



DITA-OT plugin

- Introduction..... 1
- System requirements..... 2
- Installing the Miramo DITA-OT plugin..... 3
- Files and folders..... 4
- Running the demo..... 5
- Further information or help.... 13
- Installing and integrating the plugin with oXygen..... 14

Introduction

Thank you for your interest in the Miramo DITA Open Toolkit (OT) plugin, **dita2mmpdf**, which provides a simple method for generating high-quality PDFs from DITA on Windows systems, where formatting is controlled via a GUI template design tool, **MiramoDesigner**.

This plugin uses the **mmComposer** engine to render XML to PDF. The plugin provides a mapping from DITA to the MiramoXML model using the **dita2mm** script, then applies template-controlled formatting using **mmComposer**.

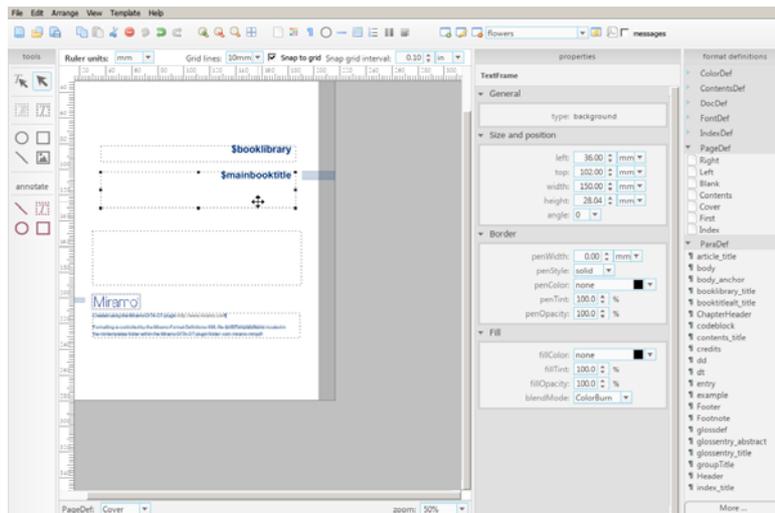
This README file describes how to run the **dita2mmpdf** demo, which produces a PDF file from the **flowers.ditamap** file located in the **samples/flowers** folder (see page 5).

Alternatively, the Miramo DITA-OT plugin may be run via **ant** using the 'dita2mmpdf' ant target (see page 9) or by invoking the 'mmpdf' transformation type using the DITA-OT 'dita -f mmpdf' command line tool (see page 8).

If you would like more information about **server versions** of Miramo, or would like to produce different output formats (FrameMaker, ePub, Kindle, HTML5 ...) please contact us (email miramo@datazone.com, phone +353 64 66 28964), and/or visit our website: <http://www.miramo.com>.

GUI template controlled formatting

The appearance of the output PDF is controlled by a **Miramo Format Definitions (MFD)** template - *the separation of form and content*. Templates may be created and modified using **MiramoDesigner** (*MiramoDesigner.exe*), a GUI page design tool supplied as part of the Miramo distribution version.



The default template (*default.mfd*) provides a single-column layout for rendering MiramoXML to PDF: [flowers_default.pdf](#)



Installing the plugin

System requirements.....2
Installing from the setup.exe
(recommended) 3
Installing from the ZIP file
(advanced)..... 3
Files and folders4

System requirements

Before installing the Miramo DITA-OT plugin the following software must be installed and working on the target system:

- *DITA-OT* version 1.8.5 or above
- *MiramoPDF* or *MiramoEnterprise* vs 1.3 or above

MiramoPDF Personal and Desktop editions run on the following operating systems:

- Windows 7 Professional (64 bit only)
- Windows 8.1 (64 bit only)
- Windows 10

MiramoPDF Server and Miramo Enterprise editions are available to run on Windows Server operating systems - contact us for more information.

Installing the Miramo DITA-OT plugin

These instructions relate to installing and integrating the Miramo DITA-OT plugin into a standalone DITA-OT installation.

To integrate with your XML authoring tool or DITA CMS, follow the instructions given by your vendor. For information on installing the plugin for use with oXygen, see “[Integrating with oXygen](#)” on page 14.

See “[Files and folders](#)” on page 4 for a list of files and folders installed in the DITA-OT **plugins/com.miramo.mmpdf** folder.

Installing from the setup.exe (recommended)

Double-click on the **MiramoDITA-OTplugin<vs>setup.exe** and navigate to the DITA-OT install root folder, which contains the ‘integrator.xml’ file. The MiramoDITA-OT plugin source (including samples, command scripts and templates) will be located in the `[dita-ot install dir]/plugins/com.miramo.mmpdf` folder, for example:

```
C:\program files (x86)\dita-ot-2.5.1\plugins\com.miramo.mmpdf
```

The **MiramoDITA-OTplugin<vs>setup.exe** will run the DITA-OT integrator and will append the `com.miramo.mmpdf` folder to the PATH environment variable. The plugin is now ready to use.

Installing from the ZIP file (advanced)

Extract the **MiramoDITA-OTplugin<vs>.zip** file into the DITA-OT plugins folder, which contains source for all the DITA-OT plugins, for example `org.dita.docbook`.

This will create a folder ‘`com.miramo.mmpdf`’ in the `[dita-ot install dir]\plugins` folder containing the files and folders listed in “[Files and folders](#)” on page 4. Next, run the DITA-OT integrator:

Right-click on the `[dita-ot install dir]\startcmd.bat` and choose runAs Administrator to start a DITA-OT console window, and enter the following command:

```
ant -f integrator.xml
```

Alternatively, for DITA-OT version 2.3 and above, log in as the administrator user and run the following command: (note the double ‘-’ before the ‘install’)

```
[dita-ot install dir]\bin\dita --install MiramoDITA-OTplugin<version>.zip
```

Set the PATH environment variable to include the `[dita-ot install dir]/plugins/com.miramo.mmpdf` folder which contains the **dita2mmpdf.cmd** script to be able to run the script from any folder.

See www.dita-ot.org for more information on installing DITA-OT plugins.

Files and folders

The dita2mmpdf plugin includes the following files and folders, located in the DITA-OT installation folder selected during installation:

<dita-ot-installdir>/plugins/com.miramo.mmpdf/

Source for dita2mmpdf plugin:

docs/Miramo_DITA-OT_Readme.pdf

This document

oxygen/MiramoPDF.scenarios

The oXygen transformation scenario for producing a PDF from the currently edited DITA file. Import into your oXygen tool using Options->import transformations

dita2mm.cmd

Command script for preprocessing a DITA input file to MiramoXML, ready for rendering to PDF. Called by:

dita2mmpdf.cmd

Command script for running the flowers demo, or for producing a PDF from a specified DITA map.

xsl/

XSL stylesheets for mapping DITA to MiramoXML

mmtemplates/

Miramo Format Definitions template:

default.mfd

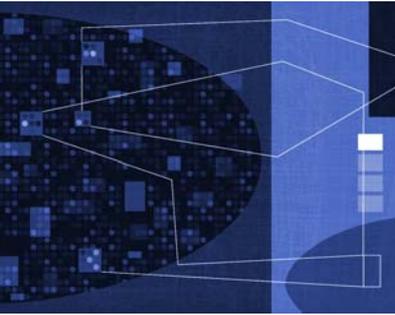
Miramo Format Definitions templates are XML files which can be edited using the Miramo page layout editor, **MiramoDesigner**.

build_dita2mmpdf.xml

XML file which contains the **ant** targets for performing topicmerge and translating the DITA content to MiramoXML (Miramo Simple Markup, which represents the document content)

samples/flowers/

The oXygen DITA flowers sample



Using the plugin

- Running the demo..... 5
- Changing the appearance of the PDF via the MFD template..... 6
- Creating a PDF using *dita2mmpdf*..... 7
- Creating a PDF using the dita command 8
- Creating a PDF using *ant*..... 9
- Controlling formatting using DITA @outputclass values 11
- Further information or help.... 13

Running the demo

Start a console window, or powershell window and enter:

```
dita2mmpdf -dev
```

If your PATH environment variable is set correctly this will run the *dita2mmpdf.cmd* script in the *[dita-ot install dir][dita-ot install dir]/plugins/com.miramo.mmpdf* folder.

(** NOTE that the first time you run the demo it may take several minutes as mmComposer builds its internal font cache)

This calls the **dita2mm** script to perform a topic merge on the flowers.ditamap, and to produce an intermediate flowers_miramo.xml file. The resulting XML is processed to PDF by Miramo.

When Miramo has finished creating the PDF you should see a message like this in the console window:

```
mmComposer: 1 : Completed with 0 errors and 0 warnings.  
PDF file <com.miramo.mmpdf>\samples\flowers\flowers.pdf created  
Finished.
```

Changing the appearance of the PDF via the MFD template

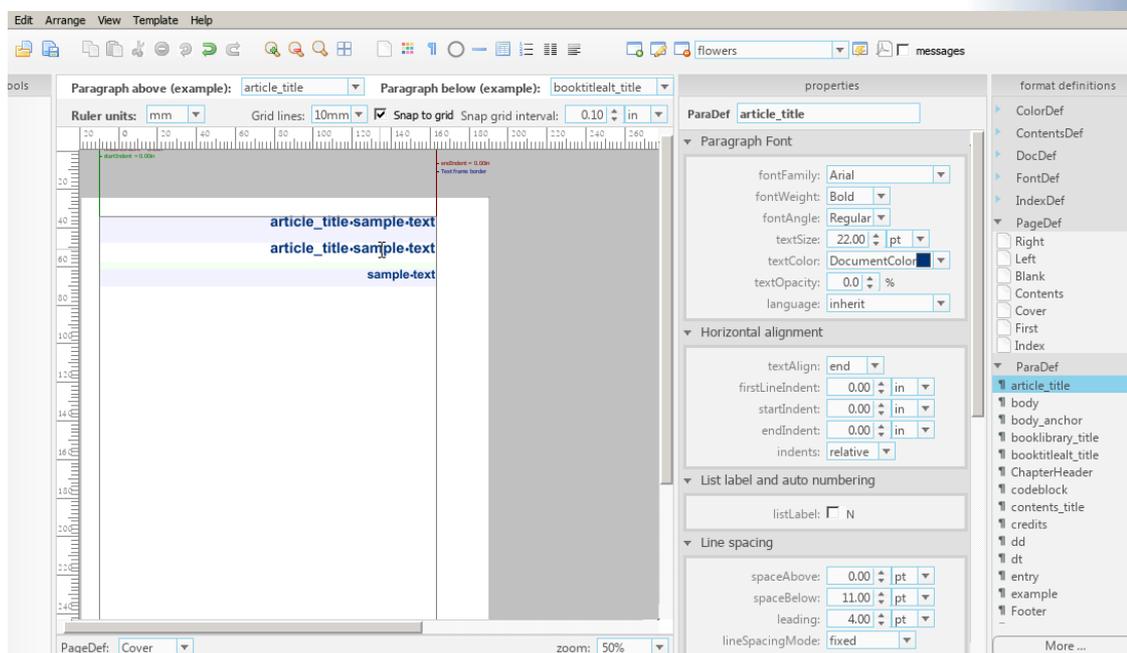
The appearance of the PDF output files is controlled by selecting one of the Miramo MFD template files located in the `<dita_install_dir>/plugins/com.miramo.com.miramo.mmpdf/mmtemplates` folder.

These are created and modified using **MiramoDesigner** which is a GUI page design tool supplied as part of the Miramo distribution version.

The default template (*default.mfd*) provides a single-column layout for rendering MiramoXML to PDF: [flowers_default.pdf](#)

Using MiramoDesigner to modify MFD files

Here is a screenshot of the main **MiramoDesigner** window, showing the properties of the *article_title* paragraph, which can be altered as required.



Creating a PDF using *dita2mmpdf*

The dita2mmpdf script has the following usage:

```
dita2mmpdf [-dev] [-ditaval file.ditaval] [ditafile [template.mfd [pdffile]]]
```

Using a different template and/or DITAVAL filter

To run the flowers sample through using a different 'twocol.mfd' template (*not supplied*) which you have created in the mmtemplates folder, use this command from the plugins/com.miramo.mmpdf subfolder in the DITA-OT install folder:

```
dita2mmpdf samples/flowers/flowers.ditamap twocol.mfd
```

This will create the file samples/flowers/flowers.pdf (the PDF file defaults to the basename of the DITA file, with a .pdf extension) using the 'twocol.mfd' template (*not supplied*). To create a differently named PDF, specify it as the last argument to dita2mmpdf, for example:

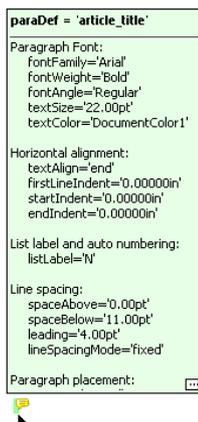
```
dita2mmpdf samples/flowers/flowers.ditamap twocol.mfd ff.pdf
```

To apply a DITAVAL filter 'print.ditaval' (*not supplied*) use the -ditaval command line option, eg:

```
dita2mmpdf -ditaval print.ditaval samples/taskbook.ditamap
```

Using the -dev option to display formatting properties in the output PDF

The PDF files created using **dita2mmpdf** can be set to contain tooltips which give formatting information for paragraph tags (green), font tags (red), and table tags (blue). Here is an example showing the **article_title** paragraph in the flowers.pdf file produced using the default template:



Growing Flowers

The tooltips are shown if the **-dev** command line option is supplied to the dita2mmpdf.cmd. (The dita2mmpdf -dev option sets the MiramoPDF command line option '-showProperties Y'). By default tooltips are not shown.

Using a different DITA input file

To run the DITA file through to a pdf with the same basename as the DITA file, use:

```
dita2mmpdf <path_to_ditafile>
```

Or add the appropriate template file and output PDF name, eg:

```
dita2mmpdf sample.dita mytemplate.mfd mysample.pdf
```

In this case 'mytemplate.mfd' must exist in the plugins/com.miramo.mmpdf/mmtemplates folder.

Creating a PDF using the *dita* command

You can use either the *dita* command-line tool or **ant** to transform DITA content to PDF using the Miramo DITA-OT plugin:

```
dita -f mmpdf -i <path_to_ditafile> [options]
```

The above assumes that folder containing *dita.bat* is included in the PATH environment variable, for example using the following powershell command:

```
$env:PATH += " ;$env:MM_DITA_DIR/bin"
```

Here is an example of using the '*dita*' -f mmpdf command line tool to publish *flowers.ditamap* to a PDF in the default output folder, './out':

```
dita -f mmpdf -i flowers.ditamap
```

Any of the ant parameters used by the Miramo DITA-OT plugin on page 9 may be specified, for example to include PDF tooltip properties in the output PDF, and to write to pdf file '*myflowers.pdf*' in the current folder:

```
dita -f mmpdf -i flowers.ditamap "-DshowProperties=Y"
"-Dpdf.file=myflowers.pdf"
```

See *dita command arguments and options* on <http://www.dita-ot.org> for more information., and *Table 1:ant parameters used by the Miramo DITA-OT plugin* on page 9 for a list of supported parameter values.

Creating a PDF using *ant*

Alternatively you can use **ant** to build the 'dita2mmpdf' target using the following syntax (powershell):

```
ant -f "$env:MM_DITA_DIR/build.xml" ["-Dparam=value" "..."] dita2mmpdf
```

The above assumes that folder containing ant.bat is included in the PATH environment variable, for example using the following powershell command:

```
$env:PATH += ";$env:MM_DITA_DIR/bin" # powershell
```

Here is an example creating flowers.pdf from the samples/flowers folder by running this command from the powershell:

```
cd "env:MM_DITA_DIR"
ant -f "$env:MM_DITA_DIR/build.xml" "-Dargs.input=samples/flowers/flowers.ditamap" "-DshowProperties=Y" "-Dpdf.file=flowers.pdf" dita2mmpdf
```

Here is another example, where the test.ditamap is located in "c:\users\Joanne\documents", and the MFD file is called 'test.mfd' and is located in c:\users\Joanne\documents:

```
cd c:\users\Joanne\documents
ant -f "$env:MM_DITA_DIR/build.xml" "-Dargs.input=samples/flowers/flowers.ditamap" "-DshowProperties=Y" "-Dpdf.file=flowers.pdf" dita2mmpdf
```

Table 1: ant parameters used by the Miramo DITA-OT plugin

ant parameter	Default value	Description
dita.dir	dita install folder	Absolute path to the DITA Open Toolkit that is being used.
args.input	<i>required</i>	Path to the source content being published.
mfd.file	default.mfd	Name of MiramoDesigner MFD template file used to control output formatting, located in \${dita.dir}\plugins\com.miramo.mmpdf\mmtemplates, or value specified by mfd.dir parameter.
mfd.dir	\${dita.dir}\plugins\com.miramo.mmpdf\mmtemplates	Folder containing \${mfd.file} template
pdf.file	input file with file extension replaced by .pdf	Path to the PDF file to be created
showProperties	Y	If this property is set to Y , the PDF output file will contain PDF tooltips describing paragraph, font and table formats. Set to N for production. Note that PDF tooltips may not be visible in non-Adobe PDF viewers.
transtype	mmpdf	mmpdf - do not modify
cwd	folder containing input file	Current working folder for the mmComposer process - sometimes useful for resolving relative URLs
clean.temp	yes	Specifies whether to remove temporary intermediate files. Allowed values: yes, no



Table 1: ant parameters used by the Miramo DITA-OT plugin

ant parameter	Default value	Description
dita.temp.dir	mmtmp	Specifies temporary folder for intermediate files

Controlling formatting using DITA @outputclass values

The Miramo DITA-OT plugin allows page layouts, covers, TOC and Index generation to be controlled using one or more 'mmpdf:' @outputclass values, as described in the table below:

Table 2: DITA @outputclass values

@outputclass value	DITA elements	Description
mmpdf:noCover	map, bookmap	Suppresses generation of Cover page
mmpdf:noTOC	map, bookmap	Suppress TOC generation
mmpdf:noIndex	map, bookmap	Suppress index generation
mmpdf:span	title, p, note	Spans title across all columns in multi-column document
mmpdf:pageBreak	title, p, note	breaks page before title/p/note paragraph
mmpdf:columnBreak (alias for break-before)	title, p, note	force title/p/note paragraph to top of next column (or page, for single-column documents)
mmpdf:section: <i>name</i>	map, bookmap, title, topic, chapter, appendix, booktitle	start new section using SectionDef 'name'. SectionDef is defined in the MFD template or using <SectionDef sectionDef="name"../> and starts a new page layout sequence

Cover pages

By default, cover pages are included under the following conditions:

map: map element has @title attribute or child **<title>** element

bookmap: bookmap element contains child **<booktitle>** element

To suppress cover generation in the above cases, set @outputclass to **mmpdf:noCover**

Table of Contents (TOCs)

By default, TOCs are included under the following conditions:

map: map element has @title attribute or child **<title>** element

bookmap: bookmap element contains a <booklists> <toc> element, for example:

```
<frontmatter>
  <booklists>
    <toc/>
  </booklists>
```

To suppress cover generation in the above cases, set @outputclass to **mmpdf:noTOC**

Indexes

By default, Indexes are included under the following conditions:

DITA source contains one or more **<indexterm>** elements

bookmap: bookmap element contains a <booklists> <indexlist> element, for example:

```
<backmatter>
  <booklists>
    <indexlist/>
```



```
</booklists> </booklists>
```

To suppress index generation in the above cases, set @outputclass to **mmpdf:noIndex**



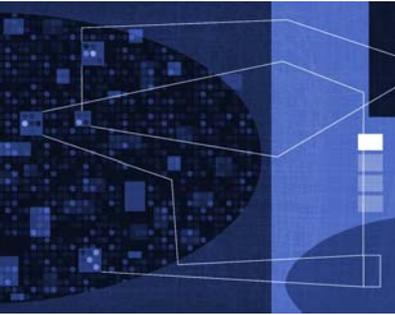
Further information or help

See the Miramo mmComposer Reference Guide for more information about the Miramo simple markup: [click here to view](#)

See the MiramoDesigner videos on the Miramo Datazone YouTube channel for more information about MiramoDesigner: [click here to view](#)

Check our website: <http://www.miramo.com> or phone us on +353 64 66 28964.

And you are always welcome to email miramo@datazone.com with comments, criticisms and questions - your feedback would be much appreciated.



Integrating with oXygen

- Installing from the setup.exe (recommended) 14
- Installing from the setup.exe (recommended) 14
- Installing from the ZIP file (advanced) 14
- Configuring oXygen..... 16

Installing and integrating the plugin with oXygen

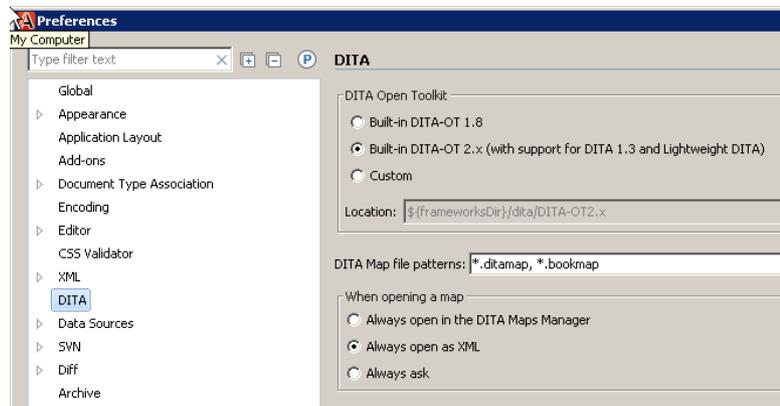
Installing from the setup.exe (recommended)

Double-click on the **MiramoDITA-OTplugin<vs>setup.exe** and navigate to the DITA-OT install root folder.

For oXygen (version 18 or above), the built-in DITA-OT folder is located in [OXYGEN_INSTALL_DIR]/frameworks/dita/DITA-OT2.x, for example:

```
C:\program files\Oxygen XML Editor 19\frameworks\dita\DITA-OT2.x
```

This corresponds to the oXygen DITA Open Toolkit 'Built-in DITA-OT 2.x' location preference shown here (options->preferences->DITA):



Note that if you choose a different folder you will have to set the **dita.dir** variable in the MiramoPDF transformation scenarios, or change the oXygen DITA-OT configured install folder in the preferences.

The MiramoDITA-OT plugin source (including samples, command scripts and templates) will be located in the [dita-ot install dir]/plugins/com.miramo.mmpdf folder, for example:

```
[OXYGEN_INSTALL_DIR]frameworks\dita\DITA-OT2.x\plugins\com.miramo.mmpdf
```

The **MiramoDITA-OTplugin<vs>setup.exe** will run the DITA-OT integrator and will append the com.miramo.mmpdf folder to the PATH environment variable.

Installing from the ZIP file (advanced)

Extract the **MiramoDITA-OTplugin<vs>.zip** file into the DITA-OT plugins folder.

For oXygen, the built-in DITA-OT plugins folder is located in [OXYGEN_INSTALL_DIR]/frameworks/dita/DITA-OT2.x/plugins, for example:

```
C:\program files\Oxygen XML Editor 19\frameworks\dita\DITA-OT2.x\plugins
```



Running the DITA-OT integrator [oXygen]

Start your preferred oXygen product, open a DITA file or map in the editor window and run the predefined transformation scenario called **Run DITA OT Integrator**: execute it from the Document->Transformation->Apply Transformation Scenario(s) dialog box. If the integrator is not visible, select the Show all scenarios option that is available in the Settings drop-down menu.

Files and folders

See “Files and folders” on page 4 for a list of files and folders which are installed in the plugins\com.miramo.mmpdf folder

Configuring oXygen

Import MiramoPDF transformation scenarios [oXygen]

Once the com.miramo.mmpdf folder is installed in the DITA-OT plugins folder, set up the MiramoPDF transformation scenario: Choose options->import transformation scenarios, then navigate to the **MiramoPDF.scenarios** file located in the com.miramo.mmpdf/oxygen subfolder (by default, [OXYGEN_INSTALL_DIR] **frameworks/dita/DITA-OT2.x/plugins/com.miramo.mmpdf/oxygen**).

This creates the **MiramoPDF** and **MiramoPDF dev** (with PDF tooltips) transformation scenarios, which are set up to produce a PDF from the currently edited DITA file, and which can be applied and modified as required. See "ant parameters used by the Miramo DITA-OT plugin" on page 9 for more information.

Make oXygen's content completion present the @outputclass values

To configure oXygen's content completion, it is necessary to copy and paste the **match** elements contained in the cc_config_miramo.xml file supplied to the DITA framework you are using.

The cc_config_miramo.xml file is located here:

```
[oxygen_install_dir]/frameworks/DITA-OT2.x/plugins/com.miramo.mmpdf/oxygen/cc_config_miramo.xml
```

and contains entries like this:

```
<match elementName="p" attributeName="outputclass">
  <items action="append">
    <item value="mmpdf:span" annotation="Spans title across all columns in multi-column document"/>
    <item value="mmpdf:pageBreak" annotation="breaks page before title/p/note paragraph"/>
    <item value="mmpdf:columnBreak" annotation="force title/p/note paragraph to top of next column
(or page, for singlecolumn documents)"/>
    <item value="mmpdf:section:name" annotation="start new section using SectionDef 'name'.
SectionDef is defined in the MFD template"/>
  </items>
</match>
```

Inside your DITA framework there is a configuration file named cc_config.xml. If you are using the built-in DITA framework then this file is located at **[OXYGEN_INSTALL_DIR]/frameworks/dita/resources/cc_config.xml**.

If you don't know exactly what framework you are using or where it is located then you can find out like this:

1. Go to Window->Show view and open the Properties view.
2. Open a DITA topic.
3. Look inside the Properties view at the Document type name. Now you know the name of your framework.
4. Go to Options->Preferences... on the Document Type Associations page, identify and edit the framework with the previously discovered name. The Storage field will reveal its location.

The <match> entries for allowing content completion for the MiramoPDF @outputclass values are located in the `[oxygen_install_dir]/frameworks/DITA-OT2.x/plugins/com.miramo.mmpdf/oxygen/cc_config_miramo.xml` file. Open it and copy and paste the entries inside the <config> root element in your DITA framework cc_config.xml file.