inēdit

Integration Resources

Index

1. neoRipEngine

Granting Executable Permissions for RipEngine CGI

The commands (granting executable permissions using chmod +x) are typically done in the following situations:

1. **Permission Issues When Running Executable Files** : If you encounter an error like Permission denied when trying to execute the files (e.g., neoKeyManagerCGI, neoRipEngineCGI, or neoVirtualVisionCGI), it means the files lack the required executable permissions. Running chmod +x resolves this issue by granting execution rights.
2. **Setting Up or Updating neoCatalog or Related Software** : When installing, updating, or configuring neoCatalog or its components, you may need to grant executable permissions to CGI files or other binaries to ensure the application functions correctly.
3. **Migrating neoCatalog to a New System** : If you're transferring the application or its components to a new server or system, the file permissions may not be preserved during the transfer. Reapplying with the correct permissions ensures the application runs smoothly.
4. **Post-Installation Configuration** : Some installation processes don’t automatically set the necessary permissions for certain files. Running these commands manually ensures that the application components can execute properly.
5. **Debugging Issues with neoCatalog Services** : If specific features or services within neoCatalog (like key management, RIP engine processing, or virtual vision) are not working, missing executable permissions might be the cause.

When NOT to Use These Commands:

- If the application is already functioning correctly, there’s no need to modify permissions.
- Avoid granting permissions unnecessarily, as it could lead to security risks if the file is misused.

Step-by-Step

1. Open the terminal/command line on your Mac.

2. Navigate to the directory by typing the following command:

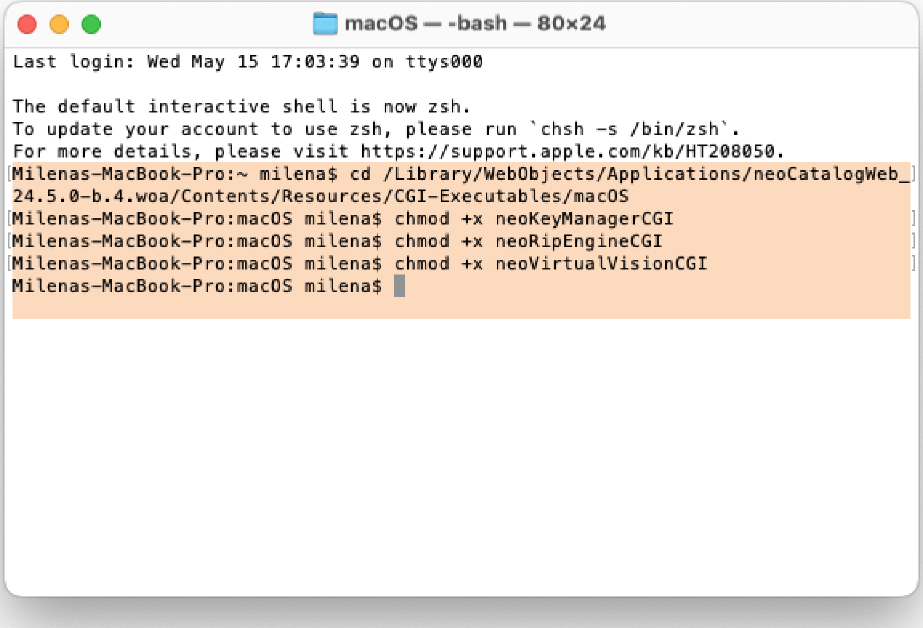
```
cd /Library/WebObjects/Applications/neoCatalogWeb_24.5.0-b.4.woa/Contents/Resources/CGI-
Executables/macOS
```

3. Grant executable permissions to the necessary files by typing the following commands:

```
chmod +x neoKeyManagerCGI
chmod +x neoRipEngineCGI
chmod +x neoVirtualVisionCGI
```

4. Verify permissions (optional) if needed by listing the files with details:

```
ls -l
```



```
Last login: Wed May 15 17:03:39 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Milenas-MacBook-Pro:~ milena$ cd /Library/WebObjects/Applications/neoCatalogWeb_
24.5.0-b.4.woa/Contents/Resources/CGI-Executables/macOS
Milenas-MacBook-Pro:macOS milena$ chmod +x neoKeyManagerCGI
Milenas-MacBook-Pro:macOS milena$ chmod +x neoRipEngineCGI
Milenas-MacBook-Pro:macOS milena$ chmod +x neoVirtualVisionCGI
Milenas-MacBook-Pro:macOS milena$ 
```

neoRipEngine Documentation

The attached document and examples describe the basic usage of the RipEngine library, the job ticket structure, and the interface used for RIP communication.

To execute XML jobs and explore practical examples, follow these steps:

1. Open the terminal, Command Prompt, or PowerShell according to your operating system.

2. Take the execution file neoRipEngineCGI and drop it into that terminal.

3. In the terminal, navigate to the location where you saved the XML job file and select it and then press Enter These files are provided as examples of the documentation requirements. Don't forget to use proper spacing after the command to execute, the path itself is precise and doesn't include any spaces or extra characters as padding.

4. After the XML job execution is complete, review the output or results. This may involve inspecting generated files, examining transformed data, or validating the output against predefined criteria.

5. Analyze the executed XML job and its outcomes to gain insights into its functionality and behavior.

6. Remember to consult the relevant documentation for more detailed instructions and advanced features.

> NOTE: Exercise caution when executing XML jobs, especially if they involve critical or sensitive data. Always ensure you have backups of your files and follow proper data handling practices.

Attachments:

4.23.0 neoRipEngine.pdf
Objects Samples.zip

2. XML Layouts Guide

Cannot generate print with layout that uses QR code

Problem

The printout cannot be generated because of a neoRipEngine error:

```
Error 120] VIPS error: neoQR result error: -1 VipsFormat: file
"/Users/milena/Documents/neoCatalogs/INEDIT-QA/QuickPrint/Layouts/SohoGothicPro-Light.otf" not a
known format vips_image_get: field "icc-profile-data" not found [Error 109] Refactor operation
failed: Image error [Error 111] ID not valid: 4 |
```

Reason

The TJB where the QR code is defined as no output resolution dpi defined.

```
<Output>
  <Space>LAB</Space>
</Output>
```

Solution

Define the output resolution in the TJB output parameters. Recommended is 144 dpi.

```
<Output>
  <ResolutionX>144</ResolutionX>
  <ResolutionY>144</ResolutionY>
  <ResolutionUnits>dpi</ResolutionUnits>
  <Space>LAB</Space>
</Output>
```

Result

```
<?xml version="1.0" encoding="UTF-8"?>
<Job>
<Sources>
    <Source Id="0" URL="${SourceImage}"/>
    <Source Id="1" URL="${SourceXCM}"/>
    <Source Id="2" URL="SohoGothicPro-Medium.otf"/>
    <Source Id="3" URL="SohoGothicPro-Regular.otf"/>
    <Source Id="4" URL="SohoGothicPro-Light.otf"/>
    <Source Id="5" URL="Logo.tif"/>
    <Source Id="6" URL="X_block.tif"/>


  </Sources>
<Layout>
    <Page Id="0" width="${__auto_parentWidth__}">
    <Objects>

      <!--Logo-->
      <Object Id="0" SourceID="5" left="0.2 cm" top="0.2 cm" height="1cm" MaintainAspectRatio="true"
>
      </Object>


      <!--QR-->
      <Object Id="0" width="1.7cm" height="1.7cm" left="0.3cm" top="1.1cm">
      <Transformations>
        <Code value="${SourceID@1.Colorations.Name} ${SourceID@1.Colorations.Coloration@0.Name}"
type="QR"/>
      </Transformations>
      </Object>

      <Object Id="1" Top="2.8cm" width="1.5 cm" left="0.4 cm">
        <Transformations>
          <Text alignment="left" fontid="4" fontsize="9" maxwidth="1cm" value="Printinfo"/>
        </Transformations>
        </Object>


      <!--Created-->
        <Object Id="13" Left="3 cm" Top="0.2 cm">
          <Transformations>
            <Text alignment="left" fontid="3" fontsize="8" maxwidth="10 cm" value="Creator:
${UserDescription}(${UserID}), ${UserMail}"/>
          </Transformations>
        </Object>

        <Object Id="12" Left="14 cm" Top="0.2 cm">
          <Transformations>
            <Text alignment="left" fontid="3" fontsize="8" maxwidth="10 cm" value="Date:
${System.Date.formater#%d/%m/%Y} - ${System.Time.formater#%H:%M:%S} "/>
          </Transformations>
        </Object>



      <!--Design Name-->
        <Object Id="1" Left="3 cm" Top="0.6 cm">
          <Transformations>
            <Text alignment="left" fontid="2" fontsize="8" maxwidth="10 cm" value="Design:
${SourceID@1.Colorations.Name}"/>
          </Transformations>
        </Object>


        <Object Id="3" Left="3 cm" Top="1 cm">
          <Transformations>
            <Text alignment="left" fontid="2" fontsize="8" maxwidth="10 cm" value="Colorway:
${SourceID@1.Colorations.Coloration@0.IData.__USR_CLR_CW_colorwayName}"/>
          </Transformations>
        </Object>
      <!--Requested-->
        <Object Id="5" Left="14 cm" Top="0.6 cm">
          <Transformations>
            <Text alignment="left" fontid="3" fontsize="8" maxwidth="10 cm" value="Request Code:
${RequestCode}"/>
          </Transformations>
        </Object>


        <Object Id="7" Left="14 cm" Top="1 cm">
          <Transformations>
            <Text alignment="left" fontid="3" fontsize="8" maxwidth="10 cm" value="Customer:
${CustomerDescription}"/>
          </Transformations>
        </Object>



      <!--Comments-->
        <Object Id="9" Left="3 cm" Top="1.6 cm">
          <Transformations>
            <Text alignment="left" fontid="3" fontsize="8" maxwidth="18 cm" value="Comments:
${Comments}"/>
          </Transformations>
        </Object>


    <!-- Color samples -->
        <Repetition count="${sourceid@1.Colorations.Coloration@0.ColorationItem.count}"
maxSize="${__auto_parentWidth__}" direction="H" height="1.4 cm" item="aColorationItem" left="3 cm"
list="sourceid@1.Colorations.Coloration@0.ColorationItem" top="2.4 cm" width="2.2 cm">

                <Conditional condition="${${aColorationItem}.channelNo}" equals="-2">
        <Conditional condition="${${aColorationItem}.method}" equals="Disable" negate="y">

          <Object Id="8" Left="0 cm" Top="0 cm">
              <Transformations>
                <Text alignment="left" fontid="3" fontsize="7" maxwidth="0.2 cm"
value="${${aColorationItem}.ChannelIndex}"/>
              </Transformations>
            </Object>

          <Object Id="9" left="0.4 cm" top="0.8 cm">
                        <Transformations>
                            <Text alignment="left" fontID="4" fontsize="7" maxwidth="1.5 cm"
value="${${aColorationItem}.IData.name}"/>
                        </Transformations>
                    </Object>

                     <Object Id="10" left="0.4 cm" top="0 cm">
                        <Transformations>
                            <Gradient Space="${${aColorationItem}.filtergradient.space}"
UseProfile="Y" Alias="${sourceid@1.Colorations.Coloration@0.ProfileInfo.Alias}" height="0.8 cm"
percentages="100" percentbase="100" width="1.5 cm">${${aColorationItem}.filtergradient.value}
</Gradient>
                        </Transformations>
                    </Object>
          </Conditional>
            </Conditional>


        <Conditional condition="${${aColorationItem}.channelNo}" equals="-2" negate="y">
          <Object Id="8" Left="0 cm" Top="0 cm">
              <Transformations>
                <Text alignment="left" fontid="3" fontsize="7" maxwidth="0.2 cm"
value="${${aColorationItem}.ChannelIndex}"/>
              </Transformations>
            </Object>



        <Conditional condition="${${aColorationItem}.method}" equals="Disable" negate="y">
          <Object Id="9" left="0.4 cm" top="0.8 cm">
                        <Transformations>
                            <Text alignment="left" fontID="4" fontsize="7" maxwidth="1.5 cm"
value="${${aColorationItem}.IData.name}"/>
                        </Transformations>
                    </Object>
                    <Object Id="10" left="0.4 cm" top="0 cm">
                       <Transformations>
                           <Gradient Space="${${aColorationItem}.filtergradient.space}"
UseProfile="Y" Alias="${sourceid@1.Colorations.Coloration@0.ProfileInfo.Alias}" height="0.8 cm"
percentages="100" percentbase="100" width="1.5 cm">${${aColorationItem}.filtergradient.value}
</Gradient>
                       </Transformations>
                   </Object>

          </Conditional>


          <Conditional condition="${${aColorationItem}.method}" equals="Disable" negate="n">
            <Object Id = "11" SourceID = "6" left="0.4 cm" top="0 cm"  MaintainAspectRatio="true">
                  </Object>
          </Conditional>

            </Conditional>

            </Repetition>


            </Objects>
  </Page>

  </Layout>
<Output>
  <ResolutionX>144</ResolutionX>
  <ResolutionY>144</ResolutionY>
  <ResolutionUnits>dpi</ResolutionUnits>
  <Space>LAB</Space>
</Output>
</Job>
```

How adjust the direction when dealing with design rotations with half-drop designs

Problem

While the full-drop (1/1) designs appear to be fine when rotating objects, the half-drop (1/2) design isn't rendering correctly. Should additional options or settings be considered specifically for half-drop designs to address this issue?

Solution

Unfortunately, there isn't a straightforward parameter to flip the image drop when rotating. However, we've come up with a workaround that may help you achieve the desired output, even though it involves some intricacies.

1. **Rotate the Image**: Before using the image, apply the necessary rotation using a tool like CGI or Photoshop This step ensures that you won't need to rely on the rotation parameter rotation="90" within the object.
2. **Update the Drop Direction**: Regardless of the rotation, you will need to adjust the drop direction in your job. Change the direction parameter from 0/V to 1/H in the Object as follows:

```
<Page Id="1">
<Objects>
<Object Id="1" SourceID="0" MaintainAspectRatio="true" autoResizeMask="Width|Height"
autoResizePropHeight="1/1" insideHeight="1 rep" insideWidth="1 rep" repeat="rapport" rotation="0">
<Transformations>
<Coloration coloID="7" idataId="0" index="0" specialcolor="0"/> </Transformations>
<RapportInfo direction="1" fraction_high="1" fraction_low="2"/> </Object>
</Objects>
<RapportInfo direction="1" fraction_high="1" fraction_low="2" height="130.00 cm" width="158.00 cm"/>
</Page>
```

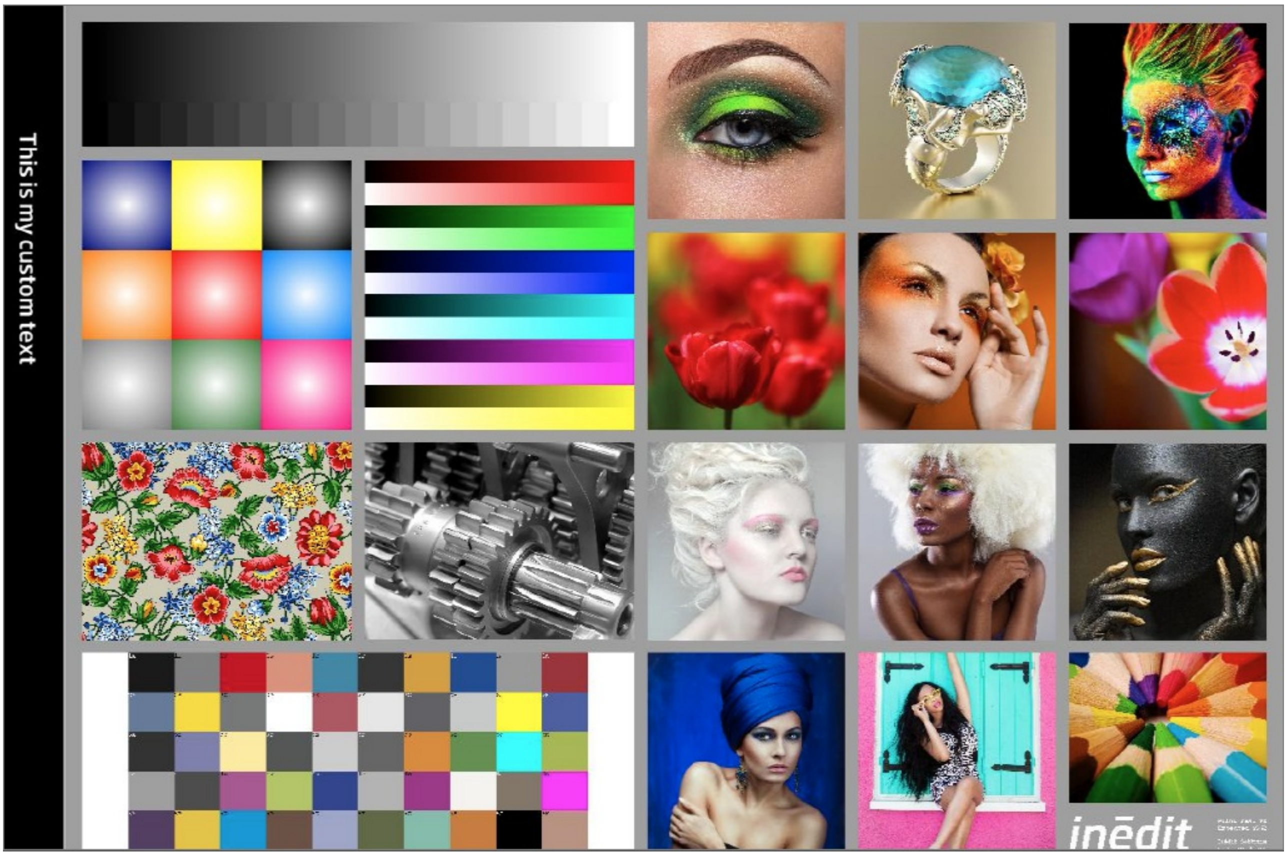
How to add background color in the XML layout

You can customize your Layout by adding a background color using a new object in the layout XML structure. This is especially useful for visual distinction and label clarity.

Example:

```
<Object Id="2" left="0 cm" top="0 cm">
  <Transformations>
    <Gradient Space="RGB" UseProfile="N" height="3cm" width="${RepeatHeight}">0,0,0</Gradient>
    <Rotate degrees="90"/>
  </Transformations>
</Object>
```

Result for band print:



Attachments:

Layout Background Color.zip

How to build a layout

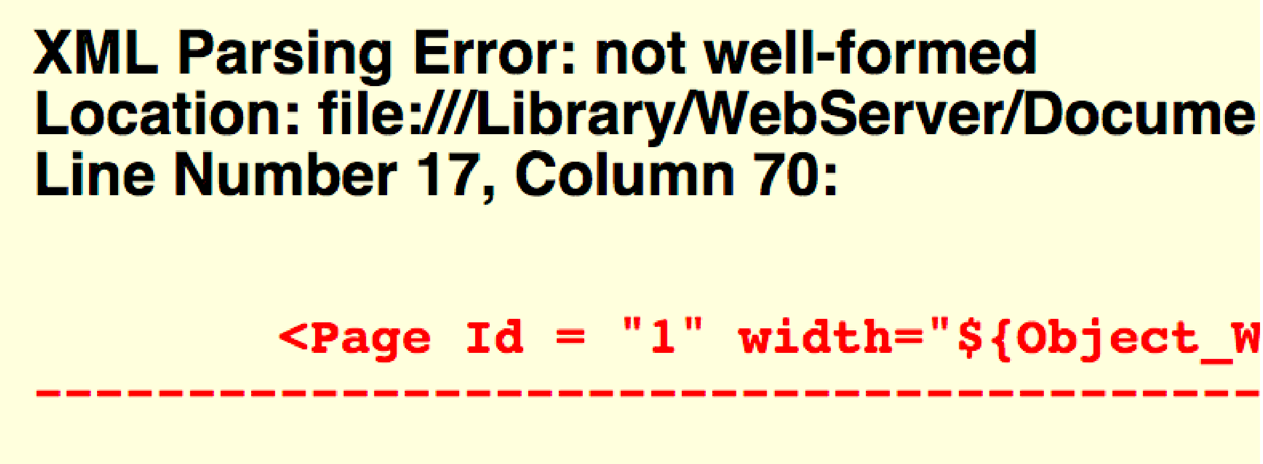
TABLE OF CONTENTS

Requirements

- Use a multichannel design with a colorway that contains color names.
- Use logo with TIFF or PSD extensions.

Tips before and while building layouts:

- Use XML editors like TextWrangler.
- All parameters and values must be inside quotation marks " ".
- Refer to the neoRipEngine attributes and parameters to build the layout in this document. It gives you advance
- Products support automatic variables which are not equal.
- Start from the top of the script and build up the object logically.
- Use comments `<!-- Comments -->` to organize and identify the objects easier.
- Do not mix different layouts samples to not make mistakes in copy/paste issues.
- Use Firefox Browser to check the format accuracy of the XML file. Drag and drop the XML file into the openec

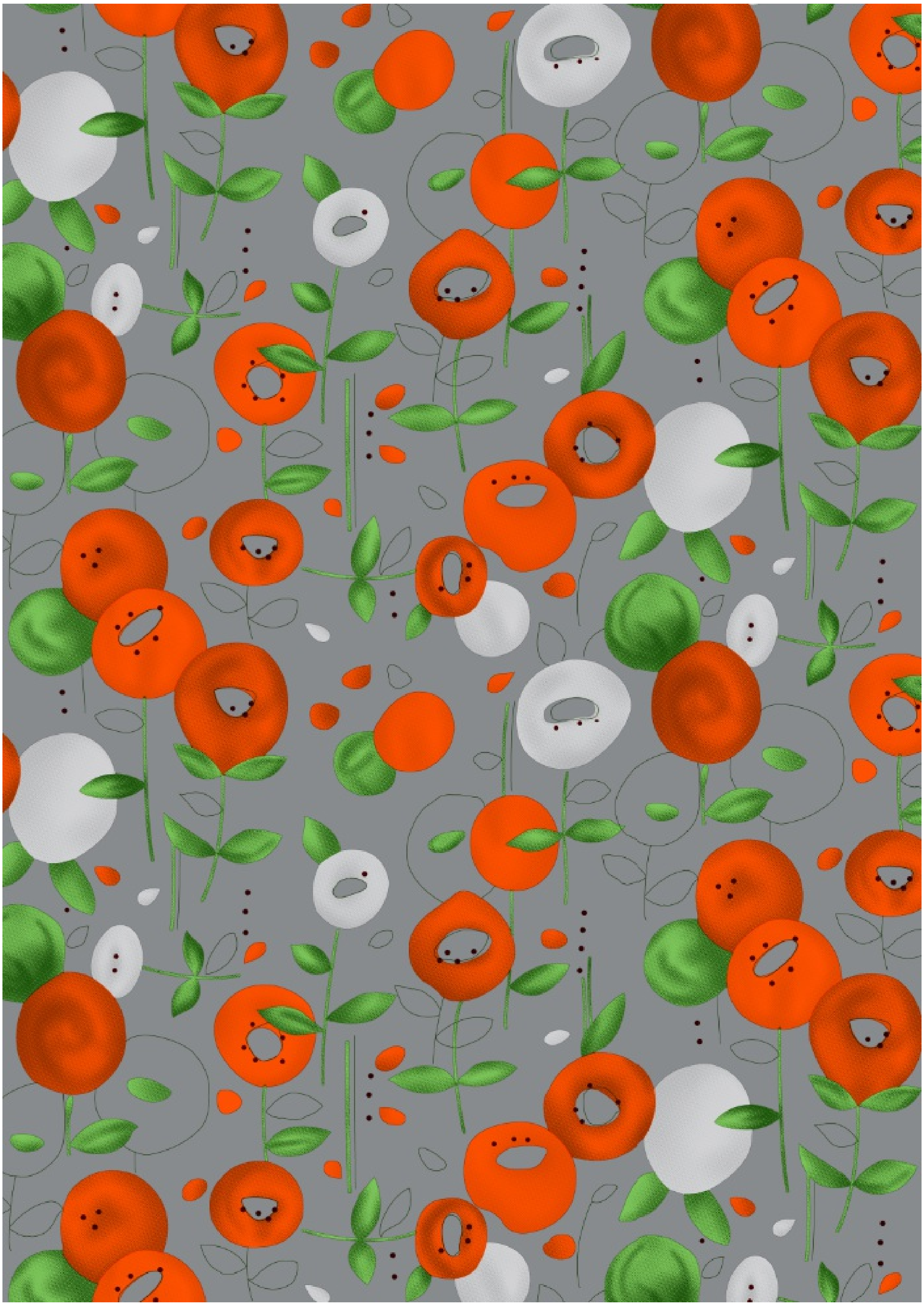


```
XML Parsing Error: not well-formed
Location: file:///Library/WebServer/Docume
Line Number 17, Column 70:

        <Page Id = "1" width="${Object_W
-----------------------------------------
```

Step-by-step

- **Target 1: XML with real-size image, Level Basic**
- **Target 2: TJB with color patches and document info, position top-left in XML Level Basic**

Target 1: XML with real-size image, Level Basic



1. Create the *Job* element: `<Job></Job>`

```
< Job >
</ Job >
```

2. Inside the Job element create the *Sources* element `<Sources></Sources>` and inside the roots of your source images `<Source Id/>`

```
< Job >
< Sources >
< Source Id = "0" URL = "${SourceImage@0}" />
< Source Id = "1" URL = "${SourceXCM@0}" />
</ Sources >
</ Job >
```

3. After Sources, inside the Job element create the *Layout* element: `<Layout></Layout>`

```
< Job >
< Sources >
< Source Id = "0" URL = "${SourceImage@0}" />
< Source Id = "1" URL = "${SourceXCM@0}" />
</ Sources >
< Layout >
</ Layout >
</ Job >
```

4. Inside the Layout element create the *Page* element: `<Page  Id></Page>` . You can create multiple Page elements.

```
< Job >
< Sources >
< Source Id = "0" URL = "${SourceImage@0}" />
< Source Id = "1" URL = "${SourceXCM@0}" />
</ Sources >
< Layout >
< Page Id = "0" width = "29.7 cm" height = "42 cm" >
</ Page >
</ Layout >
</ Job >
```

5. Inside the Page element create the *Objects* element: `<Objects></Objects>` and inside the Objects element create *Object* elements: `<Object Id></Object>` . You can create multiple Object elements.

```
< Job >
< Sources >
< Source Id = "0" URL = "${SourceImage@0}" />
< Source Id = "1" URL = "${SourceXCM@0}" />
</ Sources >
< Layout >
< Page Id = "0" width = "29.7 cm" height = "42 cm" >
< Objects >
< Object Id = "0" SourceID = "0" height = "42 cm" left = "0 cm" repeat = "rapport" rotation
= "0" top = "0 cm" width = "29.7 cm" >
</ Object >
</ Objects >
</ Page >
</ Layout >
</ Job >
```

6. Inside the Object elements create *Transformations* , *RapportInfo*, *SourceOffset x/y,* and *SourceSize h/w* elements:

```
< Job >
< Sources >
< Source Id = "0" URL = "${SourceImage@0}" />
< Source Id = "1" URL = "${SourceXCM@0}" />
</ Sources >
< Layout >
< Page Id = "0" width = "29.7 cm" height = "42 cm" >
< Objects >
< Object Id = "0" SourceID = "0" height = "42 cm" left = "0 cm" repeat = "rapport" rotation
= "0" top = "0 cm" width = "29.7 cm" >
< Transformations >
</ Transformations >
< RapportInfo direction = "${RapportInfoDirection@0}" fraction_high =
"${RapportInfoFraction_high@0}" fraction_low = "${RapportInfoFraction_low@0}" />
< SourceOffset x = "${SelectionRect_Origin_X@0}" y = "${SelectionRect_Origin_Y@0}" />
< SourceSize h = "29.7 cm" v = "42 cm" />
</ Object >
</ Objects >
</ Page >
</ Layout >
</ Job >
```

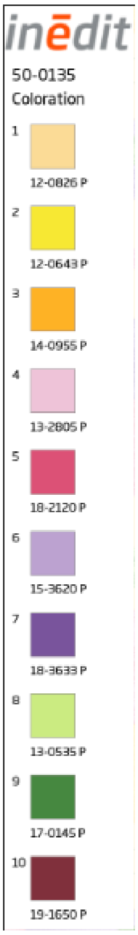
7. Inside the Transformations element create the *Coloration* element: `<Coloration/>`, or other Transformations and Advanced Transformations .

```
< Job >
< Sources >
< Source Id = "0" URL = "${SourceImage@0}" />
< Source Id = "1" URL = "${SourceXCM@0}" />
</ Sources >
< Layout >
< Page Id = "0" width = "29.7 cm" height = "42 cm" >
< Objects >
< Object Id = "0" SourceID = "0" height = "42 cm" left = "0 cm" repeat = "rapport" rotation
= "0" top = "0 cm" width = "29.7 cm" >
< Transformations >
< Coloration coloID = "1" specialcolor = "0" index = "0" /> </ Transformations >
< RapportInfo direction = "${RapportInfoDirection@0}" fraction_high =
"${RapportInfoFraction_high@0}" fraction_low = "${RapportInfoFraction_low@0}" />
< SourceOffset x = "${SelectionRect_Origin_X@0}" y = "${SelectionRect_Origin_Y@0}" />
< SourceSize h = "29.7 cm" v = "42 cm" />
</ Object >
</ Objects >
</ Page >
</ Layout >
</ Job >
```

8. Inside the Job element create the *Output* element `<Output></Output>` and inside the roots to Output settings.

```
< Job >
< Sources >
< Source Id = "0" URL = "${SourceImage@0}" />
< Source Id = "1" URL = "${SourceXCM@0}" />
</ Sources >
< Layout >
< Page Id = "0" width = "29.7 cm" height = "42 cm" >
< Objects >
< Object Id = "0" SourceID = "0" height = "42 cm" left = "0 cm" repeat = "rapport" rotation
= "0" top = "0 cm" width = "29.7 cm" >
< Transformations >
< Coloration coloID = "1" specialcolor = "0" index = "0" /> </ Transformations >
< RapportInfo direction = "${RapportInfoDirection@0}" fraction_high =
"${RapportInfoFraction_high@0}" fraction_low = "${RapportInfoFraction_low@0}" />
< SourceOffset x = "${SelectionRect_Origin_X@0}" y = "${SelectionRect_Origin_Y@0}" />
< SourceSize h = "29.7 cm" v = "42 cm" />
</ Object >
</ Objects >
</ Page >
</ Layout >
< Output >
< Space >LAB</ Space >
< BitsPerComponent >8</ BitsPerComponent >
< OutputEncodingOptions >
< CompressionType >lzw</ CompressionType >
</ OutputEncodingOptions >
</ Output >
< / Job >
```

Target 2: TJB with color patches and document info, position top-left in XML, Leve Basic



1. Follow the steps above from 1 to 7 as described above.

2. The *Sources* should contain files, besides the image and color source, the Logo, and Font:

```
< Sources >
< Source Id = "0" URL = "${SourceImage}" />
< Source Id = "1" URL = "${SourceXCM}" />
< Source Id = "2" URL = "SohoGothicPro-Regular.otf" />
< Source Id = "3" URL = "Logo.tif" />
</ Sources >
```

3. In the *Object's* element after the *Object* elements, add `<Repetition>` and `<Conditional>` elements, as shown in the script below (full TJB script sample here).

```
...
<!-- Color patches -->
<!-- Repetition and position of patches -->
< Repetition count = "${sourceid@1.Colorations.Coloration@0.ColorationItem.count}" maxSize =
"${__auto_parentHeight__}" direction = "V" height = "2.2 cm" item = "aColorationItem" left =
"0 points" list = "sourceid@1.Colorations.Coloration@0.ColorationItem" top = "3 cm" width =
"2.2 cm" >

< !-- Condition to hide/show background color patch -->
< Conditional condition = "${${aColorationItem}.channelNo}" equals = "-2" negate = "Y" >
<!-- Condition to hide disabled channels -->
< Conditional condition = "${${aColorationItem}.method}" equals = "Disable" negate = "Y" >
< Object Id = "0" left = "0.2 cm" top = "0.1 cm" >
< Transformations >
< Text fontID = "2" fontsize = "9" maxwidth = "85.4 points" value =
"${${aColorationItem}.channelIndex}" />
</ Transformations >
</ Object >

<!-- Patch gradient, size and border -->
< Object Id = "1" left = "0.7 cm" top = "0.2 cm" >
< Transformations >
< Gradient Space = "${${aColorationItem}.filtergradient.space}" UseProfile = "Y" Alias =
"${sourceid@1.Colorations.Coloration@0.ProfileInfo.Alias}" BorderColor =
"${${aColorationItem}.filterRGB#808080.value}" height = "34 points" percentages = "100"
percentbase = "100" width = "34 points" >${${aColorationItem}.filtergradient.value}</
Gradient >
</ Transformations >
</ Object >

<!-- Color name -->
< Object Id = "2" left = "0.7 cm" top = "1.5 cm" >
< Transformations >
< Text alignment = "left" fontID = "2" fontsize = "9" maxwidth = "3 cm" value =
"${${aColorationItem}.IData.name}" />
</ Transformations >
</ Object >
</ Conditional >

<!-- Condition to hide channel order when is disabled -->
< Conditional condition = "${${aColorationItem}.method}" equals = "Disable" negate = "N" >
< Object Id = "0" left = "0.2 cm" top = "0.1 cm" >
< Transformations >
< Text fontID = "2" fontsize = "9" maxwidth = "85.4 points" value =
"${${aColorationItem}.channelIndex}" />
</ Transformations >
</ Object >
</ Conditional >
</ Conditional >
</ Repetition >
...
```

4. Back in the XML file, add the new source for the TJB.

```
< Sources >
< Source Id = "0" URL = "${SourceImage@0}" />
< Source Id = "1" URL = "${SourceXCM@0}" />
< Source Id = "2" URL = "$SourceLayoutsFolderFOLDERNAME/LogoNameColorsCT_RS.tjb"
autoResolution = "disable" SetID = "0" />
</ Sources >
```

□□ Bear in mind when working with file sources in **neoColorBox** it requires adding the full folder path with the source file name: " `$Source` **`LayoutsFolderFOLDERNAME`** `/LogoNameColorsCT_RS.tjb`".

Related articles:

Default neoCatalog layouts description

neoCatalog layouts using custom fields variables

How to combine a seamless design repeat with a fixed logo repetition in XML Layout

Goal

Create a layout where the textile design repeats seamlessly, while an orange logo prints along the edge every 10 cm with a 2 cm offset.

1. Logo file

The logo saved in the required orange color (e.g., `Inedit.tif`).

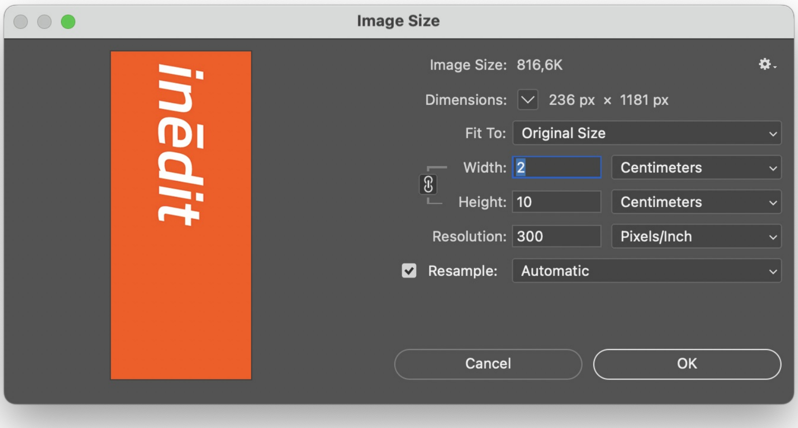


Image Size

Image Size: 816,6K

Dimensions: 236 px × 1181 px

Fit To: Original Size

Width: 2 Centimeters

Height: 10 Centimeters

Resolution: 300 Pixels/Inch

☑ Resample: Automatic

2. Layout structure

For the Design:

- **Position** : `left = 2 cm, top = 0 cm`
- **Size** : repeat width and height taken from the design itself.

- **Repeat mode** : `rapport` (continuous repetition).

For the Logo:

- **Position** : `left = 0 cm, top = 0 cm` (aligned to the edge).
- **Width** : logo width.
- **Height** : logo height.
- **Repeat mode :** `rapport` (continuous repetition).
- **Offset** : 2 cm from the edge.

Example XML (simplified):

```
<!-- Design -->
<Object Id="1" left="2 cm" top="0 cm" width="${RepeatWidth@0}" height="${RepeatHeight@0}"
MaintainAspectRatio="true" SourceID="0" autoResizeMask="Width|Height" autoResizePropHeight="1/1"
rotation="0" repeat="rapport">
  <Transformations>
<Coloration enable="${EnableColorway@0}" coloID = "1" specialcolor = "0" index =
"${ColorwayIndex@0}"/>
  </Transformations>
<RapportInfo direction="${RapportInfoDirection@0}" fraction_high="${RapportInfoFraction_high@0}"
fraction_low="${RapportInfoFraction_low@0}" />
<SourceOffset x="${SelectionRect_Origin_X@0}" y="${SelectionRect_Origin_Y@0}"/>
<SourceSize h="${RepeatWidth@0}" v="${RepeatHeight@0}"/>
</Object>


<!-- Logo -->
<Object Id="2" left="0 cm" top="0 cm" width="2cm" height="${RepeatHeight@0}"
MaintainAspectRatio="true" SourceID="10" autoResizeMask="Width|Height" autoResizePropHeight="1/1"
rotation="0" repeat="rapport">
<Transformations/>
<RapportInfo direction="${RapportInfoDirection@0}" fraction_high="${RapportInfoFraction_high@0}"
fraction_low="${RapportInfoFraction_low@0}" />
<SourceOffset x="${SelectionRect_Origin_X@1}" y="${SelectionRect_Origin_Y@1}"/>
<SourceSize h="2cm" v="${RepeatHeight@0}"/>
</Object>
```

3. Result

- The textile design repeats normally across the print.
- The orange logo is printed along the edge, repeating every 10 cm, shifted 2 cm from the margin.

Attachments:

INEDIT.xml
Inedit.tif

Layout Customisations for XML and TJB

neoRipEngine supports template features that can be used to completely customize the look and feel of the XML layout to the needs to integrate it with the image designs. The XML layouts are entirely XML format based to style the page. This document serves as an introduction to the methods available to customize the appearance and an introduction to how the XML layout is structured.

Layout customizations use a combination of XML and TJB. To achieve a particular requirement in the exported layout document file, you make changes in one or both of the following:

- The **XML** , where is used to define the structure of the exported content, including features such as the image position and repetition, simulation objects (VirtualVision), headers, and footers (defined in TJB).
- The **TJB** , where is used to define the style of elements in the exported content, such as font style, color sample patches, and the logo.
- **XML, TJB,** and sub files (logo, fonts, other files) must be stored together in one folder location.

This page provides information about advanced XML and TJB layout customizations. The information below is fo advanced users. Be aware that the advanced customizations described below require knowledge of certain parts of XML formatting. Customizations are not supported by Inedit, so our support engineers will not be able to help you with these modifications.

Inedit support for layout customization is limited. We support the mechanism for customizing the layout with XML, and we will help if the mechanism is broken or does not work as we say it should in our published examples But, since custom XML offers potentially limitless possibilities, Inedit will not support issues caused by or related to layout customizations.

□□ **Not supported:**

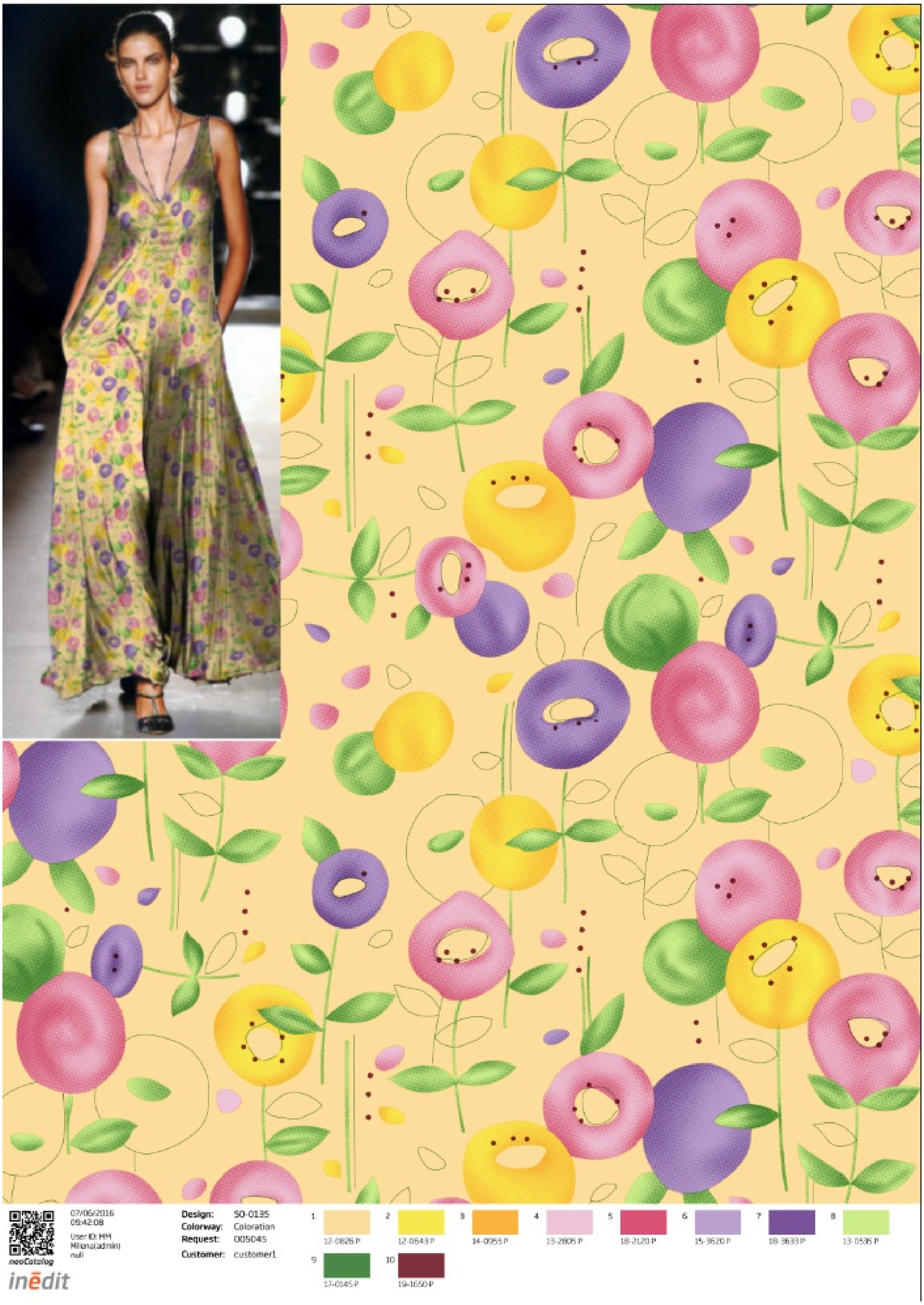
- Colored Background
- Position object relative to another object

TABLE OF CONTENTS

Examples of Customization

This section shows an example of typical customizations that you can add. (Sample layout is one of the default neoCatalog layouts Inedit_RealSize.xml).

Expected result



The layout contains information:

- Page Size: A3 vertical.
- 100% dimension.
- Design and Colorway name.
- Request Code.
- User ID, Name, and email.
- Date.
- QR code with design and colorway name.
- Logo.
- Channel color patches.
- Channel order.
- Color name in the color library.
- Customer.
- 1 VirtualVision Simulation **.**

XML Layout Structure

In the XML layout, we are using:

- Page size.
- Design image in real size.
- Colorway.
- VirtualVision and texture.
- Position of color patch samples and job info (TJB layout).

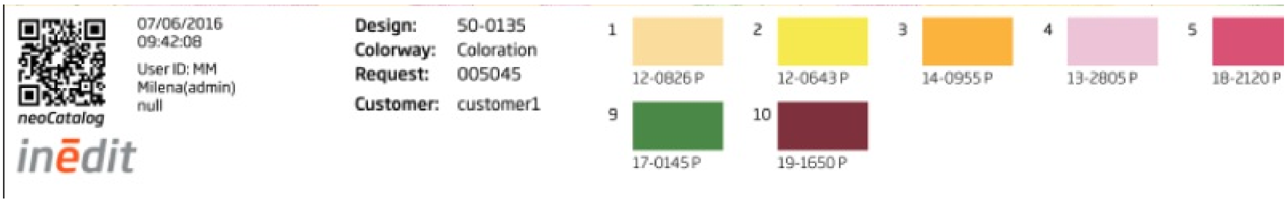
XML Sample

```
<?xml version="1.0" encoding="UTF-8"?>
<Job>
   <Sources>
      <Source Id="0" URL="${SourceImage@0}" />
      <Source Id="1" URL="${SourceXCM@0}" />
      <Source Id="2" URL="LogoNameColorsCT_RS.tjb" autoResolution="disable" SetID="0" />
      <Source Id="3" URL="${SourceVirtualVision}" />
      <Source Id="4" URL="Texture.tjb" autoResolution="disable" SetID="0" />
   </Sources>
   <Layout>
      <Page Id="0" width="29.7 cm" height="42 cm">
         <Objects>
            <!-- Image real size -->
            <Object Id="0" MaintainAspectRatio="true" SourceID="0" autoResizeMask="Width|Height"
autoResizePropHeight="1/1" height="42 cm" left="0 cm" repeat="rapport" rotation="0" top="0 cm"
width="29.7 cm">
               <Transformations>
                  <Coloration coloID="1" specialcolor="0" index="0" />
               </Transformations>
               <RapportInfo direction="${RapportInfoDirection@0}"
fraction_high="${RapportInfoFraction_high@0}" fraction_low="${RapportInfoFraction_low@0}" />
               <SourceOffset x="${SelectionRect_Origin_X@0}" y="${SelectionRect_Origin_Y@0}" />
               <SourceSize h="29.7 cm" v="42 cm" />
            </Object>
            <!-- VirtualVision -->
            <Object Id="1" SourceID="3" top="0 cm" left="0 cm" width="9 cm" insideWidth="1 rep"
insideHeight="1 rep" maintainAspectRatio="true" height="42 cm">
               <Transformations>
                  <VirtualVision previewMode="Normal" background="Image" shadows="yes">
                     <Collection Id="0" mode="Image" textureID="4" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                     <Collection Id="1" mode="Image" textureID="4" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                     <Collection Id="2" mode="Image" textureID="4" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                  </VirtualVision>
               </Transformations>
            </Object>
            <!-- Color Samples and Job Info -->
            <Object Id="2" SourceID="2" MaintainAspectRatio="true" autoResizeMask="Top|Width"
autoResizePropTop="1/1" autoResizePropWidth="1/1" autoPositionMask="Top" left="0 cm" top="40.7
cm" width="29.7 cm" />
         </Objects>
      </Page>
   </Layout>
   <Output>
      <Space>LAB</Space>
      <BitsPerComponent>8</BitsPerComponent>
      <OutputEncodingOptions>
         <CompressionType>lzw</CompressionType>
      </OutputEncodingOptions>
   </Output>
</Job>
```

TJB Sample Layout Structure

In the TJB layout, we are using:

- Design and Colorway name.
- Request Code.
- User ID, Name, and email.
- Date.
- QR code with design and colorway name.
- Logo.
- Channel color patches.
- Channel order.
- Color name in the color library.
- Customer.
- Texture for VirtualVision.



07/06/2016
09:42:08

User ID: MM
Milena(admin)
null

Design: 50-0135
Colorway: Coloration
Request: 005045
Customer: customer1

TJB color patches and Job info

```
<?xml version="1.0" encoding="UTF-8"?>
<Job>
   <Sources>
      <Source Id="0" URL="${SourceImage}" />
      <Source Id="1" URL="${SourceXCM}" />
      <Source Id="2" URL="SohoGothicPro-Medium.otf" />
      <Source Id="3" URL="SohoGothicPro-Regular.otf" />
      <Source Id="4" URL="SohoGothicPro-Light.otf" />
      <Source Id="5" URL="Logo.tif" />
      <Source Id="6" URL="X_block.tif" />
      <Source Id="7" URL="logo_neoCatalogBlack.psd" />
   </Sources>
   <Layout>
      <Page Id="0" width="${__auto_parentWidth__}">
         <Objects>
            <!-- PSD logo (logo_neoCatalogBlack.psd) -->
            <Object Id="0" SourceID="7" left="0.2 cm" top="1.7 cm" width="1.5 cm" height="1cm"
MaintainAspectRatio="true" />
            <!-- Tif logo (Logo.tif) -->
            <Object Id="1" SourceID="5" left="0.2 cm" top="2.2 cm" height="1cm"
MaintainAspectRatio="true" />
            <!-- Text placeholder Design -->
            <Object Id="2" Left="5.8 cm" Top="0.1 cm">
               <Transformations>
                  <Text alignment="left" fontid="2" fontsize="8" maxwidth="2 cm" value="Design:"
/>
               </Transformations>
            </Object>
            <!-- Design Name -->
            <Object Id="2" left="7.5 cm" top="0.1 cm">
               <Transformations>
                  <Text alignment="left" fontid="3" fontsize="8" maxwidth="2 cm"
value="${SourceID@1.Colorations.Name}" />
               </Transformations>
            </Object>
            <!-- Text placeholder Colorway -->
            <Object Id="3" Left="5.8 cm" Top="0.5 cm">
               <Transformations>
                  <Text alignment="left" fontid="2" fontsize="8" maxwidth="2 cm"
value="Colorway:" />
               </Transformations>
            </Object>
            <!-- Colorway Name -->
            <Object Id="4" Left="7.5 cm" Top="0.5 cm">
               <Transformations>
                  <Text alignment="left" fontid="3" fontsize="8" maxwidth="2 cm"
value="${SourceID@1.Colorations.Coloration@0.IData.__USR_CLR_CW_colorwayName}" />
               </Transformations>
            </Object>
            <!-- Text placeholder Request -->
            <Object Id="5" Left="5.8 cm" Top="0.9 cm">
               <Transformations>
                  <Text alignment="left" fontid="2" fontsize="8" maxwidth="2 cm"
value="Request:" />
               </Transformations>
            </Object>
            <!-- Request Code -->
            <Object Id="6" Left="7.5 cm" Top="0.9 cm">
               <Transformations>
                  <Text alignment="left" fontid="3" fontsize="8" maxwidth="2 cm"
value="${RequestCode}" />
               </Transformations>
            </Object>
            <!-- Text placeholder Customer -->
            <Object Id="7" Left="5.8 cm" Top="1.4 cm">
               <Transformations>
                  <Text alignment="left" fontid="2" fontsize="8" maxwidth="2 cm"
value="Customer:" />
               </Transformations>
            </Object>
            <!-- Customer Name -->
            <Object Id="8" Left="7.5 cm" Top="1.4 cm">
               <Transformations>
                  <Text alignment="left" fontid="3" fontsize="8" maxwidth="3 cm"
value="${CustomerDescription}" />
               </Transformations>
            </Object>
            <!-- Color patches -->
            <!-- Repetition and position of patches -->
            <Repetition count="${sourceid@1.Colorations.Coloration@0.ColorationItem.count}"
maxSize="${__auto_parentWidth__}" direction="H" height="1.4 cm" item="aColorationItem" left="10
cm" list="sourceid@1.Colorations.Coloration@0.ColorationItem" top="0.2 cm" width="2.4 cm">
               <!-- Condition to hide background and disabled channel -->
               <Conditional condition="${${aColorationItem}.channelNo}" equals="-2">
                  <Conditional condition="${${aColorationItem}.method}" equals="Disable"
negate="y">
                     <!-- Channel order -->
                     <Object Id="8" Left="0 cm" Top="0 cm">
                        <Transformations>
                           <Text alignment="left" fontid="3" fontsize="7" maxwidth="0.2 cm"
value="${${aColorationItem}.ChannelIndex}" />
                        </Transformations>
                     </Object>
                     <!-- Color name -->
                     <Object Id="9" left="0.4 cm" top="0.8 cm">
                        <Transformations>
                           <Text alignment="left" fontID="4" fontsize="7" maxwidth="1.5 cm"
value="${${aColorationItem}.IData.name}" />
                        </Transformations>
                     </Object>
                     <!-- Patch gradient and size -->
                     <Object Id="10" left="0.4 cm" top="0 cm">
                        <Transformations>
                           <Gradient Space="${${aColorationItem}.filtergradient.space}"
UseProfile="Y" Alias="${sourceid@1.Colorations.Coloration@0.ProfileInfo.Alias}" height="0.8 cm"
percentages="100" percentbase="100" width="1.5 cm">${${aColorationItem}.filtergradient.value}
</Gradient>
                        </Transformations>
                     </Object>
                  </Conditional>
               </Conditional>
               <!-- Condition to show X image and hide color name when channel is disabled  -->
               <Conditional condition="${${aColorationItem}.channelNo}" equals="-2" negate="y">
                  <Object Id="8" Left="0 cm" Top="0 cm">
                     <Transformations>
                        <Text alignment="left" fontid="3" fontsize="7" maxwidth="0.2 cm"
value="${${aColorationItem}.ChannelIndex}" />
                     </Transformations>
                  </Object>
                  <Conditional condition="${${aColorationItem}.method}" equals="Disable"
negate="y">
                     <Object Id="9" left="0.4 cm" top="0.8 cm">
                        <Transformations>
                           <Text alignment="left" fontID="4" fontsize="7" maxwidth="1.5 cm"
value="${${aColorationItem}.IData.name}" />
                        </Transformations>
                     </Object>
                     <Object Id="10" left="0.4 cm" top="0 cm">
                        <Transformations>
                           <Gradient Space="${${aColorationItem}.filtergradient.space}"
UseProfile="Y" Alias="${sourceid@1.Colorations.Coloration@0.ProfileInfo.Alias}" height="0.8 cm"
```

```
percentages="100" percentbase="100" width="1.5 cm">${${aColorationItem}.filtergradient.value}
</Gradient>
                        </Transformations>
                     </Object>
                  </Conditional>
                  <Conditional condition="${${aColorationItem}.method}" equals="Disable"
negate="n">
                     <Object Id="11" SourceID="6" left="0.4 cm" top="0 cm"
MaintainAspectRatio="true" />
                  </Conditional>
               </Conditional>
            </Repetition>
            <!-- Date -->
            <Object Id="12" Left="2.2 cm" Top="0.1 cm">
               <Transformations>
                  <Text alignment="left" fontid="3" fontsize="7" maxwidth="4 cm"
value="${System.Date.formater#%d/%m/%Y}" />
               </Transformations>
            </Object>
            <!-- Time -->
            <Object Id="13" Left="2.2 cm" Top="0.4 cm">
               <Transformations>
                  <Text alignment="left" fontid="3" fontsize="7" maxwidth="4 cm"
value="${System.Time.formater#%H:%M:%S}" />
               </Transformations>
            </Object>
            <!-- User ID -->
            <Object Id="14" Left="2.2 cm" Top="0.85 cm">
               <Transformations>
                  <Text alignment="left" fontid="3" fontsize="7" maxwidth="5 cm" value="User ID:
${UserID}" />
               </Transformations>
            </Object>
            <!-- User name -->
            <Object Id="15" Left="2.2 cm" Top="1.15 cm">
               <Transformations>
                  <Text alignment="left" fontid="3" fontsize="7" value="${UserDescription}" />
               </Transformations>
            </Object>
            <!-- User email -->
            <Object Id="16" left="2.2 cm" Top="1.45 cm">
               <Transformations>
                  <Text alignment="left" fontid="3" fontsize="7" value="${UserMail}" />
               </Transformations>
            </Object>
            <!--QR CODE-->
            <Object Id="17" width="1.5cm" height="1.5cm" left="0.2cm" top="0.2cm">
               <Transformations>
                  <Code value="${SourceID@1.Colorations.Name}
${SourceID@1.Colorations.Coloration@0.Name}" type="QR" />
               </Transformations>
            </Object>
         </Objects>
      </Page>
   </Layout>
   <Output>
      <Space>LAB</Space>
   </Output>
</Job>
```

TJB texture for simulation

```
<?xml version="1.0" encoding="UTF-8"?>
<Job>
   <Sources>
      <Source Id="0" URL="${SourceImage}" />
      <Source Id="1" URL="${SourceXCM}" />
   </Sources>
   <Layout>
      <Page Id="0">
         <Objects>
            <Object Id="0" MaintainAspectRatio="true" SourceID="0">
               <Transformations>
                  <Coloration coloID="1" index="0" />
               </Transformations>
            </Object>
         </Objects>
         <RapportInfo direction="${RapportInfoDirection@0}"
fraction_high="${RapportInfoFraction_high@0}" fraction_low="${RapportInfoFraction_low@0}" />
      </Page>
   </Layout>
   <Output>
      <ResolutionX>72</ResolutionX>
      <ResolutionY>72</ResolutionY>
      <ResolutionUnits>dpi</ResolutionUnits>
      <Space>LAB</Space>
   </Output>
</Job>
```

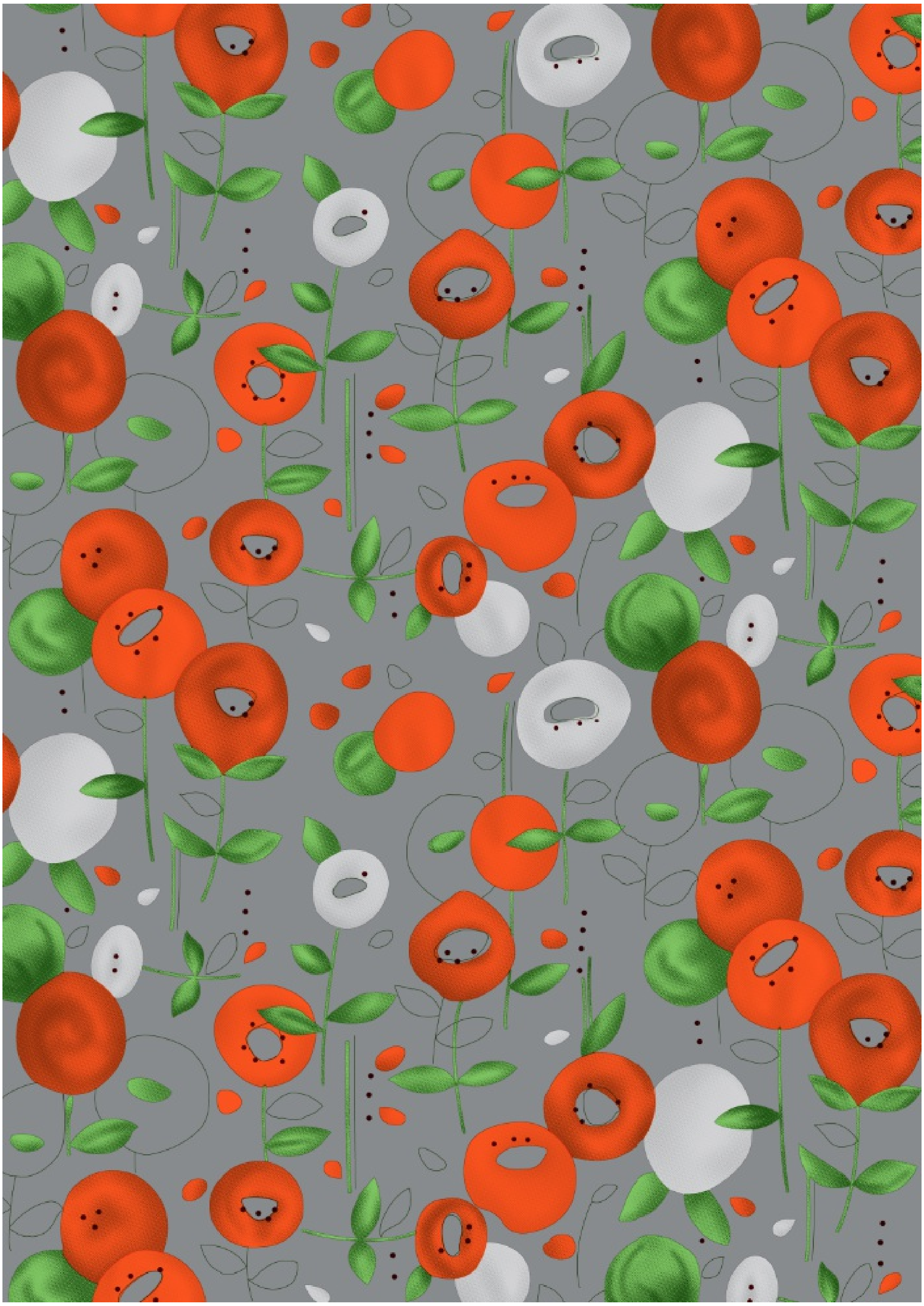
Practical examples

This section provides practical examples of sample customizations above, which are built object for object.

XML Layout

(A) Design Image

To apply the design image only, we are using the following object structure.



```
<?xml version="1.0" encoding="UTF-8"?>
<Job>
   <Sources>
      <Source Id="0" URL="${SourceImage@0}" />
      <Source Id="1" URL="${SourceXCM@0}" />
   </Sources>
   <Layout>
      <Page Id="0" width="29.7 cm" height="42 cm">
         <Objects>
            <!-- Image real size -->
            <Object Id="0" MaintainAspectRatio="true" SourceID="0" autoResizeMask="Width|Height"
autoResizePropHeight="1/1" height="42 cm" left="0 cm" repeat="rapport" rotation="0" top="0 cm"
width="29.7 cm">
               <Transformations>
                  <Coloration coloID="1" specialcolor="0" index="0" />
               </Transformations>
               <RapportInfo direction="${RapportInfoDirection@0}"
fraction_high="${RapportInfoFraction_high@0}" fraction_low="${RapportInfoFraction_low@0}" />
               <SourceOffset x="${SelectionRect_Origin_X@0}" y="${SelectionRect_Origin_Y@0}" />
               <SourceSize h="29.7 cm" v="42 cm" />
            </Object>
         </Objects>
      </Page>
   </Layout>
   <Output>
      <Space>LAB</Space>
      <BitsPerComponent>8</BitsPerComponent>
      <OutputEncodingOptions>
         <CompressionType>lzw</CompressionType>
      </OutputEncodingOptions>
   </Output>
</Job>
```

(B) Design Image and VirtualVision

We use the following structure object to apply the design image and virtual vision.

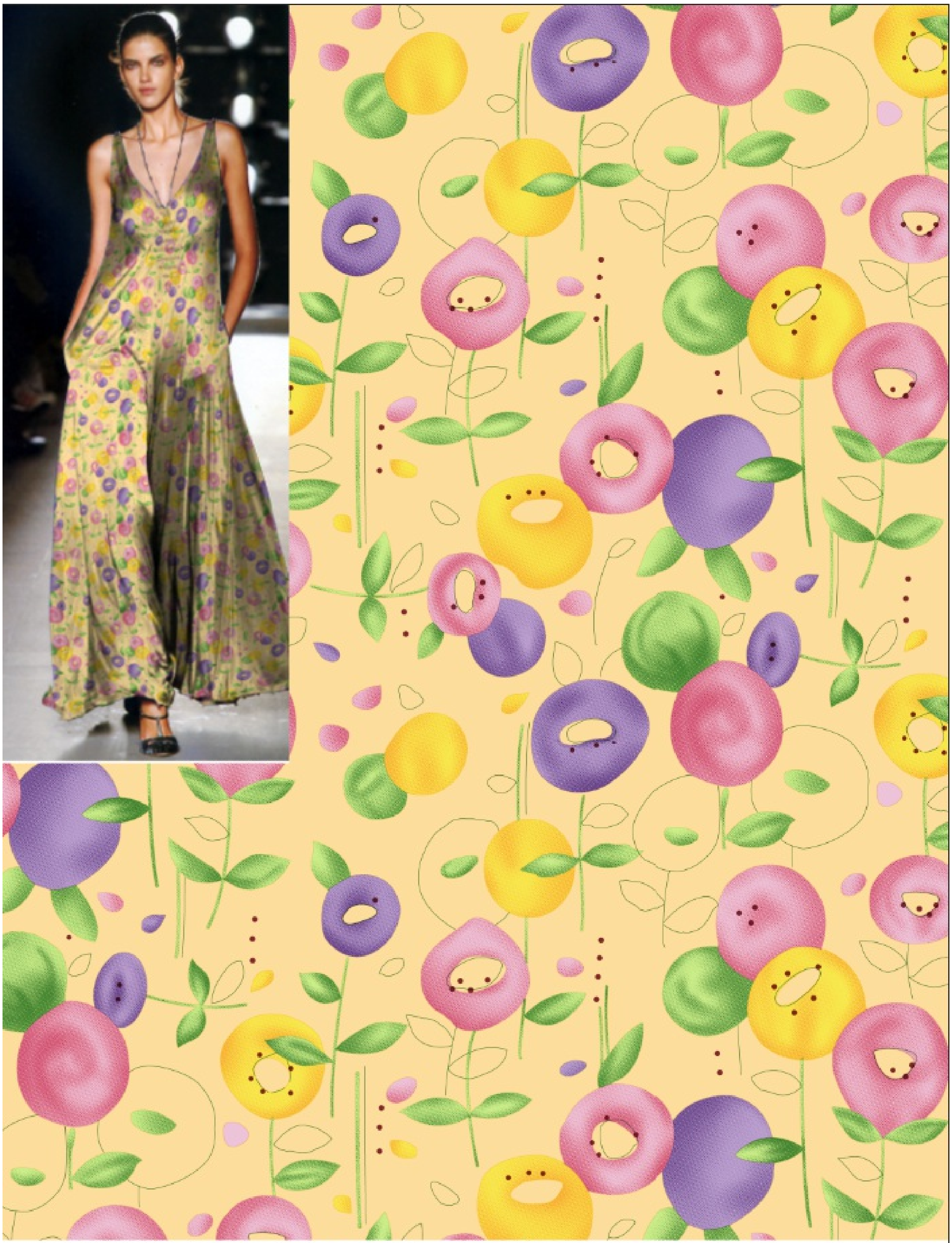
□□ **Before you begin:** To define the virtual vision transformation, the file Texture.tjb must exist, which includes the minimum requirements to apply image texture.

```
<?xml version="1.0" encoding="UTF-8"?>
<Job>
   <Sources>
      <Source Id="0" URL="${SourceImage@0}" />
      <Source Id="1" URL="${SourceXCM@0}" />
      <Source Id="2" URL="${SourceVirtualVision}" />
      <Source Id="3" URL="Texture.tjb" autoResolution="disable" SetID="0" />
   </Sources>
   <Layout>
      <Page Id="0" width="29.7 cm" height="42 cm">
         <Objects>
            <!-- Image real size -->
            <Object Id="0" MaintainAspectRatio="true" SourceID="0" autoResizeMask="Width|Height"
autoResizePropHeight="1/1" height="42 cm" left="0 cm" repeat="rapport" rotation="0" top="0 cm"
width="29.7 cm">
               <Transformations>
                  <Coloration coloID="1" specialcolor="0" index="0" />
               </Transformations>
               <RapportInfo direction="${RapportInfoDirection@0}"
fraction_high="${RapportInfoFraction_high@0}" fraction_low="${RapportInfoFraction_low@0}" />
               <SourceOffset x="${SelectionRect_Origin_X@0}" y="${SelectionRect_Origin_Y@0}" />
               <SourceSize h="29.7 cm" v="42 cm" />
            </Object>
            <!-- VirtualVision -->
            <Object Id="1" SourceID="2" top="0 cm" left="0 cm" width="9 cm" insideWidth="1 rep"
insideHeight="1 rep" maintainAspectRatio="true" height="42 cm">
               <Transformations>
                  <VirtualVision previewMode="Normal" background="Image" shadows="yes">
                     <Collection Id="0" mode="Image" textureID="3" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                     <Collection Id="1" mode="Image" textureID="3" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                     <Collection Id="2" mode="Image" textureID="3" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                  </VirtualVision>
               </Transformations>
            </Object>
         </Objects>
      </Page>
   </Layout>
   <Output>
      <Space>LAB</Space>
      <BitsPerComponent>8</BitsPerComponent>
      <OutputEncodingOptions>
         <CompressionType>lzw</CompressionType>
      </OutputEncodingOptions>
   </Output>
</Job>
```

(C) Design Image, VirtualVision and Color sample patches with Job info (TJB)

To apply the design image, virtual vision, color patches plus Job info we are using the following structure object.

> □□ If the design is RGB image (digital design), the color patches are not existing and are automatically hidden.



07/06/2016
09:42:08

User ID: MM
Milena(admin)
null

neoCatalog

inēdit

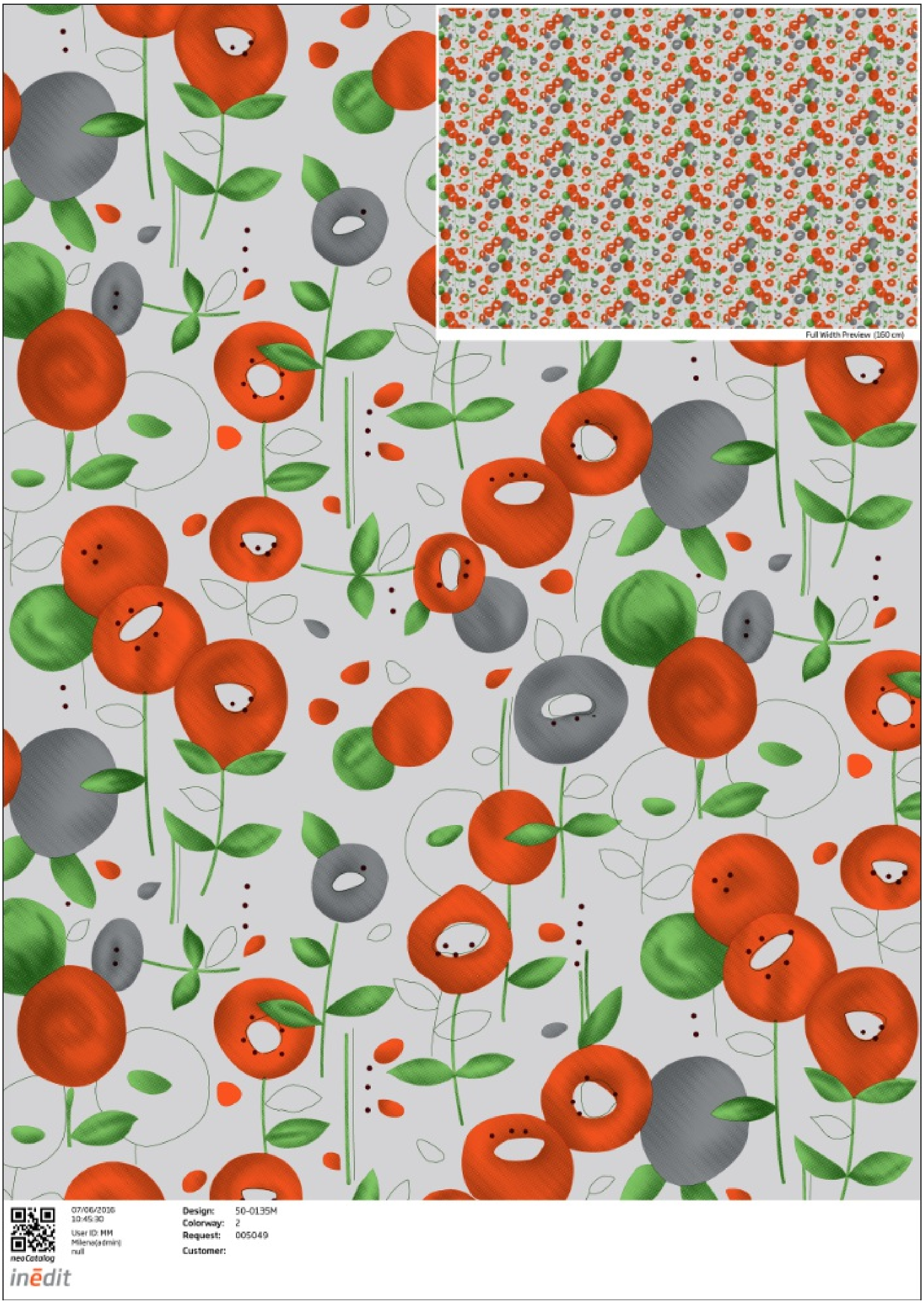
Design: SO-0135
Colorway: Coloration
Request: 005045
Customer: customer1

1	2	3	4	5	6	7	8
12-0826 P	12-0643 P	14-0955 P	13-2806 P	18-2120 P	15-3620 P	18-3633 P	13-0535 P
9	10						
17-0145 P	19-1650 P						

```
<?xml version="1.0" encoding="UTF-8"?>
<Job>
   <Sources>
      <Source Id="0" URL="${SourceImage@0}" />
      <Source Id="1" URL="${SourceXCM@0}" />
      <Source Id="2" URL="LogoNameColorsCT_RS.tjb" autoResolution="disable" SetID="0" />
      <Source Id="3" URL="${SourceVirtualVision}" />
      <Source Id="4" URL="Texture.tjb" autoResolution="disable" SetID="0" />
   </Sources>
   <Layout>
      <Page Id="0" width="29.7 cm" height="42 cm">
         <Objects>
            <!-- Image real size -->
            <Object Id="0" MaintainAspectRatio="true" SourceID="0" autoResizeMask="Width|Height"
autoResizePropHeight="1/1" height="42 cm" left="0 cm" repeat="rapport" rotation="0" top="0 cm"
width="29.7 cm">
               <Transformations>
                  <Coloration coloID="1" specialcolor="0" index="0" />
               </Transformations>
               <RapportInfo direction="${RapportInfoDirection@0}"
fraction_high="${RapportInfoFraction_high@0}" fraction_low="${RapportInfoFraction_low@0}" />
               <SourceOffset x="${SelectionRect_Origin_X@0}" y="${SelectionRect_Origin_Y@0}" />
               <SourceSize h="29.7 cm" v="42 cm" />
            </Object>
            <!-- VirtualVision -->
            <Object Id="1" SourceID="3" top="0 cm" left="0 cm" width="9 cm" insideWidth="1 rep"
insideHeight="1 rep" maintainAspectRatio="true" height="42 cm">
               <Transformations>
                  <VirtualVision previewMode="Normal" background="Image" shadows="yes">
                      <Collection Id="0" mode="Image" textureID="4" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                      <Collection Id="1" mode="Image" textureID="4" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                      <Collection Id="2" mode="Image" textureID="4" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                  </VirtualVision>
               </Transformations>
            </Object>
            <!-- Color Samples and Job Info -->
            <Object Id="2" SourceID="2" MaintainAspectRatio="true" autoResizeMask="Top|Width"
autoResizePropTop="1/1" autoResizePropWidth="1/1" autoPositionMask="Top" left="0 cm" top="40.7
cm" width="29.7 cm" />
         </Objects>
      </Page>
   </Layout>
   <Output>
      <Space>LAB</Space>
      <BitsPerComponent>8</BitsPerComponent>
      <OutputEncodingOptions>
         <CompressionType>lzw</CompressionType>
      </OutputEncodingOptions>
   </Output>
</Job>
```

Repetition preview

To apply the design image in repetitions showing as a small preview, we need to create a TJB that includes the repetition information.



Full Width Preview (160 cm)

neoCatalog

inēdit

07/06/2016
10:45:30

User ID: MM
Milena(admin)
null

Design: 50-0135M
Colorway: 2
Request: 005049
Customer:

TJB for Preview

```
<?xml version="1.0" encoding="UTF-8"?>
<Job>
   <Sources>
      <Source Id="0" URL="${SourceImage}" />
      <Source Id="1" URL="${SourceXCM}" />
      <Source Id="2" URL="SohoGothicPro-Regular.otf" />
   </Sources>
   <Layout>
      <Page Id="0" width="15.7 cm" height="10.50 cm">
         <Objects>
            <!-- Preview object size with white border -->
            <Object Id="0" Left="0 pt" Top="10.9 pt">
               <Transformations>
                  <Gradient left="0 cm" top="0 cm" width="15.7 cm" height="10.9 cm"
Percentages="100" Percentbase="100" Space="LAB32" UseProfile="N">100,0,0</Gradient>
               </Transformations>
            </Object>
            <!-- Preview image with repetition width 160cm -->
            <Object Id="2" SourceID="0" left="0.1 cm" top="0.1 cm" width="15.5 cm" height="10.5
cm" MaintainAspectRatio="true" repeat="rapport">
               <Transformations>
                  <Coloration coloID="1" specialcolor="0" index="0" />
               </Transformations>
               <RapportInfo direction="${RapportInfoDirection}"
fraction_high="${RapportInfoFraction_high}" fraction_low="${RapportInfoFraction_low}" />
               <SourceOffset x="0" y="0" />
               <SourceSize h="160 cm" v="5 cm" />
            </Object>
         </Objects>
      </Page>
      <!-- Text (optional) -->
      <Page Id="1" width="15.7 cm" height="0.40 cm">
         <Objects>
            <Object Id="1" Left="12 cm" Top="0 cm">
               <Transformations>
                  <Text alignment="left" fontid="2" fontsize="7" maxwidth="5 cm" value="Full
Width Preview  (160 cm)" />
               </Transformations>
            </Object>
         </Objects>
      </Page>
   </Layout>
   <Output>
      <Space>LAB</Space>
      <BitsPerComponent>16</BitsPerComponent>
   </Output>
</Job>
```

As the preview is positioned on the right side of the layout, we need to keep the preview when resizing the layout. The attributes in this case are:

- autoResizeMask="Left|Width"
- autoResizePropLeft="1/1"
- autoResizePropWidth="1/1"

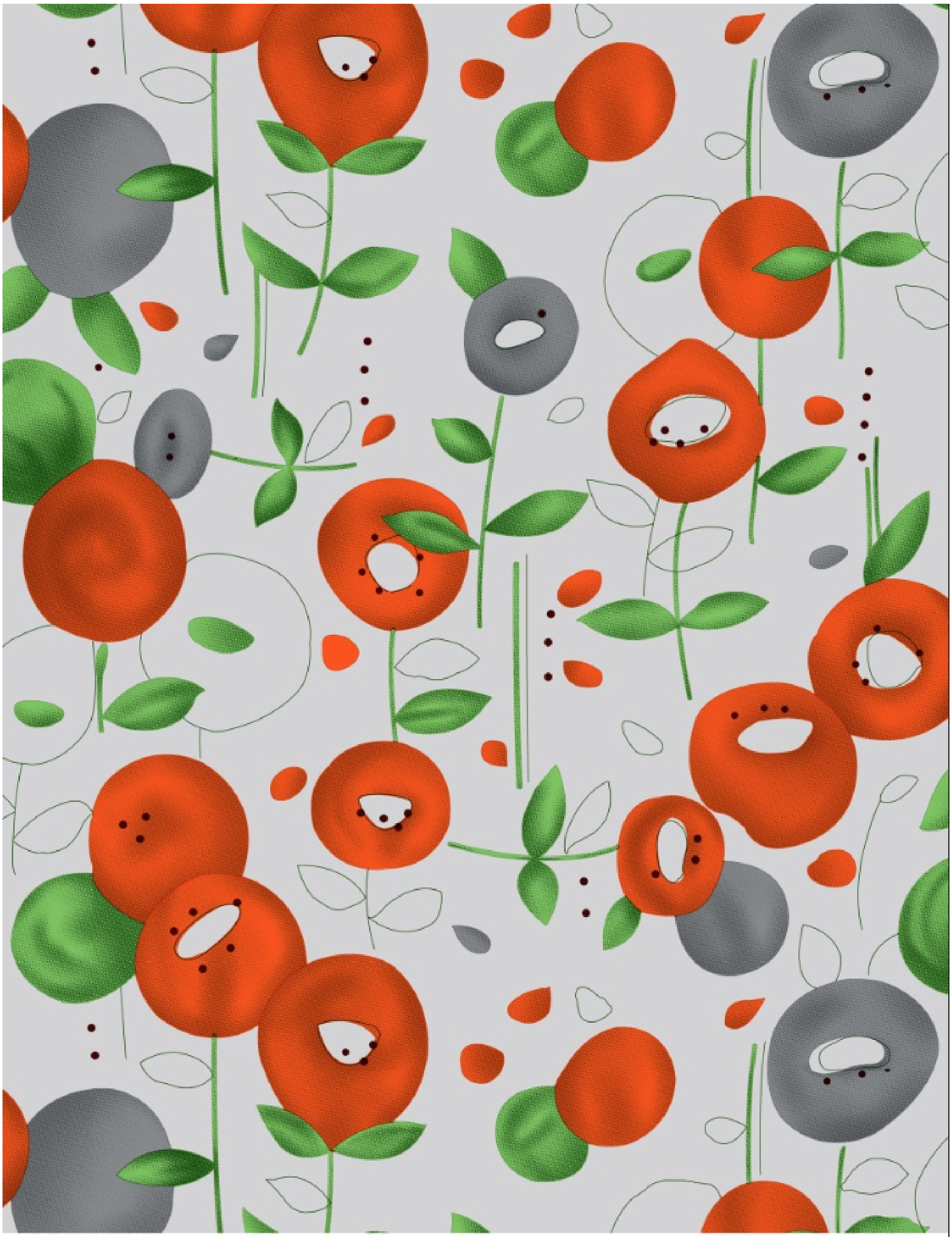
XML with TJB preview position

```
<?xml version="1.0" encoding="UTF-8"?>
<Job>
   <Sources>
      <Source Id="0" URL="${SourceImage@0}" />
      <Source Id="1" URL="${SourceXCM@0}" />
      <Source Id="2" URL="LogoNameColorsCT_RS.tjb" autoResolution="disable" SetID="0" />
      <Source Id="3" URL="${SourceVirtualVision}" />
      <Source Id="4" URL="Texture.tjb" autoResolution="disable" SetID="0" />
      <Source Id="5" URL="Preview.tjb" autoResolution="disable" SetID="0" />
   </Sources>
   <Layout>
      <Page Id="0" width="29.7 cm" height="42 cm">
         <Objects>
            <!-- Image real size -->
            <Object Id="0" MaintainAspectRatio="true" SourceID="0" autoResizeMask="Width|Height"
autoResizePropHeight="1/1" height="42 cm" left="0 cm" repeat="rapport" rotation="0" top="0 cm"
width="29.7 cm">
               <Transformations>
                  <Coloration coloID="1" specialcolor="0" index="0" />
               </Transformations>
               <RapportInfo direction="${RapportInfoDirection@0}"
fraction_high="${RapportInfoFraction_high@0}" fraction_low="${RapportInfoFraction_low@0}" />
               <SourceOffset x="${SelectionRect_Origin_X@0}" y="${SelectionRect_Origin_Y@0}" />
               <SourceSize h="29.7 cm" v="42 cm" />
            </Object>
            <!-- Preview image with repetition width 160cm -->
            <Object Id="1" MaintainAspectRatio="true" autoResizeMask="Left|Width"
autoResizePropLeft="1/1" autoResizePropWidth="1/1" autoPositionMask="Left" SourceID="5" top="0
cm" left="22.8 cm" />
            <!-- Virtual Vision (optional) -->
            <Object Id="2" SourceID="3" top="0 cm" left="0 cm" width="9 cm" insideWidth="1 rep"
insideHeight="1 rep" maintainAspectRatio="true" height="42 cm">
               <Transformations>
                  <VirtualVision previewMode="Normal" background="Image" shadows="yes">
                     <Collection Id="0" mode="Image" textureID="4" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                     <Collection Id="1" mode="Image" textureID="4" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                     <Collection Id="2" mode="Image" textureID="4" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                  </VirtualVision>
               </Transformations>
            </Object>
            <!-- Color Samples and Job Info -->
            <Object Id="3" SourceID="2" MaintainAspectRatio="true" autoResizeMask="Top|Width"
autoResizePropTop="1/1" autoResizePropWidth="1/1" autoPositionMask="Top" left="0 cm" top="40.7
cm" width="29.7 cm" />
         </Objects>
      </Page>
   </Layout>
   <Output>
      <Space>LAB</Space>
      <BitsPerComponent>8</BitsPerComponent>
      <OutputEncodingOptions>
         <CompressionType>lzw</CompressionType>
      </OutputEncodingOptions>
   </Output>
</Job>
```

Full Rapport

To apply the design image in full rapport.

> □□ Fixed warning text to notice in embedded in the TJB that the image is not in real size.



neoCatalog

07/06/2016
10:45:43

User ID: MM
Milena(admin)
null

Design:	50-0135M
Colorway:	2
Request:	005050
Customer:	

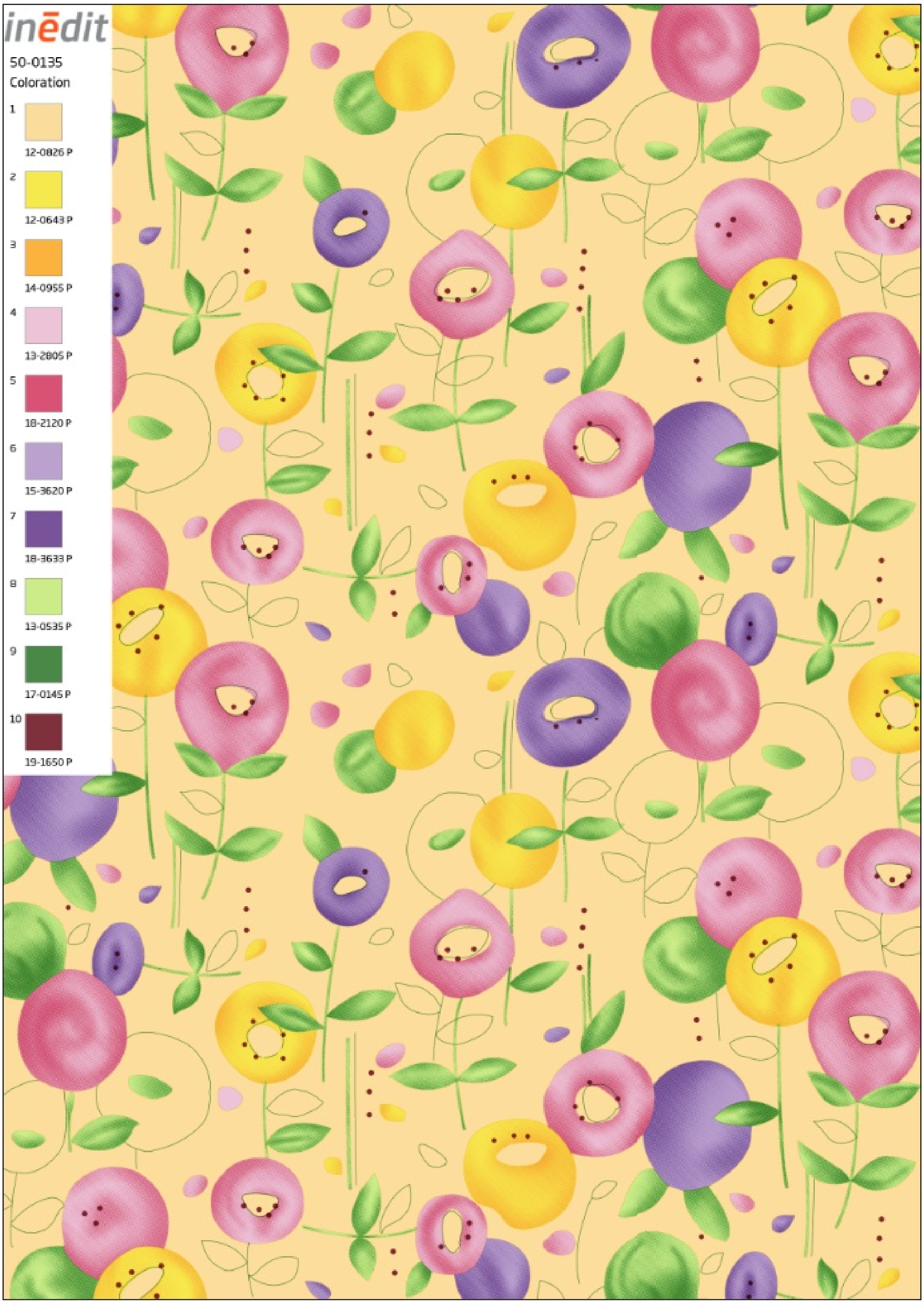
Attention: Full Rapport. Not real size dimension

inēdit

```
<?xml version="1.0" encoding="UTF-8"?>
<Job>
   <Sources>
      <Source Id="0" URL="${SourceImage@0}" />
      <Source Id="1" URL="${SourceXCM@0}" />
      <Source Id="2" URL="LogoNameColorsCT_RS.tjb" autoResolution="disable" SetID="0" />
      <Source Id="3" URL="${SourceVirtualVision}" />
      <Source Id="4" URL="Texture.tjb" autoResolution="disable" SetID="0" />
   </Sources>
   <Layout>
      <Page Id="0" width="29.7 cm" height="42 cm">
         <Objects>
            <!-- Image Full Rapport -->
            <Object Id="0" MaintainAspectRatio="true" SourceID="0" autoResizeMask="Width|Height"
autoResizePropHeight="1/1" height="42 cm" left="0 cm" repeat="rapport" rotation="0" top="0 cm"
width="29.7 cm">
               <Transformations>
                  <Coloration coloID="1" specialcolor="0" index="0" />
               </Transformations>
               <RapportInfo direction="${RapportInfoDirection@0}"
fraction_high="${RapportInfoFraction_high@0}" fraction_low="${RapportInfoFraction_low@0}" />
               <SourceOffset x="${SelectionRect_Origin_X@0}" y="${SelectionRect_Origin_Y@0}" />
               <SourceSize h="1 rep" v="1 rep" />
            </Object>
            <!-- VirtualVision -->
            <Object Id="1" SourceID="3" top="0 cm" left="0 cm" width="9 cm" insideWidth="1 rep"
insideHeight="1 rep" maintainAspectRatio="true" height="42 cm">
               <Transformations>
                  <VirtualVision previewMode="Normal" background="Image" shadows="yes">
                     <Collection Id="0" mode="Image" textureID="4" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                     <Collection Id="1" mode="Image" textureID="4" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                     <Collection Id="2" mode="Image" textureID="4" rotation="0" offsetX="0"
offsetY="0" scale="1.0" />
                  </VirtualVision>
               </Transformations>
            </Object>
            <!-- Color Samples and Job Info -->
            <Object Id="2" SourceID="2" MaintainAspectRatio="true" autoResizeMask="Top|Width"
autoResizePropTop="1/1" autoResizePropWidth="1/1" autoPositionMask="Top" left="0 cm" top="40.7
cm" width="29.7 cm" />
         </Objects>
      </Page>
   </Layout>
   <Output>
      <Space>LAB</Space>
      <BitsPerComponent>8</BitsPerComponent>
      <OutputEncodingOptions>
         <CompressionType>lzw</CompressionType>
      </OutputEncodingOptions>
   </Output>
</Job>
```

Vertical color sample patches

To apply the simple color patches in vertical order in TJB with the color name, channel order, design, and colorway name in XML.


inēdit
50-0135
Coloration
1
12-0826 P
2
12-0643 P
3
14-0955 P
4
13-2805 P
5
18-2120 P
6
15-3620 P
7
18-3633 P
8
13-0535 P
9
17-0145 P
10
19-1650 P

TJB Sample

```
<?xml version="1.0" encoding="UTF-8"?>
<Job>
   <Sources>
      <Source Id="0" URL="${SourceImage}" />
      <Source Id="1" URL="${SourceXCM}" />

      <Source Id="2" URL="SohoGothicPro-Regular.otf" />

      <Source Id="3" URL="Logo.tif" />

   </Sources>
   <Layout>
      <Page Id="0">
```

```
         <Objects>

            <!-- Tif logo (Logo.tif) -->

            <Object Id="0" SourceID="3" height="1 cm" left="0 cm" top="0.2 cm" width="3.5 cm"
MaintainAspectRatio="true">

               <Transformations></Transformations>

            </Object>

            <!-- Design and Colorway name -->

            <Object Id="1" left="0.2 cm" top="1.5 cm">

               <Transformations>

                  <Text alignment="left" fontID="2" fontsize="12" maxwidth="99 points"
value="${sourceid@0.Document.name} ${sourceid@1.Colorations.Coloration@0.Name}" />

               </Transformations>
            </Object>

            <!-- Color patches -->
            <!-- Repetition and position of patches -->

            <Repetition count="${sourceid@1.Colorations.Coloration@0.ColorationItem.count}"
maxSize="${__auto_parentHeight__}" direction="V" height="2.2 cm" item="aColorationItem" left="0
points" list="sourceid@1.Colorations.Coloration@0.ColorationItem" top="3 cm" width="2.2 cm">

               <!-- Condition to hide/show background color patch -->
               <Conditional condition="${${aColorationItem}.channelNo}" equals="-2" negate="Y">
                  <!-- Condition to hide disabled channels -->
                  <Conditional condition="${${aColorationItem}.method}" equals="Disable"
negate="Y">
                     <Object Id="0" left="0.2 cm" top="0.1 cm">
                        <Transformations>
                           <Text fontID="2" fontsize="9" maxwidth="85.4 points"
value="${${aColorationItem}.channelIndex}" />
                        </Transformations>
                     </Object>
                     <!-- Patch gradient, size and border -->
                     <Object Id="1" left="0.7 cm" top="0.2 cm">
                        <Transformations>
                           <Gradient Space="${${aColorationItem}.filtergradient.space}"
UseProfile="Y" Alias="${sourceid@1.Colorations.Coloration@0.ProfileInfo.Alias}"
BorderColor="${${aColorationItem}.filterRGB#808080.value}" height="34 points" percentages="100"
percentbase="100" width="34 points">${${aColorationItem}.filtergradient.value}</Gradient>
                        </Transformations>
                     </Object>
                     <!-- Color name -->
                     <Object Id="2" left="0.7 cm" top="1.5 cm">
                        <Transformations>
                           <Text alignment="left" fontID="2" fontsize="9" maxwidth="3 cm"
value="${${aColorationItem}.IData.name}" />
                        </Transformations>
                     </Object>
                  </Conditional>
                  <!-- Condition to hide channel order when is disabled -->
                  <Conditional condition="${${aColorationItem}.method}" equals="Disable"
negate="N">
                     <Object Id="0" left="0.2 cm" top="0.1 cm">
                        <Transformations>
                           <Text fontID="2" fontsize="9" maxwidth="85.4 points"
value="${${aColorationItem}.channelIndex}" />
                        </Transformations>
                     </Object>
                  </Conditional>
               </Conditional>
            </Repetition>
         </Objects>
      </Page>
   </Layout>
   <Output>
      <Space>LAB</Space>
   </Output>
</Job>
```

Related articles:

neoRipEngine v2.10

Logic in Object Conditions in Layouts

Conditional logic plays a crucial role in layouts when making decisions based on specific conditions. Sometimes, users need to show different results based on rules beyond the initial conditions. In these cases, adding empty conditions value to variables after `<Output></Output>` section does the expected processing.

```
<Variables>
    <Variable name = "SourceSOMETHING" value = ""/>
</Variables>
```

Example Scenario:

In this scenario, imagine using neoCatalog to view designs with and without trademarks. When printing with a layout using these conditions, the print document will display the trademark logo. If a trademark is not included in the design, it is showing the default logo.

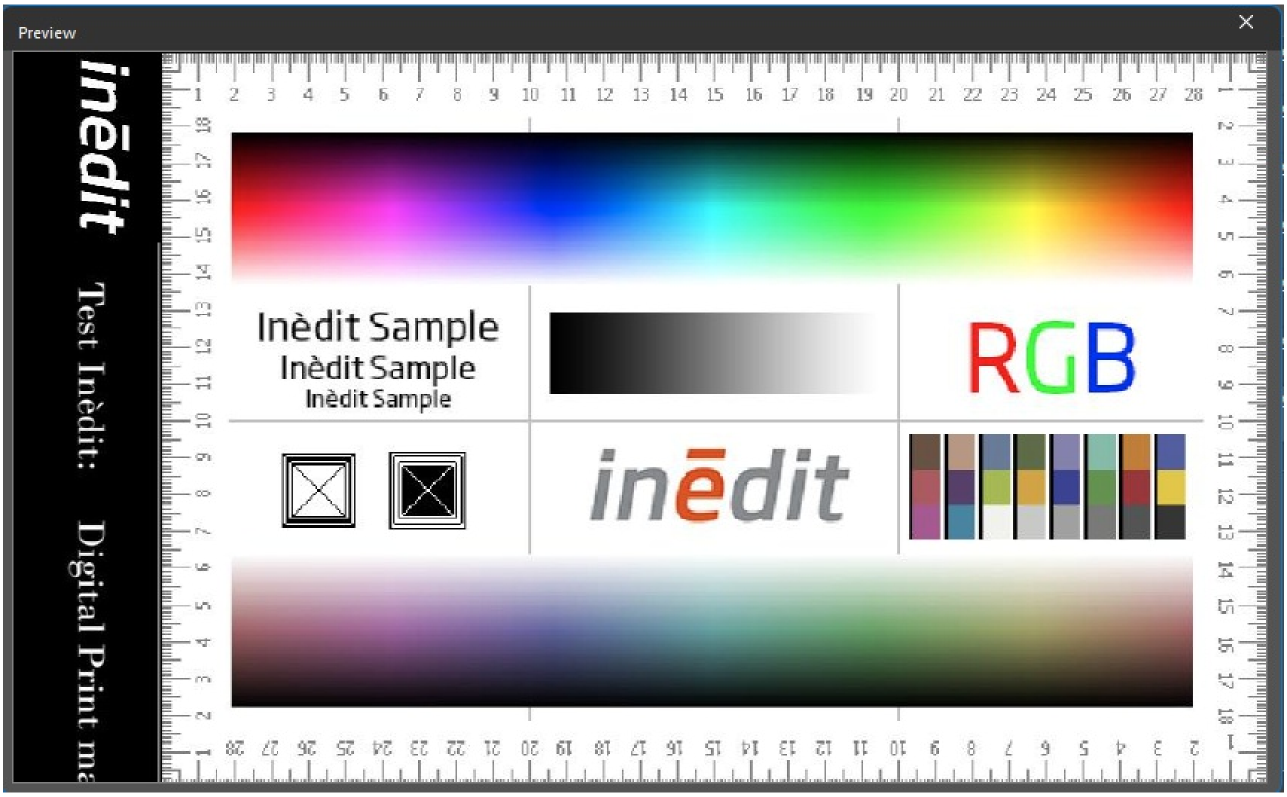
```
<?xml version="1.0" encoding="UTF-8"?>
<Job>
<Sources>
    <Source Id="0" URL="${SourceImage}"/>
    <Source Id="1" URL="${SourceXCM}"/>
    <Source Id="5" URL="Logo.tif"/>
    <Source Id = "98" URL = "${SourceTrademarkLogo}"/>
    <Source Id = "99" URL = "${SourceSubTrademarkLogo}"/>
  </Sources>
<Layout>
    <Page Id="0" width="${__auto_parentWidth__}">
    <Objects>
      <!-- Variable MUST be defined always for conditionals-->
       <Conditional condition="${SourceTrademarkLogo}" negate="Y">
        <Object Id="0" SourceID="5" left="0.2 cm" top="0.2 cm" width="3cm"
MaintainAspectRatio="true" >
        </Object>
      </Conditional>
       <Conditional condition="${SourceTrademarkLogo}">
        <Object Id="0" SourceID="98" left="0.2 cm" top="0.2 cm" width="3cm"
MaintainAspectRatio="true" >
        </Object>
      </Conditional>
            </Objects>
  </Page>

  </Layout>
<Output>
  <Space>LAB</Space>
</Output>
<Variables>
    <Variable name = "SourceTrademarkLogo" value = ""/>
</Variables>
</Job>
```

Print Server XML Layout for repeat with vertical

banner

This layout is designed for the Print Server to generate print job that includes a vertical banner in color or white depending on configuration, and is intended for designs that repeat across the layout.

TABLE OF CONTENTS



Requirement

This layout must only be applied to input designs that are vertically long enough, meaning the design height must exceed the combined vertical space of the logo and text block. If the design is shorter than this minimum requirement, it must be repeated vertically until the total height is sufficient to accommodate all layout elements without overlap.

- **Example:** If the input design height is **35 cm** and the combined height of the logo and text is **40 cm**, the design must be repeated until the total length exceeds **40 cm** — i.e., duplicated at least once.

Black Banner Example: Key Elements Explained

1. Design Area

- **Function** : Main surface where the repeating pattern is applied.
- **Position** : Begins 4 cm from the left edge, top aligned at 0 cm.
- **Size** : Uses dynamic variables like `${RepeatWidth}` and `1 rep` .
- **Properties** :
  - Maintains aspect ratio.
  - Pattern repeats (`repeat="rapport"`).
  - Auto-resizes based on layout dimensions.
  - Nearest-neighbor interpolation for scalability.
  - Includes `<RapportInfo>` to define repeat behavior.

2. Background Gradient

- **Purpose** : Vertical stripe to enhance contrast or visual separation.
- **Position** : Top-left corner of the page.
- **Rotation** : 90°, applied using the `rotation="90"` attribute.
- **Dimensions** : 40 mm wide and as tall as `${RepeatHeight}` .
- **Color** : Solid black (RGB: 0,0,0).

3. Logo Image

- **Purpose** : Displays branding or product identity.
- **Source** : `SourceID="1"` → `INEDIT_NEGATIU.tif` .
- **Position** : 10 mm from the left edge, top-aligned (0 mm).
- **Transformations** : None specified — default rendering applies.

4. Text Label

- **Purpose** : Adds informative vertical text alongside the logo.
- **Position** : 15 mm from the left edge, 60 mm from the top.
- **Width** : 2 cm.
- **Rotation** : 90° for vertical orientation.
- **Text Content** : `"Test Inèdit:     Digital Print made in Spain"`
- **Font** : Uses `FontID="2"` → `BOD_R.ttf` .

5. RapportInfo

- **Function** : Controls design repetition.
- `direction="0"` → Vertical repeat.
- `fraction_high="1"` , `fraction_low="1"` → Full-unit repeat.

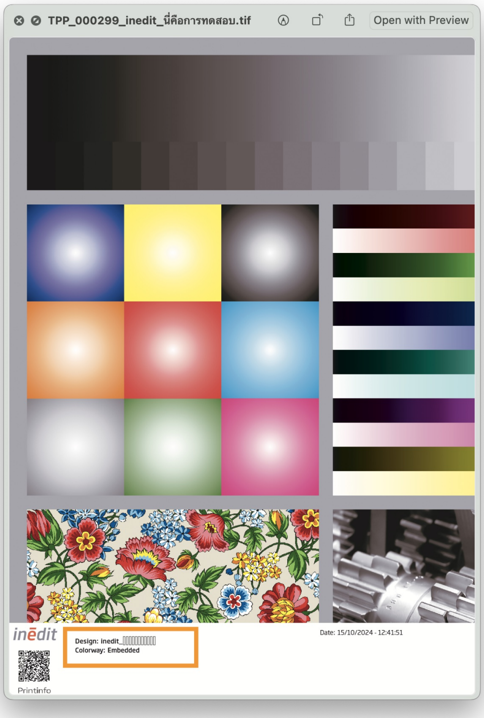
Attachments:

INEDIT_NEGATIU.tif
BOD_R.TTF
TEST NEGATIVO INEDIT.xml

Unicode fonts for Asian languages for Layouts

Problem

When using files that contain Asian characters and using some fonts in Printing Informations and Statistics, then the file is displayed in square boxes inside the Print layout.



Design: inedit_▯▯▯▯▯▯▯▯▯▯▯▯
Colorway: Embedded

Date: 15/10/2024 - 12:41:51

Reason

This issue typically involves text in Middle East or Asian languages (Arabic, Chinese, Hindi, etc.). On Windows 10 desktops, this issue typically involves text in languages other than the languages for which that system is configured.

One of the most comprehensive Unicode fonts for Windows is Microsoft's **Arial Unicode MS** . However, the size of the font is 14 megabytes, which can restrict downloading for users with slower connections.

Solution

Copy the font Arial Unicode MS inside the Layout source folder and change the font source inside the TJB of th
Layout.

Related articles:

Why the Interpolation Method matters

Problem

When in the XML the interpolation method (**interpolationMethod**) is missing in the object (**<Object/>**), this causes a loss in the performance when generating the preview and the response has a delay.

```
<Object SourceID="0" Top="0mm" Left="0mm" Rotation="270" Width="239.41mm" Height="337.9mm"
InsideWidth="1 rep" InsideHeight="1 rep" Id="0" />
```

Solution

Adding **interpolationMethod="nearest"** in the object will provide faster preview generation. The "nearest" value will use the nearest pixel quicker in order to generate the preview.

The interpolationMethod parameter is required when handling preview generations in Print Server.

In neoStampa jobs, the parameter is already applied.

```
<Object SourceID="0" Top="0mm" Left="0mm" Rotation="270" Width="239.41mm" Height="337.9mm"
interpolationMethod="nearest" InsideWidth="1 rep" InsideHeight="1 rep" Id="0" />
```

3. neoCatalog REST API

neoCatalog REST API v1.6.5

REST (REpresentational State Transferservices), which allows you to perform operations like read, modify, add, o delete Data from your data tables. REST architecture involves reading a designated web page that contains an XMI file. The XML file describes and includes the desired content. Once dynamically defined, consumers may access the interface. The response data of a REST call is in JSON, XML, or HTML format. There are many things you can with the REST API.

For example:

- A website can access data from JavaScript.
- A web server can show data on a website.
- You can upload large amounts of data.
- You can download recent data.
- You can export all of your data.

In the attached document, you'll find a full listing of all the available endpoints.

Related articles:

neoCatalog API Key

Attachments:

neoCatalog REST API-v2-20210125_090223.pdf