UserDocsApiLinker Requirements

# Abstract

We want to generate two sets (.NET and Java) of user documentation from single documentation. To do this, there are two tasks that need to be automated:

* Filtering of platform specific text (run, paragraphs, tables etc). This is done by a separate tool called **UserDocsSplitter** and discussed elsewhere.
* Displaying API member names (classes, methods, properties etc) in the document correctly according to the platform. Also linking API member names to the API pages. This is done by **UserDocsApiLinker** and discussed in this document.

# UserDocsApiLinker Overview

This tool performs the following steps:

1. Find references to API members in the documentation text. These references are just text formatted in a special way. Let’s say text formatted using the **ApiLink** *character style*.
2. Replace references to API members with hyperlinks that display text appropriate for the .NET or Java platform and link to appropriate pages in the API documentation. For example **Font.NameBi** should become **Font.NameBi** for .NET and **Font.getNameBi** for Java and linked to [http://www.aspose.com/documentation/.net-components/aspose.words-for-.net-and-java/**aspose.words.font.namebi.html**](http://www.aspose.com/documentation/.net-components/aspose.words-for-.net-and-java/aspose.words.font.namebi.html)and [http://www.aspose.com/documentation/.net-components/aspose.words-for-.net-and-java/**com/aspose/words/font.html#NameBi**](http://www.aspose.com/documentation/.net-components/aspose.words-for-.net-and-java/com/aspose/words/font.html#NameBi) correspondingly.

The details of replacements are discussed below.

# Requirements

## R1 – References to API Members

This table describes requirements about how to reference API members from the documentation.

| Id | Name | Description |
| --- | --- | --- |
| R1.1 | Api Link Style | References to API members are formatted using the **ApiLink** character style in Microsoft Word.  The system needs to find all runs of text formatted using the **ApiLink** style and process them. In the following example the bold text is formatted using the **ApiLink** style.  Example: *Use the DocumentVisitor.VisitDocumentStart method to do something.* |
| R1.2 | Api Link Multiple Runs | Due to Microsoft Word behavior, one reference to an API member can consist of several consecutive runs formatted using the **ApiLink** style and the system must support that.  In the following example the bold text is formatted using the ApiLink style, but it consists of several runs (because it has other different formatting attributes). MS Word can split text into runs at any point even if the formatting is the same. The system should find all consequtive **ApiLink** runs and process them as one.  Example: *Use the DocumentVisitor.VisitDocumentStart method to do something.* |
| R1.3 | Api Link Friendly Syntax | The Api Link can be entered in a number of ways inside a document. This is called the “friendly syntax” because it makes it easy for the author of the document to link to API members by typing their names easily.  Examples of different references that all produce the same result: *Use the DocumentVisitor.VisitDocumentStart method to do something.*  *Use the Aspose.Words.DocumentVisitor.VisitDocumentStart method to do something.*  *Use the DocumentVisitor.VisitDocumentStart(Document) method to do something.*  *Use the DocumentVisitor.VisitDocumentStart(Aspose.Words.Document) method to…*  Below are some more examples using the “friendly syntax” that must be supported. |
| R1.3.1 | Link to Type | DocumentVisitor  Aspose.Words.DocumentVisitor |
| R1.3.2 | Link to Method | DocumentVisitor.VisitDocumentStart  DocumentVisitor.VisitDocumentStart(Aspose.Words.Document) |
| R1.3.3 | Link to Ctor | Document.#ctor  Document.#ctor(System.String)  Aspose.Words.Document.#ctor |
| R1.3.4 | Link to Property | Font.NameBi  Aspose.Words.TabStopCollection.Item(System.Int32) |
| R1.3.5 | Link to Enumeration Value or Field | ParagraphAlignment.Left  Aspose.Words.ParagraphAlignment.Left |
| R1.3.6 | Link to Event | SaveOptions.HtmlExportImageSaving |
| R1.4 | Reuse Code for Friendly Syntax from ExampleManager | We already have the “friendly syntax” defined and supported in the [ExampleManager](http://romeok.planetdns.net:8080/display/org/ExampleManager.exe) tool that injects code examples into the API documentation.  You need to refactor ExampleManager code to extract and reuse the logic that parses API references and matches them to correct members in the assembly using reflection. |

## R2 – Display Names

This table describes requirements how to replace references to API members with display text.

Note in these tables for clarity I use the “full syntax” that is emitted by C# compiler into XML documentation. E.g. T: - type, M: - member, P: - propery, F: - field or enum value, E: - event.

| Id | Name | Description |
| --- | --- | --- |
| R2.1.1 | Type Name .NET | T:Aspose.Words.DocumentVisitor -> **DocumentVisitor**  T:Aspose.Words.ParagraphAlignment -> **ParagraphAlignment**  T:Aspose.Words.INodeCollection -> **INodeCollection** |
| R2.1.2 | Method Name .NET | M:Aspose.Words.DocumentVisitor.VisitDocumentStart(Aspose.Words.Document) -> **DocumentVisitor.VisitDocumentStart**  M:Aspose.Words.Document.RemoveAllChildren -> **Document.RemoveAllChildren**  Example: *Use the DocumentVisitor.VisitDocumentStart method to do something.* |
| R2.1.3 | Ctor Name .NET | M:Aspose.Words.Document.#ctor -> **Document**  M:Aspose.Words.Document.#ctor(System.String) -> **Document**  Example: *Use the* ***Document*** *constructor to do something.* |
| R2.1.4 | Property Name .NET | P:Aspose.Words.Font.NameBi -> **Font.NameBi**  P:Aspose.Words.TabStopCollection.Item(System.Int32) -> **TabStopCollection.Item**  Example: *Use the* ***TabStopCollection.Item*** *indexer to get the required value.* |
| R2.1.5 | Enum Value Name .NET | F:Aspose.Words.ParagraphAlignment.Left -> **ParagraphAlignment.Left** |
| R2.1.6 | Event Name .NET | E:Aspose.Words.SaveOptions.HtmlExportImageSaving -> **SaveOptions.HtmlExportImageSaving** |
| R2.2.1 | Type Name Java | class T:Aspose.Words.DocumentVisitor -> **DocumentVisitor**  enum T:Aspose.Words.ParagraphAlignment -> **ParagraphAlignment**  interface T:Aspose.Words.INodeCollection -> **INodeCollection** |
| R2.2.2 | Method Name Java | Note camelCasing for Java method names.  M:Aspose.Words.DocumentVisitor.VisitDocumentStart(Aspose.Words.Document) -> **DocumentVisitor.visitDocumentStart**  M:Aspose.Words.Document.RemoveAllChildren -> **Document.removeAllChildren**  Example: *Use the DocumentVisitor.visitDocumentStart method to do something.* |
| R2.2.3 | Ctor Name Java | M:Aspose.Words.Document.#ctor -> **Document**  M:Aspose.Words.Document.#ctor(System.String) -> **Document**  Example: *Use the* ***Document*** *constructor to do something.* |
| R2.2.4.1 | Property Name Java Getter/Setter | P:Aspose.Words.Font.NameBi -> **Font.getNameBi/setNameBi**  Example: *Use the* ***Font.getNameBi/setNameBi*** *property to do something.* |
| R2.2.4.2 | Property Name Java Getter Only | This is the case where the property is ready only.  P:Aspose.Words.Font.Border -> **Font.getBorder**  Example: *Use the* ***Font.getBorder*** *property to do something.* |
| R2.2.4.3 | Property Name Java Indexer | P:Aspose.Words.TabStopCollection.Item(System.Int32) -> **TabStopCollection.get/set**  Example: *Use* ***TabStopCollection.get/set*** *to get the required value.* |
| R2.2.4.4 | Property Name Java Boolean | Some Boolean property names do not have the “get” prefix in the Java API.  P:Aspose.Words.CompositeNode.HasChildNodes -> **CompositeNode.hasChildNodes**  If the property name starts with: “Is”, “Has”, “Can”, “Should”, “Could”, “Will”, “Shall”, “Contains”, “StartsWith”, “EndsWith” then the display name should not have the “get” prefix. |
| R2.2.4.5 | Property Name Java Indexer Overloaded | T:Aspose.Words.HeaderFooterCollection.Item(System.Int32) -> **HeaderFooterCollection.get**  T:Aspose.Words.HeaderFooter.Collection.Item(Aspose.Words.HeaderFooterType) -> **HeaderFooterCollection.getByHeaderFooterType**  This is caused by the fact that .NET enums are ported to Java as integer constants. This will create two indexer properties with the same signature. To avoid that, the second getter is renamed.  The instructions how to rename are specified using special tags inside the C# XML documentation.  TODO 0 RK Add wiki pages about Java doc tools and think how to link with that info. |
| R2.2.5 | Enum Value Name Java | F:Aspose.Words.ParagraphAlignment.Left -> **ParagraphAlignment.LEFT**  F:Aspose.Words.BreakType.PageBreak -> **BreakType.PAGE\_BREAK**  F:Aspose.Words.PaperSize.A4 -> **PaperSize.A4**  F:Aspose.Words.PaperSize.Paper10x14 -> **PaperSize.PAPER\_10\_X\_14** |
| R2.2.6 | Event Name Java | E:Aspose.Words.SaveOptions.HtmlExportImageSaving -> throw an exception  There are no events in Java. We implement these using interfaces in Java. Such text must be platform specific and filtered out at the **UserDocSplitter** stage. UserDocsApiLinker should throw an exception if encounters a reference to an event while generating documentation for Java. |

## R3 – Hyperlinks

This table specifies what URLs the system needs to insert when replacing references to API members in the documentation.

|  |  |  |
| --- | --- | --- |
| Id | Name | Description |
| R3.1 | Per Topic | All items discussed below must be performed on the “per topic” basis. The user documentation is later split into topics at every paragraph formatted using the styles Heading 1 – Heading 4.  E.g. if you encounter a paragraph formatted using one of those styles, you need to reset the state and start processing a new topic. |
| R3.2 | Link Once Per Topic | If an API member is referenced multiple times from within one topic, only the first occurrence should be hyperlinked. All other occurrences should be just made bold display text.  Example: *Use one of the* [*Document.Save*](file:///C:\Users\User\AppData\Local\Microsoft\Windows\Temporary%20Internet%20Files\Content.Outlook\EYQ0897I\aaa) *methods to save the document.* ***Document.Save*** *allows to save to a file, stream or to a client browser.* |
| R3.3 | .NET Documentation HTML and CHM targets | The documentation for Aspose.Words for .NET is generated into two targets:  HTML – this is a set of HTML files uploaded to the Aspose website for online viewed.  CHM – this is a compiled help file included in the Aspose.Words download.  The hyperlinks generated by the tool need to work correctly in both targets. To achieve this, I suggest the hyperlinks should be relative, not absolute. |
| R3.4 | Link to Aspose.Words for .NET API Pages | The Aspose.Words for .NET API documentation is generated using [Aspose.NDoc](http://romeok.planetdns.net:8080/display/org/NDocConsole.exe) (a modified version of NDoc).  You need to examine the rules NDoc follows to generate names for HTML pages and apply the same. |
| R3.4.1 | Validate Api Links | Use reflection on Aspose.Words.dll to actually find that the corresponding API member exists in the assembly. If the member cannot be found (e.g. the reference in the documentation is incorrect), throw an exception. |
| R3.4.2 | Type Name .NET | T:Aspose.Words.DocumentVisitor -> http://www.aspose.com/documentation/.net-components/aspose.words-for-.net-and-java/**aspose.words.documentvisitor.html** |
| R3.4.3 | Method Name .NET | M:Aspose.Words.DocumentVisitor.VisitDocumentStart(Aspose.Words.Document) -> http://www.aspose.com/documentation/.net-components/aspose.words-for-.net-and-java/**aspose.words.documentvisitor.visitdocumentstart.html** |
| R3.4.3.1 | Method Name .NET Overload | If a method has overloads, then the HTML page name will have a suffix, such as \_2, \_3 and so on.  M:Aspose.Words.Document.Save(System.String, Aspose.Words.SaveFormat) -> http://www.aspose.com/documentation/.net-components/aspose.words-for-.net-and-java/**aspose.words.document.save\_overload\_2.html**  Possible Solution 1: Use reflection on Aspose.Words.dll to find all Save methods and find the matching signature. The first method in the order of declaration will have no suffix, the second will have suffix \_2 etc.  Possible Solution 2: Alternatively, you can change Aspose.NDoc code to generate HTML method names using the signatures. For example instead of generating **aspose.words.document.save\_overload\_2.html** you could generate **aspose.words.document.save.system.string.aspose.words.saveformat.html** |
| R3.4.4 | Ctor Name .NET | M:Aspose.Words.Document.#ctor -> http://www.aspose.com/documentation/.net-components/aspose.words-for-.net-and-java/**aspose.words.documentconstructor1.html**  M:Aspose.Words.Document.#ctor(System.String) -> http://www.aspose.com/documentation/.net-components/aspose.words-for-.net-and-java/**aspose.words.documentconstructor2.html**  Maybe you should use the Possible Solution 2 from the above for methods and constructors. |
| R3.4.5 | Property Name .NET | P:Aspose.Words.HeaderFooterCollection.Item(System.Int32) -> http://www.aspose.com/documentation/.net-components/aspose.words-for-.net-and-java/**aspose.words.headerfootercollection.item1.html**  P:Aspose.Words.HeaderFooterCollection.Item(Aspose.Words.HeaderFooterType) –> http://www.aspose.com/documentation/.net-components/aspose.words-for-.net-and-java/**aspose.words.headerfootercollection.item2.html**  Similar to the above. I would recommend chaning Aspose.NDoc to generate **aspose.words.headerfootercollection.item.aspose.words.headerfootertype.html** instead of **aspose.words.headerfootercollection.item2.html** |
| R3.4.6 | Enum Value Name .NET | F:Aspose.Words.ParagraphAlignment.Left -> http://www.aspose.com/documentation/.net-components/aspose.words-for-.net-and-java/**aspose.words.paragraphalignment.html**  Note linking to the enum page, not to a particular enum value. |
| R3.4.7 | Event Name .NET | E:Aspose.Words.SaveOptions.HtmlExportImageSaving -> http://www.aspose.com/documentation/.net-components/aspose.words-for-.net-and-java/**aspose.words.saveoptions.htmlexportimagesaving.html** |
|  |  | TODO 0 RK Java hyperlinks requirements. |
|  |  |  |
|  |  |  |