New in Version 7

New Output Format: Help 3

- Adds support for the new **Microsoft Help 3** help format. The Help 3 format has been introduced with **Visual Studio 2010**, it can be used for stand alone Help systems and for integration of documentation into the Visual Studio 2010 documentation.
- Adds a new output configuration format for Help 3.
- Adds new settings pages for Help 3 output format specific options in the Files & Folders section of the Help 3 configuration settings on page:

  > [Configuration]->Files & Format->Help 3
- Adds **new project databases** for **linking into the Visual Studio 2010 documentation** including an overview database containing all namespaces and classes of .NET 4, a detailed database for the System.Windows namespace containing all symbols in it and a detailed database for the System.Collections namespace containing all symbols in it.

New Features and Improvements

- Adds a **new locking feature in the Contents Manager** view that allows better locking of the topic structure. The Contents Manager now has 3 different states:
  1. Design mode, allowing changes to the topic hierarchy, export state, folder topic state, in TOC state and sub-topic state check boxes.
  2. TOC Lock mode which allows changes to export state, folder topic state, in TOC state and sub-topic state check boxes but prevents changes to the topic hierarchy.
  3. Full Lock mode which disallows any state change to the topic hierarchy and state check boxes.
- Adds support for **incremental search to the Project Manager**. To search just start typing the item you are looking for, the Project Manager will locate the best match incrementally if one is available.
- Adds a topic editor command to **insert a new section at the beginning of the current paragraph** instead of appending it at the end of the topic. This is in addition to the existing `Add Section` command that adds a new section at the end of the topic.
- Larger images in the editor are now resized smoothly.
- The following editor features now have the option to assign keyboard shortcuts to them via the `Tools->Keyboard Shortcuts` tool: `Edit->Undo All`, `Format->Set List Number`, `Edit->Convert to Text` and `Assign to Symbol`.
- When using XMLDoc in your project Doc-O-Matic now automatically applies section tags to default sections when writing comments back into source files. This assures better compatibility with Visual Studio Intellisense which requires summary tags and does not work properly with

untagged text.

- Adds support for excluding double-extension wildcard file masks with folders. This allows for exclusion of designer files in Visual Studio projects such as "Resources.designer.cs" or "Settings.designer.vb' which are usually not documented by using the wildcard "*.designer.cs" or "*.designer.vb". In previous versions only the last extension part was matched (in this case .vb or .cs) which made wildcard exclusion inappropriate for this kind of files.
- Adds support for Visual Studio 2010 solutions, Delphi 2010 project files and C++Builder 2010 project files.
- Adds support for **XMLDoc documentation using the `<list>` tag** and its dependend tags `<listheader>, <item>, <term> and <description>` to produce both numbered and bulleted lists as well as value description tables with or without headers.
- Adds support for XMLDoc `<see langword="___"/>` tags converting the individual langword to normal text.
- Adds processing support for the Sandcastle XMLDoc `<inheritdoc />` tag in order to avoid warnings.
- Adds support for XMLDoc `<typeparam name="___" >, </typeparam>` tag which works much like the the `<param name="___" >, </param>` tags and constructs a parameter table in the output.
- The automatic property conversion feature now also supports IDL properties declared through functions with return values in addition to the standard declaration that `HRESULT` returning functions with a by-reference parameter:

```
[propget] long Value();
[propget] HRESULT Value([out, retval] long *value);
```

- Adds processing of C++/CLI property details to identify read-only and write-only properties so they can be distinguished in the output by the read-only and write-only indicators:

```
// read only
property const char * Name
{
  const char *get();
}
// read/write
property int Version
{
  int get();
  void set(int value);
}
// write only
property double Number
{
  void set(double value);
}
```

- Adds processing of C# property accessors to identify read-only and write-only properties so they can be distinguished in the output by the read-only and write-only indicators:

```
// read/write
public string Name
{
  get { return name; }
}
public int Version
{
  get { return version; }
  set { version = value; }
}
// write only
public double Number
{
  set { number = value; }
}
```

- Adds a **Visual Studio 2010 template project** for creating projects that integrate visually with the Visual Studio help. Alternatively use the default setup which adds features not present in the Visual Studio help, like graphical class hierarchies, expandable sections, neighbor topic links and more. This template automatically uses the standard Visual Studio 2010 project database for tight integration of the generated output.
- Adds **template projects for Embarcadero RAD Studio 2010 and 2011** for creating projects that integrate visually with the Embarcadero RAD Studio help. These templates automatically use the corresponding project database for tight integration of the generated output.
- Adds a **template project for publishing release notes** using the same style as the Doc-O-Matic release notes.
- Makes the template project selection in the new project dialog more accessible.
- **Dramatically increases performance** for loading and saving project databases. Loading is up to 17 times faster, saving is up to 100 times faster than with previous releases. This means that there's no longer a long delay before a build process can start which makes testing of incremental changes to the documentation by building an update a lot more responsive.
- **Improves auto-linking** of source code and text with external targets contained in project databases such as the Visual Studio Help systems or the Delphi Help systems.
- Adds a the `embarcadero.rs_xe` and `embarcadero.rs2010` namespaces to the predefined namespaces list in the Help 2 configuration for integration into Embarcadero RAD Studio 2011 and Embarcadero RAD Studio 2010.
- Adds a new project database for automatic linking into the **Embarcadero RAD Studio 2010 Help system**.
- Adds a new project database for automatic linking into the **Embarcadero RAD Studio 2011 Help system**.
- Adds a **new warning message** [416] for topics with internal inconsistencies with respect to the navigation hierarchy. These are usually caused by topics that have folder flags set when they should not.
- Adds support for generation of **XHTML1 conformant** HTML output.
- Adds a new HTML configuration option to switch between XHTML1 and HTML 4.01 output conformance. This setting can be found on settings page:

  > [Configuration]->Files & Format->Output Options->Encoding

- Adds a new option for **automatic wrapping of long lines in declaration syntax code**. If the code line gets longer than the threshold set up in the configuration, Doc-O-Matic will attempt to find a sensible location to break to a new line if possible. Wrapping is switched off by default. The new option can be found on settings page:

  > [Configuration]->AutoDoc->Code Options

- Adds a new feature for HTML based output named **Section Tabs**. Section tabs turn ordinary sequences of sections into tab-sets where only one section is visible at a time and where the user can click tabs to switch between sections. One application is **syntax sections tabs for multi-language syntax** output where each language appears in a tab. Syntax sections are tabbed sections in the new default configuration. A topic can have any number of tabbed sections. Sections in the same tab set that a configured to appear in a sequence are put into the same tab-set on the output page.

- Adds new formatting elements to configure the section tabs' appearance.
- **New visual style** of the default HTML configurations to closely match the style of the Visual Studio 2010 help. The default setup also adds unique Doc-O-Matic features like collapsible sections, neighbor topics, graphical class hierarchies and more and also include syntax tabs.
- Adds new **Corner Radius** formatting properties **for HTML** based output's border configuration of formatting elements. The border radius properties are supported in various browsers including Firefox, Safari and Internet Explorer 9. Browsers that do not support the border radius property simply ignore the settings and render the border in the usual manner.
- Adds two new variables for header and footer configuration of the PDF output.

  1. %PC% resolves to the page count of the entire document.
  2. %PCS% resolves to the page count of the section the page is contained in. Usually documents restart the page count in the main section and the index, so the first topic gets page 1 and the index is numbered differently (a,b,c, ...) also starting at 1. If for instance the document contains 4 pages before the first topic (welcome page, copyright page and TOC), 20 topic pages and 3 pages of index, %PC% will resolve to 27 while %PCS% will resolve to 4, 20 and 3 in the corresponding sections of the document.

- Adds a new option for PDF output to print page header and footer on the copyright page. This option is availabe on configuration settings page:

  > [Configuration]->Files & Format->Output Options

- The wildcard namespace lookup during Help 2 registration using TFHxReg is now case insensitive.
- Adds the `-m` command line option to TFHxReg that allows for merging of namespaces. This is useful if the automatic namespace merging that usually triggers upon first usage does not happen for some reason.

Noteworthy Fixes

- Fixes a crash during output generation for topics that exhibit internal inconsistencies, such as topics that are used as `COMBINE` targets which also have the folder flag set. This is a confirguration error that makes the topic unavailable as combine target and caused the crash in earlier releases. This and other cases now trigger a `[416]` warning message during output generation.

- Fixes an issue in the C# parser that could cause array specifications to be lost from property types when printing the property syntax.

```
public class C {
    public SomeType[] myProperty { ... };
};
```

- Fixes an issue with Pascal source code that uses anonymous methods or functions which may cause comments not being picked up from after the usage location in the implementation section of the unit.

- Fixes an issue with automatic linking when using flat namespace lists instead of namespace hierarchies which could cause some potential link targets not getting linked.

- Fixes an issue with association of comments in Visual Basic .NET enum declarations that make use of enum value assignments which could lead to comments being assigned to the following enum member instead of the one that it really belonged to.

- Fixes an issue with Delphi enum values that use parenthesized expressions like this:

```
TEnum = (
  Value1 = (1 + 2),
  Value2 = (Value1 + 2),
  Value3 = (Value2 + 2)
);
```

- Fixes an issue with comma separated field declarations in C/C++ where the first field's array specifier could be dropped:

```
typedef struct {
       int a[16], b[16];
} S;
```

- Fixes an issue with parsing Delphi abstract classes which were not handled correctly:

```
type
  TClass = class abstract(TObject)
    ...
  end;
```

- Fixes in the JavaScript file that is automatically generated for HTML based output formats.

- Fixes tab order issue on "Files & Folders" configuration settings page.

- Fixes an issue with C++, C# and Java classes that caused constructors and destructors of base classes to be listed in the method navigation tables when inherited member listings were switched on.

- Fixes an issue with XMLDoc see tags inside of tables that were not handled correctly and caused the see links to be ignored and tag content to be emitted as plain text.

- Fixes an issue with inserting documentation at the implementation location of class functions in

Delphi code that caused the comment to be inserted between `class` and `function` keywords:

```
class function TClass.Func(i: Integer): Integer;
```

- Fixes an issue with the body source extraction of Delphi class methods that caused the `class` keyword not to be included in the body source code printed in the output.
- Fixes an issue with code translation from Delphi to C# that incorrectly converted Delphi `out` parameters to C# `ref` parameters instead of using C# `out`.
- Fixes an issue with the default PDF configuration that incorrectly set the text alignment of the left-aligned table cell formatting element to justified alignment. This can be fixed in existing projects by setting the alignment property of the formatting element `Table Cell Left` to the correct value (Left) on the PDF configuration settings page

  > [Configuration]->Files & Format->Page Layout->Formatting.

- Fixes an issue with parsing nested C++ template arguments in the form that could cause incorrect parsing from the template declaration down the rest of the source file:

```
typedef T < A, B < C >> newT;
const char *invisible;
```

- Fixes an issue with building the class hierarchy from C++ classes that use template classes as ancestors:

```
template <T> class Base {}
class Sub : Base<x> {}
```

- Fixes an issue with building the class hierarchy from Java classes that use generic classes as ancestors:

```
public class Base<T>{}
public class Sub extends Base<x> {}
```

- Fixes various issues with standard compliance of generated HTML files. Doc-O-Matic generated XHTML files, HTML 4 files, JavaScript and CSS files are now standards compliant and well formed.
- Fixes a problem with syntax highlighting of declaration source code that could lead to incorrectly formatted output.
- Fixes a problem with the German localization of the main window's toolbar items.