



# DITA-OT plugin

Introduction:.....	1
Files and folders: .....	2
Installing the plugin: .....	2
Running the demo: .....	4
Command line options:.....	4
Using the DITA-OT mmpdf output transformation: .....	6
Controlling formatting using DITA @outputclass values: .....	8
Further information or help:.....	8

## 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.

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

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 [support@datazone.com](mailto:support@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. Alternatively, if you prefer to use FrameMaker as a template editor, contact us for more information on our FrameMaker MFD export plugin.

See "Using the -dev option to display formatting properties in the output PDF" on page 5 for information on using the **-dev** command line option when developing a new template.

The default template (*default.mfd*) provides a single-column layout for rendering MiramoXML to PDF: *flowers\_ default.pdf*

## System requirements

Miramo must be **downloaded and installed** before installing the dita2mmpdf DITA-OT plugin.

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

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

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



## DITA-OT plugin

### Files and folders

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

#### **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** or saved out from a FrameMaker template, or edited manually using your favourite XML editor.

##### **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 *[not included in ZIP file]*

### Installing the plugin

The dita2mmpdf plugin may be installed for each DITA-OT version you have installed on the system. Note that the folder where the DITA OT is located may need to have **full write access permissions** set to it.



## DITA-OT plugin

### Installing and integrating with oXygen

**IMPORTANT:** The folder where the DITA OT is located needs to have **full write access permissions set to it**. In particular, if you are choosing to install the plugin in the DITA-OT plugin bundled with oXygen (recommended), and oXygen is installed in a read-only location such as c:\program files, you will need to change the security settings on the DITA-OT2.x folder to give full control access to all users.

The DITA-OT installation bundled with oXygen is located in the `<oXygen install dir> frameworks/dita/DITA-OT2.x/plugins`, for example:

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

In this case, the Miramo DITA-OT plugin will be installed in:

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

Log in as an administrator user, extract the ZIP file into the `<oXygen install dir> frameworks/dita/DITA-OT2.x/plugins` folder, or to your chosen `<dita install dir>` (but note if you choose a different folder you will have to set the **dita.dir** variable in the transformation scenario).

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.

Next, set up the MiramoPDF transformation scenario: Choose options->import transformation scenarios, then navigate to the **MiramoPDF.scenarios** file located in the `<oXygen install dir> frameworks/dita/DITA-OT2.x/plugins/com.miramo.mmpdf/oxygen` subfolder.

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 7 for more information.

### Installing from the setup.exe

Double-click on the `MiramoDITA-OTplugin<vs>setup.exe` and navigate to the DITA-OT install folder, and follow the instructions on-screen. This will automatically set the PATH environment variable to include the `plugins/com.miramo.mmpdf` folder which contains the **dita2mmpdf** script so it can be run from any folder.

### Installing from the ZIP file

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 plugin

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

Set the PATH environment variable to include the `<dita 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](http://www.dita-ot.org) for more information on installing the ZIP file with earlier versions of the DITA-OT.

## 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>/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 ditadir\samples\flowers\flowers.pdf created  
Finished.
```

## Command line options

The `dita2mmpdf` script has the following usage:

```
dita2mmpdf [-dev] [-ditaval file.ditaval] [ditafilename] [template.mfd [pdffilename]]
```

### 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 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:

## DITA-OT plugin

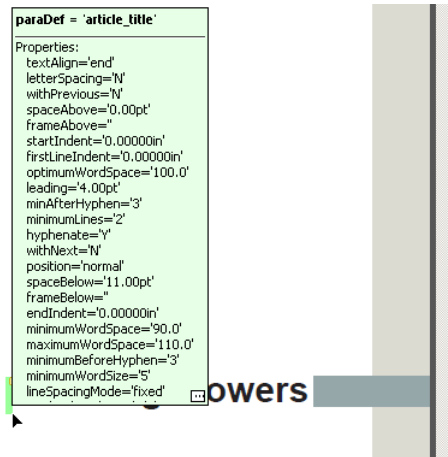
```
dita2mmpdf samples/flowers/flowers.ditamap twocol.mfd ff.pdf
```

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:



The tooltips shown if the **-dev** command line option is supplied to the dita2mmpdf.cmd; by default tooltips are not shown.

### 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 **plugins/com.miramo.com.miramo.mmpdf/mmtemplates** folder.

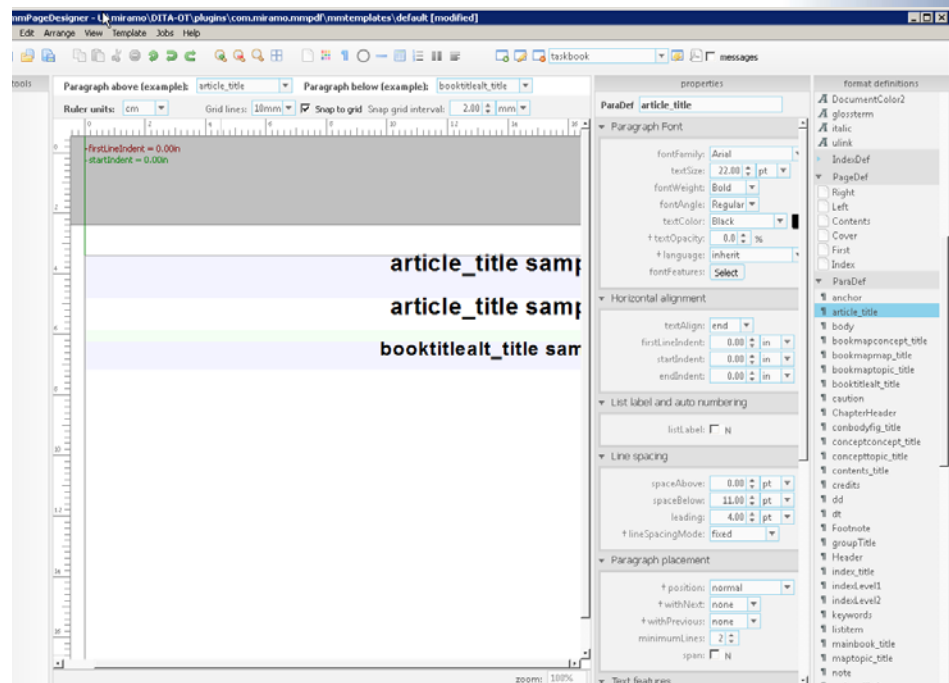
These are created and modified using **MiramoDesigner** which is a GUI page design tool supplied as part of the Miramo distribution version.

### 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.



## DITA-OT plugin



### 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.

### Using the DITA-OT mmpdf output transformation

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

Here is an example of using the 'dita' -f mmpdf command line tool:

```
dita -f mmpdf <path_to_ditafile>
```

Or you can use ant to build the 'dita2mmpdf' target from the samples/flowers folder like this:



## DITA-OT plugin

```
ant -f ../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 <code>\${dita.dir}\plugins\com.miramo.mmpdf\mmtmp\ates</code>
pdf.file	input file with file extension replaced by .pdf	Path to the PDF file to be created
showPropertes	Y	If this property is set to <b>Y</b> , the PDF output file will contain PDF tooltips describing paragraph, font and table formats. Set to <b>N</b> for production.
transtype	mmpdf	<b>mmpdf</b> - do not modify
cwd	folder containing input file	Current working folder for the <b>mmComposer</b> process - sometimes useful for resolving relative URLs
clean.temp	yes	Specifies whether to remove temporary intermediate files
dita.temp.dir	mmtmp	Specifies temporary folder for intermediate files



# DITA-OT plugin

## 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:span	title, p	Spans title across all columns in multi-column document
mmpdf:noTOC	map, bookmap	Suppress TOC generation
mmpdf:noIndex	map, bookmap	Suppress index generation
mmpdf:pageBreak	title, p, note	breaks page before title/p/note paragraph
mmpdf:columnBreak ( <i>alias for break-before</i> )	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, topic/title, topic, chapter, appendix, booktitle	start new section using SectionDef 'name'. SectionDef is defined in the MFD template or using <b>&lt;SectionDef sectionDef="name"../&gt;</b> and starts a new page layout sequence

## 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 [support@datazone.com](mailto:support@datazone.com) with comments, criticisms and questions - your feedback would be much appreciated.