methods have changed.

```
class demo {
  void foo(T*& t);
  void foo(const T*& t);
}
```

- For C++ source code, **adds** the `const` keyword for const parameters in parameter type lists (topic titles, navigation tables).

- **Changes** how Doc-O-Matic generates symbol IDs for generic classes in Delphi. The generic argument type (if any) is no longer part of the symbol ID and thus no longer contained in the topic ID of such topics. As a result the topic IDs of external documentation for such classes may need to be updated. No action is required for in-source documentation.

```
type
  // the topic ID is TFoo@T
  TFoo<T:TControl> = class
    // the topic ID is TFoo@T.Bar
    procedure Bar;
    end;
```

- **Fixes** a problem with Delphi source code and generic classes or records where implementation location comments and implementation location source code was not picked up correctly.

- **Changes** how Doc-O-Matic looks for suitable parsers for source parsing: The search is now case insensitive. Previously this wasn't the case and as a result the search may have yielded no result, thus the default parser was used (C/C++).

- Doc-O-Matic now ensures all parser file extensions are automatically lower case when loading old projects or when editing file types on settings page `[General Settings] > General > File Types`.

- **Fixes** a potential crash during startup on Win32 systems.

- **Fixes** a crash while generating HTML output caused by invalid topic markup.

- **Fixes** a problem with the font size of source code printed in the Content Manager's source view when using a Hi-DPI display.

## New in Version 10.0.3 – 2017-03

- **Fixes** a problem with XML output were the symbol hierarchy tags could end up mangled.

- **Fixes** a problem with the "Add Folder" dialog which didn't scale on HI-DPI displays.

## New in Version 10.0.2 – 2017-02

- **Adds High-DPI support:** This update ensures that Doc-O-Matic works great and looks fantastic on 4k screens

and other high-resolution displays. This update fixes a number of problems in the UI when using High-DPI displays or when using screen scaling factors greater than 100%.

- **Adds Implementation Details:** Implementation Details is a new feature to control the default export state of symbols on settings page `[Configuration] > AutoDoc > Export`.

  - By default this option is active, which means all symbols' default export states are as they were before.

  - When the "Implementation Details" option is turned off, symbols that are considered implementation details will have their default export state turned to "Not exported".

  - The following symbols' default export state can be influenced using this setting:

    - Symbols declared in C/C++ source files (as opposed to header files).
    - Symbols declared in the `implementation` section of Delphi units.

- **Adds** C# parser support for the `Nullable<T>` shorthand "`T?`".

- **Improves** the handling of XMLDoc `see` and `seealso` links by normalizing the link targets while parsing XMLDoc.

- **Adds** a new option named "Add generic arguments to class and methods in title" on settings page `[Configuration] > AutoDoc > AutoDoc Content > Titles`. When turned on (the default) any classes or methods with generic/template arguments will have these arguments printed in topic titles and in links on navigation pages.

- **Adds** the generic argument specification to method names in the Content Manager for methods that have such specifications.

- **Adds** an entry to the default type conversion matrix for null so default values/initializers in parameter lists are properly converted in multi-language syntax output.

- **Updates** many of the supplied spell checker dictionaries.

- **Fixes** a problem with generating class hierarchies that contain classes generic/template classes.

- **Fixes** an issue with resolving link targets in navigation tables printed in folder topics (such as overloaded folders, member type folders, visibility folders) where link targets would not be found even though the same link is properly resolved in the corresponding symbol's documentation page.

- **Fixes** a problem when resolving non-qualified link targets into targets from link databases if the links are contained in content of member descriptions.

- **Fixes** a crash when converting UTF-16BE files to UTF-8 format.

- **Fixes** a problem handling XMLDoc `<see cref="abc"/>` and `<paramref name="abc"/>` tags inside XML item lists.

- **Fixes** C# default parameter values not being output in the generated declaration syntax.

- **Fixes** a problem with enum parsing in C#, C++, Pascal and VB that can cause enum members to be ignored when using attributes.

- **Fixes** a problem when using member or visibility folders and overloaded method folders at the same time that

could cause some methods end up in the wrong overloaded folders under certain circumstances. The fix for this problem involves a change how topic IDs for overloaded method folders are generated. If you have added content to overloaded method folders, the topic IDs for these folder topics may need to be updated (you can use the Topic Editor's "Assign to Symbol" feature for this purpose).

- **Fixes** a problem with the indentation of enum members in the generated syntax under certain circumstances.

- **Fixes** a problem with the Search panel in the Topic Editor which could appear incorrectly after expanding the topic properties pane.

- **Fixes** a problem in QA View and Class Hierarchy View that swallowed hints containing the pipe symbol.

- **Fixes** a problem in HTML output where folder topics did not get exported when they are part of a topic hierarchy which was merged on a single output page through the use of the "Join Sub Topics" flag in one of its parent topics. Now these topics will appear as expected as a simple topic heading.

- **Fixes** a problem with index entries in the searchable HTML index if the index entries contained special characters.

- **Fixes** other minor issues and configuration default settings.

# New in Version 10.0.1 – 2016-12

- **Adds** the Visual Studio 2017 project template.

- **Adds** support for building and integrating Help Viewer help for Visual Studio 2017.

- **Adds** access to the "Proxy" options on the license key dialog so users behind non-transparent proxy servers can access the network properly when requesting, installing or uninstalling licenses.

# New in Version 10.0 (Main Release)

## Doc-O-Matic is a 64-bit application

- 64-bit means virtually limitless project sizes.

- Faster processing and output generation for large projects.

- Doc-O-Matic is still available as a 32-bit version for compatibility with older hardware and operating system versions.

- Previously projects with a lot of code and documentation could hit hard limits, requiring partitioning of projects. This is no longer necessary.

# Completely new User Interface

- Doc-O-Matic now uses the Microsoft Ribbon interface. It comes with a lightweight visual style that is both inviting and easy to use and perfectly fits the modern Windows design.

- All commands are now available from the ribbon, instead of individual tool bars, making the user interface much more accessible and much less cluttered at the same time.

- In Doc-O-Matic all main views have context-sensitive ribbon tabs.

- In addition to the ribbon, all commands that are available in ribbon tabs are available as context-menu commands in the individual views.

# New Features and Improvements

- Doc-O-Matic 10 comes with updated artwork including completely new, standardized symbol icons.

- Sealed classes, structs, enums, templates and so on now have their own icon.

- Text variables can now be configured to use a label:

  - If you are using text variables which always use the same label, you can now have Doc-O-Matic apply the label automatically upon output generation.

  - The variable will behave as if it was labeled in the editor, with all options labels support, such as text replacement, formatting, prefix and suffix as well as exclusion from output.

  - Using this mechanism, you can easily exchange a variable's behavior in the output without changing any text in the documentation.

- For template/generic classes, Doc-O-Matic now prints the template arguments in the Content Manager tree, so they can be distinguished easily.

- New Convert-On-Open feature for UTF-8:

  - In order to edit topics using the integrated Topic Editor, they now must be contained in an UTF-8 or ASCII encoded source or topic file. Files in different encodings (Ansi, UTF-16, UTF-32) are parsed and processed but their content can't be edited directly.

  - When opening a project (as well as on each subsequent project update), Doc-O-Matic automatically verifies the encoding of source, DTX and snippet files and offers to convert them to UTF-8. UTF-8 conversion is completely optional, the documentation generation will work as before for content in files that are not UTF-8 encoded.

  - When viewing a topic whose file encoding is incompatible, Doc-O-Matic offers a convenient way to convert the encoding of that file with just a single click, right in the editor.

  - A new application option named "Ask to convert source code files to UTF-8" has been added to settings page

`Parsing and Conversion` which allow for disabling the conversion query for source files.

- When adding a new output configuration to a project, you can now choose to add a pre-configured configuration from one of the installed template projects. This makes it super-easy to try out new template configurations in existing projects, without having to copy all the settings from the old project into a new one. On the new "Add from Template" tab in the "Add Configuration" dialog, Doc-O-Matic now lists all templates. Upon selection of a template, it lists the configuration this template comes with. When a configuration is added, Doc-O-Matic copies all output-configuration settings from that template project configuration into a new configuration of the current project.

## Code and Input Processing

- Doc-O-Matic now supports declarations of multiple template/generic classes with the same name but different template/generic argument lists. In previous versions only one such class would survive.

  ```
  public class Template<T, A> {}
  public class Template<T, A, C> {}
  ```

- Adds support for C++ final classes and methods.

- Adds support for C++ unnamed namespaces.

  ```
  namespace { ... }
  ```

- Adds support for C# async methods.

  ```
  async private void OnStatusChanged(Geolocator sender, EventArgs e) {}
  ```

- Adds support for Java enums.

  ```
  public enum Day { SUN, MON, TUE, WED, THU, FRI, SAT }
  ```

- Adds support for Java generic classes

  ```
  public class Box<T> {}
  ```

- Adds support for XMLDoc tag attributes values that use single quotes (when reading documentation, when writing Doc-O-Matic will always use double quote). Previously only double quotes were supported.

  ```
  <see cref='System.Windows.Component'/>
  <see cref="System.Windows.Component"/>
  ```

- Adds "devdoc" as a recognized XMLDoc tag attached to the Description section of the factory default settings as well as the Visual Studio 2015 template project.

## Topic Editor

- New Search Panel

doc-**⚙**-matic®

- Text searching from within editor is more streamlined using an on-demand quick-access search panel at the bottom.
- To find text, use CTRL-F (or the corresponding ribbon command) to open the panel
- Enter the text you're looking for and press the Enter key, repeat the search by pressing the F3 key, stop the search by pressing the ESC key or clicking the close button on the panel.
- The search panel comes with auto-complete suggestions and a search history.
- Embraces the new ribbon UI, all commands are now available from the "Topic Editor" ribbon tab as well as from the context menu.
- Adds mouse-hover hints for labeled text.
- Adds mouse-hover hints for links.
- The image name field in the Add/Edit Image dialog now supports auto-complete for rapid image lookup.
- The link target field in the Add/Edit Link dialog now supports auto-complete for rapid topic ID lookup.
- The topic properties inspector now makes better use of the available space by arranging the properties in columns.
- The editor now supports transparent PNG images.

## Topic Preview

- The topic preview now displays any type of text including tables, links, etc.
- The project can now be navigated through the preview. Links in the topic preview are live and click-able and open the corresponding topics they link to.

## Tools

- Adds a new tool under the "Project" ribbon tab, section "More" named "Make Configuration Consistent". This tool compares the current configuration with all other configurations in the project and will copy any text variables and text labels the current active configuration might be missing.
- When adding a new configuration, it automatically gets consistent text variable and labels from the other configurations in the project.

## General Output

- The message view configuration is now more easily accessible by a button row that sits on the split panel itself. It is now also possible to see the number of messages in the different types (hint, info, warning, error) while the

message pane is collapsed.

- Local and global class hierarchies are now generated as PNG files (was BMP). This not only makes the output smaller but also increases output speed as the files don't need to be converted for the output formats.

- As the different output formats have other means to produce the project copyright in various places, the !!COPYRIGHT_FOOTER special topic is no longer used.

## AutoDoc and Navigation Tables

- The option "List inherited class members" on settings page `[Configuration] > AutoDoc > AutoDoc Content > Content Options` now also applies to the member overview page, when generated.

- Adds the new option "Pull overload list folder methods to main navigation level" on settings page `[Configuration] > AutoDoc > AutoDoc Content > Content Options` which makes overloaded methods directly available in the methods lists instead of the folder topic. The folder topic for these methods is still accessible via the TOC.

- When adding inherited members to navigation tables, you can now optionally add a hint text about the origin of the member using the new "Add inherited hint text" option. The added text can be set up in the corresponding format field.

- Adds new option "Names with parameter type list for overloaded methods" on settings page `[Configuration] > AutoDoc > Section Options`. When selected (the new default), method names of overloaded methods in navigation tables will be appended by a parameter type list.

- Replaced the automatically populated "File" section with a more generally useful "Container" section. This section now contains not only the file but also the namespace and the module a symbol is contained in (if any). All three options can be controlled and switched off individually via the new `[Configuration] > AutoDoc > AutoDoc Content > Container Section` configuration settings page.

- Removes the "Overloaded list in classic style" option.

- The AutoDoc Symbol Titles (settings page `[Configuration] > AutoDoc > AutoDoc Content > Titles`) and AutoDoc Topic Titles (settings page `[Configuration] > AutoDoc > AutoDoc Content > Titles`) both have an additional column to set link titles. This makes it possible to separate the page title, which was used as the link text previously, from the link text to the pages, if the link text does not need to be as verbose as the topic title that is printed on the page.

- The "Reset Output Options" feature on settings page `[Configuration] > AutoDoc` now allows you to reset Doc-O-Matic built-in image paths and Doc-O-Matic built-in images. Using this option will reset any flag images, indicator images, output images to the current factory default if they have not been customized. It will also ensure all Doc-O-Matic supplied image paths are set up correctly and will add output-format specific additional images.

# New HTML Features

- Completely redesigned topic paths (bread crumbs):
  - Enables extremely fast navigation to sibling topics.
  - Quickly find sibling topics on any level up to the current root. For example, you can instantly access a different class from a member topic of a different class in the same namespace.
  - Upon click on an entry in the topic path, each level in the current topic path exposes the sibling topics on this level.
  - The displayed path is automatically and dynamically limited to one line. If a path is longer than there is space available, it gets truncated starting at the root.
  - Dynamic topic paths are available in the displayed topic instantly, there are no delays for server side queries.
  - This feature can be controlled via the "Dynamic topic path browsing" option on configuration settings page `[Configuration] > Files & Format > Layout > Topic Paths`.
- Dynamically adds table of contents and index search:
  - When using the dynamic Table of Contents (the default), this option can also add an embedded Table of Contents accessible through the TOC page link. The TOC is loaded dynamically when required. This is extremely convenient because it makes it possible to browse the TOC without leaving the current page. A new option on the TOC configuration page has been added that controls this feature.
  - When using dynamic index search (the default), this option can also add an embedded keyword search UI accessible through the index page link on the top of each page. It is just like the keyword search interface on the index page but accessible instantly without having to leave the current page. A new option on the index configuration page has been added that controls this feature.
  - Due to security issues, some browsers prevent dynamic content loading from local file systems (Microsoft Edge). If a dynamic embedding attempt fails for any reason, the corresponding page will be loaded normally. Also, dynamic page content loading does not work inside compiled HTML help.
- The page links (header and footer links to TOC, index, home page, etc.) are now configurable to use either text, image or both (the new default).
- The generated CSS file now uses friendly class names. This makes is much easier to create customized, static CSS versions with special features.
- HTML output generation is now up to 25% faster.
- Section expand/collapse and tabbed section persistence is now implemented via HTML local storage, if available, otherwise via cookies.
- Simplified and streamlined the generated HTML by utilizing jQuery JavaScript that is detached from the HTML content, thus making it much cleaner.
- The dynamic page elements like expandable sections, tabbed sections, embedded TOC and the new table

filters use the jQuery back end.

- The fixed header feature now uses standard CSS and simple jQuery functions.

- JavaScript has been updated and dependencies on deprecated browser technologies have been removed. Also removed have been the cookie and persistent storage naming options. These names are now pre-selected in the script file.

- Member table filtering:

  - All navigation tables containing class members now automatically have interactive filters for `inherited` members as well as for members that have a visibility of `protected`, `internal`, `private` or `protected internal`.

  - Only those filters that are applicable in a table are visible. In other words, if a table doesn't contain private members, it will not display a `private` filter. Filters that are turned off cause the corresponding rows in navigation tables to become invisible. If a row belongs to multiple filters at the same time, if any filter is active, it becomes invisible, regardless of the state of the other filters.

  - Filters are persistent across pages and sessions.

  - There are new formatting elements that control the look and feel of the filters.

  - A new option named "Add filters for member tables" on settings page `[Configuration] > Files & Format > Sections > JavaScript` controls this feature.

- Adds a new option page at `[Configuration] > Files & Format > Layout > Formatting > Additional CSS` which provides CSS that isn't attachable as the formatting elements' "Additional CSS" property. Any CSS entered here will be appended to the Doc-O-Matic-generated CSS. The option makes it easy to add new or overwrite existing behavior.

- Adds new file templates for jQuery and the Doc-O-Matic add-on used for dynamic page behaviors (jqfunc.js) to configuration settings page `[Configuration] > Files & Format > Files & Folders > Templates`. Doc-O-Matic ships with multiple jQuery versions, default is jquery-1.12.4-min.js. jsfunc.js implements all the interactive behaviors in generated HTML files and is required for proper functioning.

- All CSS rules now use points (pt) as unit instead of pixels (px).

- All HTML output is now encoded as UTF-8, the encoding option has been removed.

- The HTML document type can now be set on settings page `[Configuration] > Files & Format > Output Options > Document Type`. Also, adds the "HTML 5" option (the new default) and it is highly recommended to switch existing projects to it.

- A new encoding option has been added to HTML Help project options on page `[Configuration] > Files & Format > HTML Help > Project Options` which sets the encoding for the HTML Help Table of Contents and Index.

- The popup section feature now uses much better UI and placement behavior.

- Adds the new "Embed image data in HTML files" option to configuration page `[Configuration] > Files & Format > Output Options`. This option makes it possible to produce HTML files that are completely self

sufficient, even when using images. This option is ideal of stand-alone Readme files for example. The image data will be embedded in the HTML file itself. As a result, the HTML file size can grow dramatically, depending on the images used. For this reason, this option should only be used for single-page output because using image references (the default) is much more memory conserving.

- The option "Display links below the topic title" on configuration settings page `[Configuration] > Files & Format > Sections > Navigation Links > Format` no longer controls the general appearance of links below the title. If the option "Print feedback link in title link list" on configuration settings page `[Configuration] > Files & Format > Layout > Page Header > Feedback` is active it will be output regardless of whether the navigation links are output. The same applies to section anchor links (which will appear if a section is configured to have an anchor).

- Files can now be dragged into the "Additional Files" list on configuration settings page `[Configuration] > Files & Format > Files & Folders > Additional Files`.

- Adds the new "Always display global links" option to configuration page `[Configuration] > Files & Format > Output Options`. When turned on, this option ensures the welcome page displays global links to contents-, index-, reference- and glossary-page one can use to navigate, even if all global links in the header are disabled (see settings page `[Configuration] > Files & Format > Layout > Page Header > Header Links`),

- The new HTML default output configuration styling and its features embrace the clean UI look of Doc-O-Matic, the setup is much clearer and easier to work with for the documentation reader and more straight forward to customize.

- Adds a new project info section to the Welcome Page which prints the project author, the copyright and version number.

## A new Printable Option: RTF

- Doc-O-Matic adds a new printable RTF output to its supported output formats. This new output format, in addition to being printable, also is editable in any word processor such as Microsoft Word or Libre Office Writer.

- Doc-O-Matic generated RTF output comes with a great many features that users already know from the printable PDF.

- The RTF output configuration closely resembles the settings available for the PDF output. This includes:

  - Topic numbering options

  - Page break options

  - Page layout

  - Header and footer output options as well as

  - Flexible topic title formatting options.

  - The formatting elements available in RTF are largely equivalent to the ones available in the PDF output.

- General document features include title page, copyright page, table of contents and the index.

- The RTF format uses all supported image formats (PNG, BMP, GIF, JPG). PNG images are preferred.

- Images are embedded into the RTF file as PNG format. This means that PNG source images are preferred because they don't need to be converted.

- This new format replaces the old WinHelp format.

## Templates and Project Databases

- Adds a number of new project databases for Visual Studio 2017, Visual Studio 2015 and older versions of Visual Studio as well as Embarcadero RAD Studio 10.0 (Seattle) and Embarcadero RAD Studio 10.1 (Berlin).

- Doc-O-Matic comes with new project templates for Visual Studio 2017 and 2015 and Embarcadero RAD Studio 10.0 and 10.1 which make it extremely easy to produce documentation that integrates visually into the existing online help.

## New integrated Product Feedback

- The new product feedback form is completely self-sufficient and no longer requires entering a mail server or credentials.

- The current Doc-O-Matic project file (.dox) can be attached to the report. This is useful if you want tech support have a look at a project setting, for example.

- The new Application Problem dialog now adds more information to the report that can be useful in identifying and fixing the issue.

## Other UI Changes

- The section configuration now has clearer layout.

- The section starter string configuration was moved to the section configuration dialog.

- The default section can now be changed by clicking the state radio button in the section list directly.

- The topic ID field in the Add/Edit Module dialog now supports auto-complete for rapid topic ID lookup.

- The Update-All reminder setting can now be controlled from the application options.

- Configuration names can now be changed more easily directly in the settings view by double-clicking the selected configuration's tab or by right-clicking the tab and selecting "Rename configuration...".

- The settings pages are now scrollable if the height of the window isn't sufficient to display the entire page.

- The Formatting Element editor dialog now correctly resizes its preview when large fonts are selected.

doc-**-matic®

- The Content Manager now comes with a new command named "Show Source File in Windows Explorer" available from the context menu which opens a new Windows Explorer window and selects the source file the selected symbol is contained in.

- In an attempt to simply the New Project Wizard, the HTML and PDF configurations option pages have been removed.

# Noteworthy Fixes

- Fixes incorrect output and a potential crashes related to auto-formatted variables.

- Fixes a number of potential crashes while building output.

- Fixes a crash in the editor when viewing a topic that uses XMLDoc with non-standard value description lists with more than 2 columns.

- Fixes a problem with C++ friend operators that are implemented inline, like

```cpp
class A {
   friend A operator+(const A& a, const A& b) { ... }
}
```

- Static asserts are now automatically ignored anywhere in C++ declarations.

- Fixes an issue in the topic editor that could cause dropping of empty paragraphs when displaying topics with certain content.

- Fixes a problem with automatic navigation text generation that can cause output problems if the source text contains lists or any kind of formatting or labeling within the word-count limit. Doc-O-Matic now stops copying when it encounters lists and processes formatting correctly, regardless of whether the word count limit has been reached.

- Fixes a problem with comments inside conditional defines that are not immediately in front of the symbol they belong to, which may have caused the maximum comment distance to be exceeded and thus the symbol remained undocumented:

```cpp
#ifdef ENGLISH_HELP
   // Line 1
   // Line 2
   // Line 3
   // Line 4
#elif GERMAN_HELP
   // Zeile 5
   // Zeile 6
   // Zeile 7
```

```
    // Zeile 8
#endif
void Func();
```

- Fixes a problem with VB property declarations that don't declare accessor methods

```
public class Test
  public property MyProp as string
end class
```

- Fixes a problem with the title variables where the %NS% variable wasn't correctly resolved for class members.