



Pintu Publish Tutorial

[Preparing your XML document]

Contents

1	Introduction.....	3
2	Using the examples provided with Pintu Publish	4
3	Example 1: Catalog 01 , Type " <i>Page</i> "	5
3.1	Step 1: Creating a book file	5
3.2	Step 2: Creating a document file	5
3.3	Step 3: Adding pages to your document.....	6
3.4	Step 4: Adding some content.....	6
3.5	Step 5: Adding tables	8
3.6	Step 6: Adding Images	10
4	Example 2: Catalog 02 , Type " <i>Flow</i> ", table of content, index.	12
4.1	Going with the flow	12
4.2	Cover/Inlay pages.....	14
4.3	Table of contents.....	14
4.4	Index.....	17

1 Introduction

Pintu Publish provides many options to create your desired publication in InDesign. We supply 6 example catalogues with your trial download to demonstrate these options.

This tutorial explains **how to format your XML files** in order to reach the desired output in your publications. This tutorial also refers to specific **InDesign lay-out elements** that should be included in your template in order to create an automated lay-out with Pintu Publish.

There are **two main techniques** to create your publication with Pintu Publish:

- **"page lay-out"**: uses left- and right-master pages to create your publication
- **"flow lay-out"**: creates new pages as more content is found on the page

This tutorial will help you select the technique that best fits your design needs.

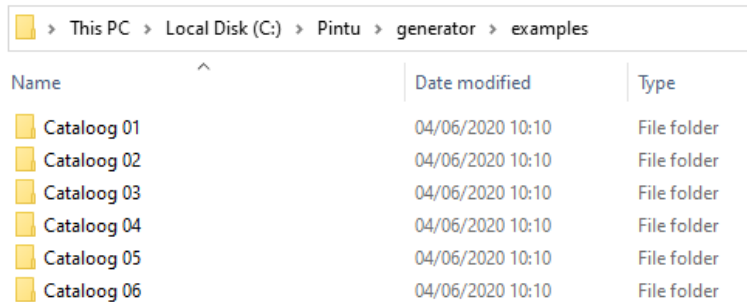
We invite you to read the complete tutorial since the basics are explained early on and we build upon that knowledge to explain the more specific techniques later on.

Screenshots in this tutorial are taken in InDesign 2020 with the workspace set to *"Advanced"* mode.

2 Using the examples provided with Pintu Publish

You can run any of the provided examples using Pintu Publish.

The examples are provided under the folder "examples" in the generator folder.



Name	Date modified	Type
Catalog 01	04/06/2020 10:10	File folder
Catalog 02	04/06/2020 10:10	File folder
Catalog 03	04/06/2020 10:10	File folder
Catalog 04	04/06/2020 10:10	File folder
Catalog 05	04/06/2020 10:10	File folder
Catalog 06	04/06/2020 10:10	File folder

Pintu Publish works through a simple command line. More information on how to request a license file and how to start working through the command prompt is explained in [this video](#).

For a complete overview of the tag references, please [visit our website](#).

This is the command line setup to run the first example:

```
[InstallFolder]\generator\pintu_publish.bat -fc -nr -d -p generator -f  
"[InstallFolder]\generator\examples\Catalog 01\output\Catalog 01.xml"
```

Where:

- [InstallFolder] = The folder where you installed the software
- Option "-fc" = closes the generated files after completion In InDesign
- Option "-nr" = Leave the original XML file as it is after completion
- Option "-d" = Show extra debug information
- Option "-p generator" = The parser needed to create the examples in its containing directory
- Option "-f [PathToXmlFile]" = Use this to refer to the selected example file in the output folder

Each example directory contains all necessary files required to run the example, including assets and template.

The "::::" references in XML examples refer to the root directory path of the .xml file that you want to generate with the parser. This can be changed to your desired directory.

3 Example 1: Catalog 01, Type "Page"

This example shows the use of the pages technique, with left- / right-master pages in InDesign to create your publication.

This example generates pages and places content in each text frame available on the master page lay-out in the template. This *page*-oriented positioning of content is adequate in situations where you know how much content is available and can anticipate the required room needed to actually place it.

But this is not always practical if your data has been created with a variable height in mind. In that case, see [Example 2](#) to become familiar with flow-based lay-out and tags.

3.1 Step 1: Creating a book file

The first step simply sets up the containing "book" or ".indb" file for your catalogue. To accomplish this we just need to set up the XML file first.

1. Create your ".xml" file and add the following lines:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE BOOK SYSTEM "http://dtd.glbserve07/DTD.generator/tags.dtd">
```

If you wish to set up your own doctype reference to the ".dtd" file, you will need to host it somewhere first, but this step is not required for **Pintu Publish** to continue with the generation.

2. Add a reference to your book file by adding the "<book>" tag:

```
<BOOK output="::rootPath::/output/Catalog_01.indb">...</BOOK>
```

The output should refer to the book file that needs to be created.

Make sure the directory referred to exists and the file has the extension ".indb"

3.2 Step 2: Creating a document file

Next, we need to have one or more documents or ".indd" files to actually put the contents in. Because of the way InDesign requires resources, we advise to split your project into multiple documents by chapter or some other boundary that makes sense to your project.

3. Create the "<doc>" tag for each document you wish to create.

```
<DOC output="::rootPath::/output/Catalog_01.indd"
template="::rootPath::/templates/catalog.indt">...</DOC>
```

This requires an output file path to the ".indd" file you wish to create and the existing template ".indt" file. At this point in time the template should just be an empty, newly created, template file.

3.3 Step 3: Adding pages to your document

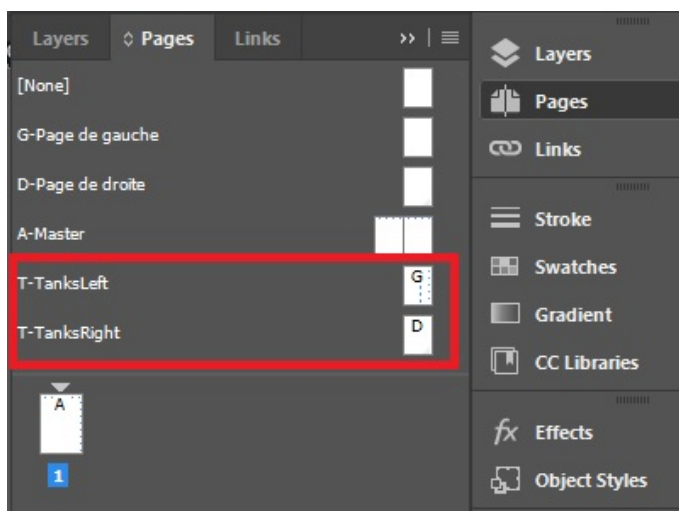
Content can only be positioned on pages, so the next step consists in creating a page.

4. Create the "*<page>*" tag for each page you wish to create:

For the "*page*" type distribution you have to create a page manually. See "*flow*" type in later examples for the difference.

<PAGE masterPage="T-TanksLeft">...</PAGE>

Each page requires a master page to be specified. This can be the default "A-Master" found in a default template file in InDesign. But it is advised to make your own and include it in your InDesign template. In this example we have a left (T-TanksLeft) and right (T-TanksRight) master page. Each has its own design and available content containers.



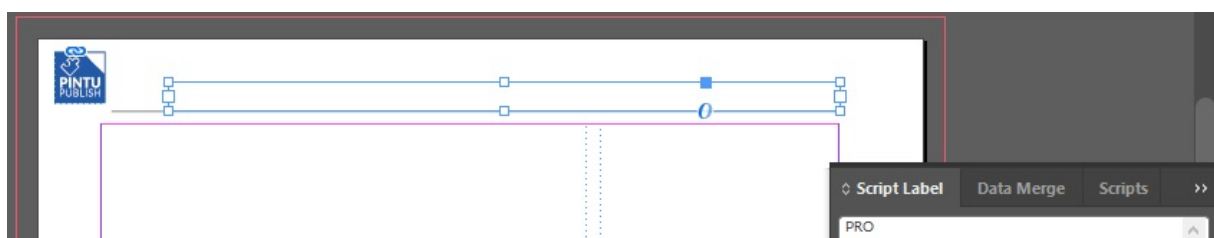
3.4 Step 4: Adding some content

The most basic of content can be added using an available text frame and paragraph in InDesign.

5. Adding a "*<textframe>*" tag to place the content in:

<TEXTFRAME id="PRO">...</TEXTFRAME>

The text frame accepts a single "id" which is a unique identifier set in your InDesign template. The unique identifier actually references a "script label" on the text frame found in your template master page.



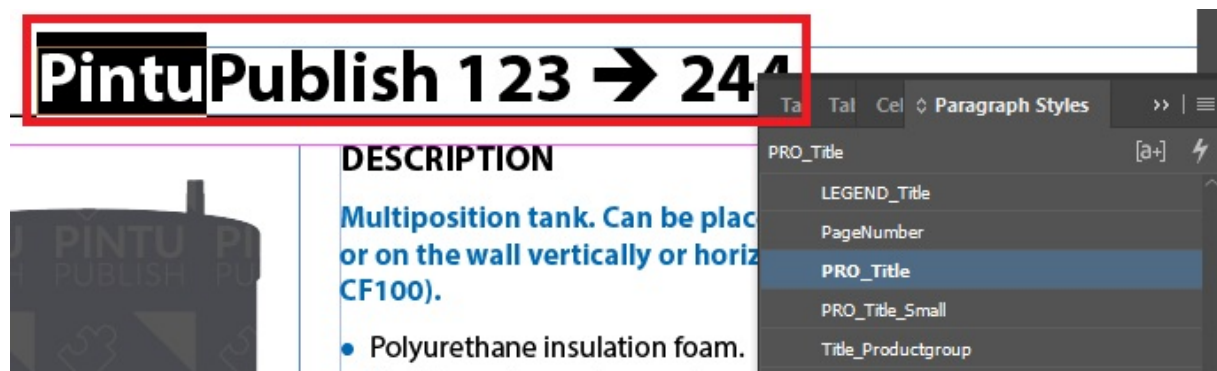
As you can see in this example the “*T-TanksLeft*” contains a text frame with the “*PRO*” script label. This has to be unique for any given master page. Each text frame can be accessed using a different “*<textframe>*” tag.

6. Adding some paragraph content to the selected text frame with “*<p>*” tag:

```
<TEXTFRAME id="PRO">
  <P class="PRO_Title" align="L">PintuPublish 123 <span class="Arrow">è</span>
  244</P>
</TEXTFRAME>
```

The paragraph does not need any attributes, if you only wish to use the default paragraph style set in your InDesign template.

Again, we advise to use your own styles to customize your design: giving each paragraph a purpose and unique style makes sure you can adjust your lay-out later in InDesign with minimal effort.



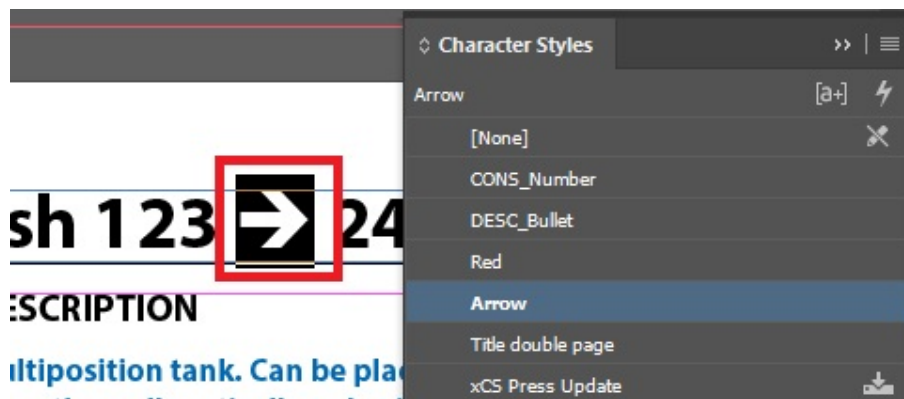
The “*class*” attributes is the name of your paragraph style found in the template. Other attributes can be used to overrule the basic style behaviour. In this case “*align*” to “*L*” means the paragraph is forced to align to the left regardless of the settings found in the style.

Paragraphs are the most common tags used in **Pintu Publish** as everything of content must be placed in a “*story*” in InDesign. This means that nearly everything is contained within the boundaries of a paragraph.

7. Applying the “**” tag:

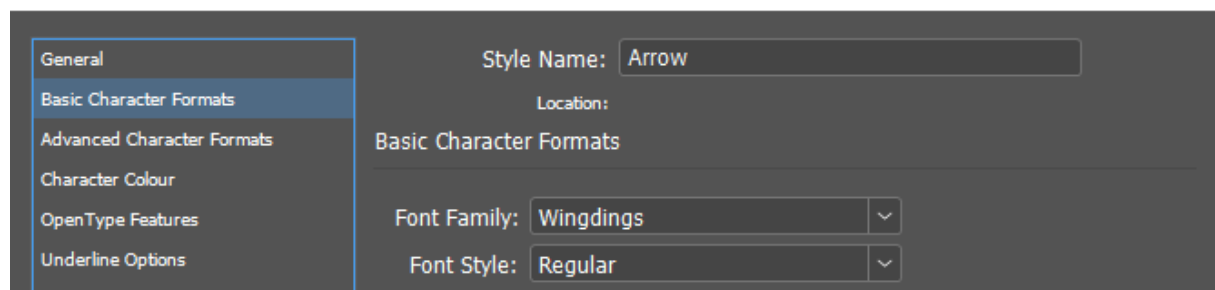
```
<TEXTFRAME id="PRO">
  <P class="PRO_Title" align="L">PintuPublish 123
  <span class="Arrow">è</span>
  244</P>
</TEXTFRAME>
```

The second most common tag found is the “*span*” tag. Which is used to denote the character styles in InDesign. Here we can see a good example of how we can use a character style to apply a different font to achieve a better visual design in text.



The arrow denoting our range of products is simply a character style in our template that has a different font style applied to it. We then wrap the glyph required to create the icon in our font with our style "Arrow".

Character Style Options



3.5 Step 5: Adding tables

Tables are commonly used to display product related data or to create unique lay-outs on a single page. In this example we have a table containing articles and their relevant data.

Reference	Name	Price
0000012	Smart PB 100	€ -,
0000013	Smart PB 130	€ -,
0000014	Smart PB 160	€ -,
0000015	Smart PB 210	€ -,
0000016	Smart PB 240	€ -,

Reference	Description	Price
ACCESSORIES		

- Using the "<table>" tag:

```
<TEXTFRAME id="DESC">
  <P class="XCS_Table" align="L">
    <TABLE width="114mm" strokeWeight="0pt" spaceBefore="10mm">
      ...
    </TABLE>
  </P>
</TEXTFRAME>
```


This tag is the start of your table and contains the settings for the table frame itself. A table width is required in order to calculate the column widths. Other settings you may include are: the stroke surrounding the table and before/after spacing.

9. Using the "<th>" and "<tr>" tags: table header, table row

```
<TEXTFRAME id="DESC">
  <P class="XCS_Table" align="L">
    <TABLE width="114mm" strokeWeight="0pt" spaceBefore="10mm">
      <TH minHeight="5mm" maxHeight="50mm">
        ...
      </TH>
      <TR minHeight="5mm" maxHeight="50mm">
        ...
      </TR>
    </TABLE>
  </P>
</TEXTFRAME>
```

These tags define the row specifications for your header or data rows.

For the headers, these specifications will be repeated on subsequent pages/columns in your text frame (as any other table with headers in InDesign). Options are available to adjust the behaviour including min/max height and keep with next lines.

10. Using the "<td>" tag: table data

```
<TEXTFRAME id="DESC">
  <P class="XCS_Table" align="L">
    <TABLE width="114mm" strokeWeight="0pt" spaceBefore="10mm">
      <TH minHeight="5mm" maxHeight="50mm">
        <TD width="14 mm" strokeWeight="0pt" bStrokeColor="swatch=Black"
bStrokeTint="100" bStrokeWeight="0.75pt" lInset="0mm">
          <P class="DESC_TableHeader" align="L">Reference</P>
        </TD>
        <TD strokeWeight="0pt" bStrokeColor="swatch=Black" bStrokeTint="100"
bStrokeWeight="0.75pt" lInset="0mm">
          <P class="DESC_TableHeader" align="L">Name</P>
        </TD>
        <TD width="14 mm" strokeWeight="0pt" bStrokeColor="swatch=Black"
bStrokeTint="100" bStrokeWeight="0.75pt" rInset="0mm">
          <P class="DESC_TableHeaderRight" align="L">Price</P>
        </TD>
      </TH>
      <TR minHeight="5mm" maxHeight="50mm">
        <TD strokeWeight="0pt" bStrokeColor="swatch=Black" bStrokeTint="100"
bStrokeWeight="0.25pt" lInset="0mm">
          <P class="DESC_TableValueLeft" align="L">0000001</P>
```

```

</TD>
<TD strokeWeight="0pt" bStrokeColor="swatch=Black" bStrokeTint="100"
bStrokeWeight="0.25pt" lInset="0mm">
  <P class="DESC_TableValueLeft" align="L">PintuPublish 123</P>
</TD>
<TD strokeWeight="0pt" bStrokeColor="swatch=Black" bStrokeTint="100"
bStrokeWeight="0.25pt" rInset="0mm">
  <P class="DESC_TableValuePrice" align="L">€ -,</P>
</TD>
</TR>
...
</TABLE>
</P>
</TEXTFRAME>

```

The table data contains your actual content and works just like a text frame when it comes to adding content. It accepts paragraphs and character styles. It can also contain other tables to create complex lay-outs.

- Settings include the width, strokes, insets and other typical design elements in a cell.
- A width is not required, but, if left unset, all columns will have variable widths.
- When the table must accommodate long text, it is recommended to make enough room for your content in order to avoid overflow.

As stated previously InDesign puts everything in its own story. This is done using a paragraph. This means that a table must be assigned a paragraph style and the content inside the table must also be assigned a specific paragraph style.

3.6 Step 6: Adding Images

The two main ways of adding images is

- by filling in an existing image frame (similarly to a text frame),
- by adding the images inline. This technique was used in this example.

11. Using the "" tag:

```

<p class="XCS_ImageInline" align="L">
  
</p>

```

It is as simple as using a paragraph or a character style.

- The image tag must be placed inside the story with the usual technique using paragraphs.

- The image settings require a "src" that points to an image file available to the installed InDesign software, on a network drive for instance.
Caution: the type of slashes in your file path does matter.
- Other settings include the maximum width and height, strokes, the type of image which might consist of inline, anchored or baseline. Examples of the anchored version are applied in other example files provided with the Pintu Publish generator.

As with all examples shown here, anything inside a paragraph can easily be adjusted visually by making sure they get their own unique style. That way small changes like inset, font adjustments, colouring, shifting of position, etc. can be done by just adjusting the paragraph style in your template.

All other content inside this example has been created using these 6 easy steps. This is a strong starting point for any project. Things like datasheets, catalogue pages or price lists can all be created based on these tags.

4 Example 2: Catalog 02, Type "Flow", table of content, index.


This example shows the use of a "flow" generation that creates new pages as more content is found on the page. This example also includes some options to create your own table of contents and index.

4.1 Going with the flow

In this example we have product groups that could contain any number of products. We wish to place these on a baseline grid that forces specific content to start from these position if possible. We also do not know if the product table will overflow over to a new page or not, so we have to incorporate a "flow" to make sure all content is placed in the layout, even if it does not fit the page.


PB DAKVENSTERS

UITZET-TUIMELDAKVENSTER HOUT




V40P

PB-V Q3 prepared

hout met witte acryl-lak
2-lagig glas Q3  Ug = 1,0

ArtikelNr	MC	Afm. [cm]	Eff. Glas. [m2]	L	B	G	Vko	# Verp.	# Pallet	Prijs excl.	Prijs incl.	Levering
00026	02	55x98		0,55	0,98	32,00	ST	1,00	7,00	€ 1369,57	€ 1445,97	
00027	05	78x98		0,78	0,98	37,00	ST	1,00	7,00	€ 1413,18	€ 1499,95	
00028	06	78x118		0,78	1,18	42,00	ST	1,00	7,00	€ 1449,10	€ 1542,2	
00029	07	78x140		0,78	1,40	49,00	ST	1,00	7,00	€ 1479,06	€ 1579,18	
00030	08	94x118		0,94	1,18	49,00	ST	1,00	7,00	€ 1502,41	€ 1607,92	
00031	09	94x140		0,94	1,40	56,00	ST	1,00	7,00	€ 1540,56	€ 1654,08	
00032	10	114x118		1,14	1,18	59,00	ST	1,00	7,00	€ 1547,87	€ 1662,92	
00033	11	114x140		1,14	1,40	65,00	ST	1,00	7,00	€ 1651,80	€ 1789,68	
00034	12	134x98		1,34	0,98	57,00	ST	1,00	7,00	€ 1559,46	€ 1676,95	
00035	17	134x140		1,34	1,40	73,00	ST	1,00	7,00	€ 1649,47	€ 1785,86	



baseline

UITZET


FPU-V

hout met
2-lagig gl

ArtikelNr
00040
00041
00042
00043
00044
00045
00046
00047
00048
00049


PB DAKVENSTERS

UITZET-TUIMELDAKVENSTER HOUT



V40P

PB-V B52 prepared

hout met witte acryl-lak
2-lagig glas B52  Ug = 1,0

ArtikelNr	MC	Afm. [cm]	Eff. Glas. [m2]	L	B	G	Vko	# Verp.	# Pallet	Prijs excl.	Prijs incl.	Levering
00036	06	78x118		0,78	1,18	42,00	ST	1,00	7,00	€ 1515,71	€ 1624,01	
00037	07	78x140		0,78	1,40	50,00	ST	1,00	7,00	€ 1559,08	€ 1675,28	
00038	10	114x118		1,14	1,18	50,00	ST	1,00	7,00	€ 1613,33	€ 1742,13	

UITZET

FPU-V

hout met
2-lagig gl

ArtikelNr
00050
00051
00052
00053
00054
00055
00056

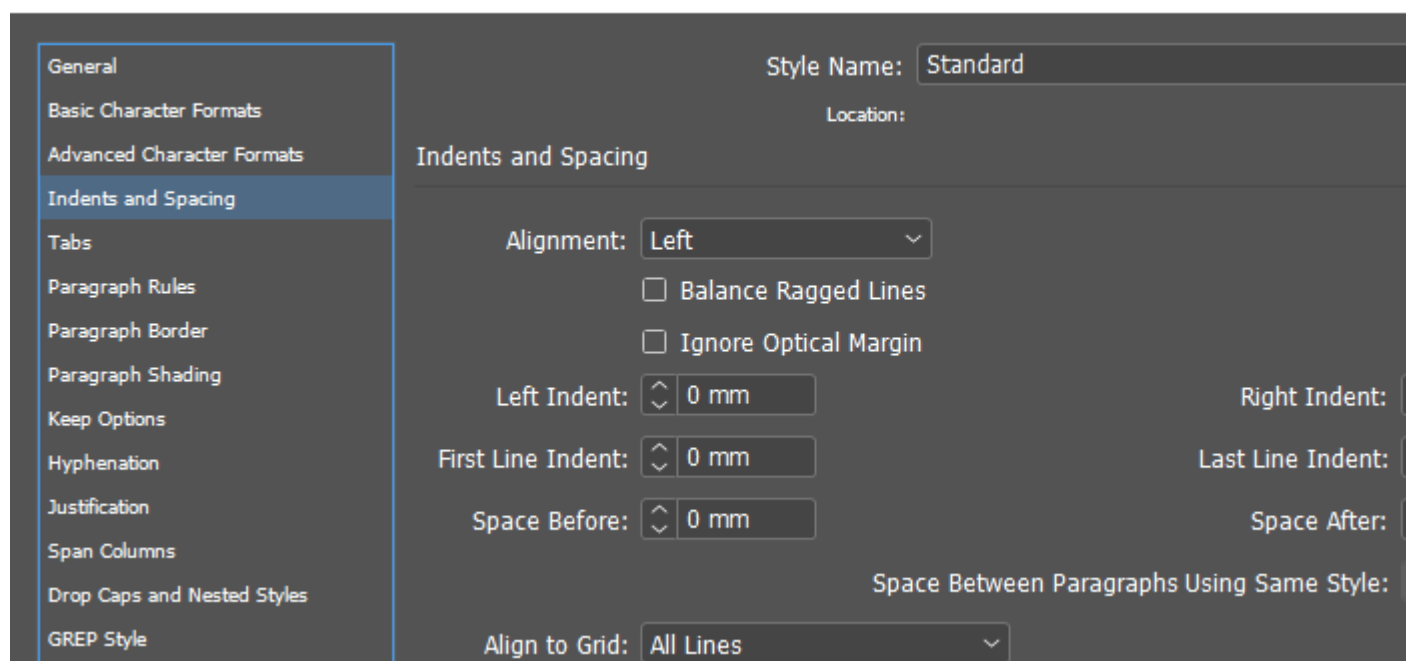
- The text frame id "flow":

```
<PAGE masterPage="A-Master">
  <TEXTFRAME id="flow">...</TEXTFRAME>
</PAGE>
```

Unlike previous example using *"page"*, we now use a very specific script label called *"flow"*. To start using this, make sure your master page (*in this case "A-Master"*) has only one text frame with this script label name. If you are working with a spread (*like this example*), both the left and right page may contain a text frame with this id. Each page is considered a separate entity in InDesign.

Besides this minor change, there is nothing more to do than place your content as seen in the previous example and apply the correct paragraph styles found in your template. In this example the paragraph style *"Standard"* is aligned to the grid (*Indents and spacing > align to grid*).

Paragraph Style Options



When the content you place goes beyond the height of the text frame with id *"flow"*, the software will create a new page with the exact same master page as the previous one. Then apply the rest of the content to the text frame found with id *"flow"*. Thus creating a variable amount of pages to place all content.

This technique has its drawbacks:

- If the content you wish to place is larger than the height available but is prevented from breaking into smaller parts, the software will create a large amount of overflow pages before eventually giving up.

Examples:

- o a long table with all rows set to *"keepWithNext"*;
- o a single row with a height set to a maximum that is higher than the text frame;
- o large images without a proper maximum height set.
- The lay-out remains within the boundaries of a single master page. In order to offer more flexibility, you can use a spread to assign different design elements on the left and right page, or use the away/towards spine placement options in a paragraph style for example.

In the "flow" lay-out, table will overflow and break if possible, making use of the table header to repeat your header content on the next page.

You can force a page break by simply having a paragraph style start on a new page or starting a new "<page>" tag.

```
<PAGE masterPage="A-Master">
  < TEXTFRAME id="flow">...</TEXTFRAME>
</PAGE>
<PAGE masterPage="A-Master">
  < TEXTFRAME id="flow">...</TEXTFRAME>
</PAGE>
```

4.2 Cover/Inlay pages

A cover or inlay page might be used to spread out the content before and after it, but have no real content of their own. You can always just add a page without content by setting the master page to the cover or inlay you wish and just close the tag again.

```
<PAGE masterPage="C-Info" />
```

This creates a simple page at the desired position with no dynamic content.

4.3 Table of contents

To start creating a table of contents, you first need to lay-out your content. Once your content has been laid out in a document, and that document is positioned inside a book, we can finally start to add the table of contents.

```
<BOOK output="::rootPath::/output/Catalog_02_Book.indb" exportPdf="0"
  pdfQuality="high"
  exportJpg="0" package="0" packageImages="0" >
  <DOC output="::rootPath::/output/Catalog_02_TOC.indd"
    template="::rootPath::/templates/catalog.indt">
    <TOC masterPage="B-Toc"
fromBook="::rootPath::/output/Catalog_02_Book.indb"
    tocName="TOC">
    <TOCDEF style="TocLevel1" appliedStyle="TOC1" separator="^t"
      pageNumberPosition="afterEntry" />
    <TOCDEF style="TocLevel2" appliedStyle="TOC2"
      separator="^t" pageNumberPosition="afterEntry" />
    <TOCDEF style="TocLevel3" appliedStyle="TOC3" separator="^t"
      pageNumberPosition="afterEntry" />
    </TOC>
    ...
  </DOC>
</BOOK>
```

2. The "<toc>" tag: Table of Contents

```
<TOC masterPage="B-Toc" fromBook="::rootPath::/output/Catalog_02_Book.indb">
...
</TOC>
```

Like the "<page>" tag, the table of contents (<toc>) tag *requires a template master page to define its lay-out*. It is also of unknown height, which means this automatically functions as a "flow" type page. This means you will need to add a text frame with id "flow" to your master page for it to work. (See "B-Toc" master page in example template)

The other required part is to connect your table of contents to your book with the "fromBook" attribute. This should be the full path to your ".indb" file.

3. The "<tocdef>" tag: Table of Contents levels

This tag adds the levels to your table of contents. Some requirements must be met inside your content.

```
<TOC masterPage="B-Toc" fromBook="::rootPath::/output/Catalog_02_Book.indb">
  <TOCDEF style="TocLevel1" appliedStyle="TOC1" separator="^t"
pageNumberPosition="afterEntry" />
  <TOCDEF style="TocLevel2" appliedStyle="TOC2" separator="^t"
pageNumberPosition="afterEntry" />
  <TOCDEF style="TocLevel3" appliedStyle="TOC3" separator="^t"
pageNumberPosition="afterEntry" />
</TOC>
```

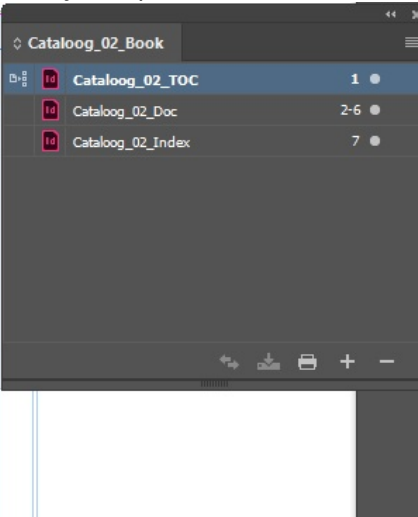
You will need to have multiple paragraph styles in use with unique names. The table of contents automatically has to detect these paragraphs inside of your book and subsequent documents.

```
<P class="TocLevel1">FPP</P>
<P class="TocLevel2">PB-X</P>
<P class="TocLevel3">PB-X Q3 prepared</P>
```

In the table of contents definition, you select the paragraph to detect with the "style" attribute. With the "appliedStyle" you tell it to apply the given paragraph style to the newly

created table of content line. Each "*<tocdef>*" represents a new level in the final generation.

FPP	3
PB-X	3
PB-X Q3 prepared	3
PB-X B52 prepared	3
PB-X Q5 prepared	3
FPW	4
PB-V	4
PB-V Q3 prepared	4
PB-V B52 prepared	4
PB-V Q5 prepared	4
FPU	5
FPU-V	5
FPU-V Q3 prepared	5
FPU-V B52 prepared	5
FPU-V Q5 prepared	5



4.4 Index

To start creating an index, you first need to lay out your contents. Once your content is available in a document, and that document is positioned inside a book, we can start to add the index.

```
<BOOK output="::rootPath::/output/Catalog_02_Book.indb">
  <DOC output="::rootPath::/output/Catalog_02_Index.indd"
  template="::rootPath::/templates/catalog.indt">
    <PAGE masterPage="D-Index">
      <INDEX title="Index" titleStyle="Index Title" level1Style="Index Level 1"
      level2Style="Index Level 2" level3Style="Index Level 3" pageNumberStyle="Text"
      followingTopicSeparator="^t" betweenPageNumbersSeparator=","
      pageRangeSeparator="" betweenEntriesSeparator=""
      beforeCrossReferenceSeparator="" includeSectionHeadings="0" />
    </PAGE>
  </DOC>
</BOOK>
```

Unlike the Table of Contents, the index is placed inside of a "*<page>*" tag.

The attributes "title" and "titleStyle" apply an optional title at the top before the index starts and a paragraph style.

The "level1Style" up to "level5Style" applies a paragraph style to the different levels, see markers.

Other settings include separators and separator characters.

4. The "*<indexmarker>*" tag:

This adds the levels to the index and is required to generate the index from your content.

```
<P class="xCS_Index">
  <INDEXMARKER level1="FPP"/>
  <INDEXMARKER level1="FPP" level2="PB-X"/>
  <INDEXMARKER level1="FPP" level2="PB-X" level3="PB-X Q3 prepared"/>
</P>
```

Each index marker has its own content to display in the index.

For each level of content you need to create an index marker in your document.

In this example we have "*PB-X Q3 prepared*" as a level three content that belongs to "*PB-X*" on level two, which in turn belongs to "*FPP*" on level one.

Each level your index marker is attached to, must previously exist in your content.

In this example index marker for "*FPP*" on level one has to already exist before you place the level two marker for "*PB-X*".

If the content is set up correctly, the index will "flow" onto as many pages as it needs to fit.

FPP.....	3
PB-X.....	3
PB-X B52 prepared	3
PB-X Q3 prepared.....	3
PB-X Q5 prepared.....	3
FPU.....	5
FPU-V.....	5
FPU-V B52 prepared	5
FPU-V Q3 prepared.....	5
FPU-V Q5 prepared.....	5
FPW.....	4
PB-V.....	4
PB-V B52 prepared	4
PB-V Q3 prepared.....	4
PB-V Q5 prepared.....	4