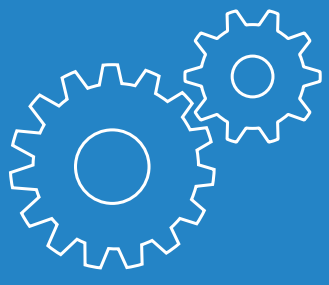
SECURE. NETWORKS.

# LANCOM Wireless ePaper Server

## Rendering Reference



LANCOM Systems

# Contents

# 1 Contents

This documentation contains information about the image rendering process used by the LANCOM Wireless ePaper Server.

It is divided into the following sections:

- **General information** – Introduction and general information about the image rendering process.
- **Elements** – Describes all elements and attributes that are used to define a template.
- **Utils** – Documentation about all Util methods that can be used inside a template to modify content.
- **Examples** – Several examples using different templates to generate an image.

# 2 General

The graphics rendering process is based on *XSL (Extensible Stylesheet Language)* templates. The template is used to specify the fields that are rendered into the image. The individual image for the display is then generated by applying the template to an XML. This is based on the properties of the display and is extended by a custom XML from the user.

The following steps are executed when processing a template:

1. Generate an XML-based record on the display.
2. Apply the template (XSL) for the generated XML record. This results in a document containing the fields, which are specified in this reference and which reference values from the XML.
3. Render the image based on the previous output.

### XSL template

The XSL template is used in the XML record. After applying the template, a source for generating the image is created. The structure of a template should look like this (example):

**Template Example**

```
     <?xml version="1.0" encoding="UTF-8"?>

     <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="1.0">

     <xsl:template match="Record">

     Space for field definitions and XML record data referencing


     </xsl:template>

     </xsl:stylesheet>
```

### XML record

An XML record is automatically generated before an image is rendered. The XML has the following structure:

**Record Example**

```
     <?xml version="1.0" encoding="UTF-8"?>

     <Record>

      <Label>

       <Id>B1000000</Id>

       <LabelType>iMAGOTAG 2.7</LabelType>

       <DisplayHeight>176</DisplayHeight>

       <DisplayWidth>264</DisplayWidth>

      </Label>

      <Task>

       <CurrentDate>11.09.2012</CurrentDate>

       <CurrentTime>15:13:53</CurrentTime>
```

```
      <ExternalId>0</ExternalId>
      <Page>0</Page>
     </Task>
    </Record>
```

The XML specified by the user is appended after the "Task" element.

This means that this XML can be assigned to any field in the XSL template, e.g.

<xsl:value select="Label/Id"/>

assigns the ID of the display.

### Utils

The Utils template was created to manipulate text content for the generated image. For more information about the different supported methods, their parameters and outputs, see the section *Utils*.

### CSS support

The templates support the import of one or more CSS files. This specifies the attributes of the different fields. Every tag, except for the root image tag, supports the HTML attributes such as "ID" and "class" to reference from the CSS file.

The LANCOM Wireless ePaper Server supports the definition of CSS1. The attributes used in the CSS file have the same identifier as the attributes in the template (e.g. in template: font-size="12", in CSS: font-size: 12;).

In order to import a CSS file, a <css> tag and the "href" attribute have to be defined in the root image tag. The "href" attribute specifies the relative path to the CSS file from the template file.

For the application, see *2.7"* on page 68 in the section "CSS example".

# 3 Elements

This chapter describes all elements and attributes that are used to define a template.

## 3.1 The element <image>

**Description:**

The root element of the image template. It specifies the basic attributes (e.g. size, background) of the image to be generated as well as default values for font and color.

**Attributes**

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **colors** | The color of the element or of the contained text. | false | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | false | *2.7"* on page 68 |
| **invert** | Inverts the color of the image. | false | true \| false | false | - |
| **rotation** | Sets the rotation of the output image. The image can only be rotated by 0, 90, 180 and 270 degrees. | false | 0\|90\|180\|270 | 0 | - |
| **background-image** | Specifies the background image of an element. The path to the image should be specified relative to the template. The image is too small, it is placed in the upper-left corner. | false | A path relative to the template file or to a URL (e.g. "../test/image.png or url.to/image.png"). | No default value | - |
| **width** | The width of the element, specified in pixels. | false | The size, value specified in pixels. | No default value | - |
| **height** | The height of the element, specified in pixels. | false | The size, value specified in pixels. | No default value | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **background-color** | The background color of the element. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | Transparent | - |
| **font-family** | The name of the font that is to be selected. | true | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Arial | - |
| **font-size** | Sets a range of font sizes, separated by colon. The text is rendered with the largest possible (and fixed) font size. | true | A positive integer specifies the font size in pt, or several integers, separated by a colon as alternate font sizes. | 12 | - |
| **font-type** | Specifies whether a text should be printed italic or normal. | true | normal \| italic | normal | - |
| **font-weight** | Specifies whether a text should be printed bold or normal. | true | normal \| bold | normal | - |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **condense** | If the text is too long, it is automatically compressed using the optimal compression ratio. | true | One or more floating point number(s) separated by a colon. | 1.0 | *4.4" on page 102* |
| **line-spacing** | Sets the spacing between text lines in pixels (negative values decrease the line spacing). | true | A positive or negative integer value. | 0 | *2.7" on page 68* |
| **clip-height** | The height of the clipping area (display area). | false | The size, value specified in pixels. | No default value | - |
| **clip-width** | The width of the clipping area (display area). | false | The size, value specified in pixels. | No default value | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **clip-x** | The x-coordinates for the clipping area. | false | An integer >= 0 and less than the specified image size. | No default value | - |
| **clip-y** | The y-coordinates for the clipping area. | false | An integer >= 0 and less than the specified image size. | No default value | - |
| **wrap** | Changes the line break in a text field.<br>**Normal:**<br>Line break at spaces and hyphens.<br>**None:**<br>No line breaks. | true | normal \| none | normal | - |
| **dithering** | Enables or disables dithering for the generated image. | false | true \| false | false | - |

### Child elements

| | |
|---|---|
| <field> | See the section *The element <field>* |
| <rect> | See the section *The element <rect>* |
| <line> | See the section *The element <line>* |
| <triangle> | See the section *The element <triangle>* |
| <ellipse> | See the section *The element <ellipse>* |
| <polygon> | See the section *The element <polygon>* |

## 3.2 The element <field>

### Description:

The element `field` specifies a rectangular container, which an image, text or barcode can be positioned in. The position of the field can be specified either with the x- and y-coordinates of the upper-left corner or by specifying the x-coordinate only. In the latter case, the field is positioned exactly after the last field for which the y-coordinate was specified (or 0 in the event that the y-coordinate was not set for any field).

### Attributes

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **rotation** | A positive or negative angle of rotation of the corresponding field contents. | false | An integer >= 0 and less than the specified image size. | 0 | *4.4"* on page 102 |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |
| **x** | The position of the element on the x-axis, in pixels. | false | An integer >= 0 and less than the specified image size. | No default value | - |
| **y** | The position of the element on the y-axis, in pixels. | false | An integer >= 0 and less than the specified image size. | No default value | - |
| **width** | The width of the element, specified in pixels. | true | The size, value specified in pixels. | No default value | - |
| **height** | The height of the element, specified in pixels. | true | The size, value specified in pixels. | No default value | - |
| **max-height** | Sets the maximum height of the field. This attribute is used in combination with a relative y-position. The height of the rendered field is smaller than or equal to the specified maximum height value, depending on its content. | false | A positive integer value (> 0) | No default value | *2.7"* on page 68 |
| **background-color** | The background color of the element. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | Transparent | - |
| **border** | Specifies the thickness of the border around the element. A value of | false | A positive integer value (> 0) | 0 | *4.4"* on page 102 |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| | 0 means there is no border. | | | | |
| **border-color** | Sets the color of the selected border. | false | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **border-style** | Sets the line type of the border. | false | dotted \| dashed \| solid | solid | - |
| **align** | Sets the horizontal alignment of the element/text. | true | left \| right \| center \| justify | Left-aligned | - |
| **valign** | Sets the vertical alignment of the element/text. | true | top \| center \| bottom | top | - |
| **rounded-corners** | Specifies which corners should be rounded. Several corners can be selected, as separated by a comma (,). | false | top_left \| top_right \| bottom_left \| bottom_right | No default value | *7.4"* on page 116 |
| **corner-radius** | Radius of the rounded corners of the rectangle/field. A radius of value zero means there are no rounded corners. The radius of the corners is automatically adjusted if it is too large. The maximum radius is equal to (min(width,height) - 1) / 2. | false | The size, value specified in pixels. | 0 | *7.4"* on page 116 |
| **wrap** | Changes the line break in a text field.<br>**Normal:**<br>Line break at spaces and hyphens.<br>**None:**<br>No line breaks. | true | normal \| none | normal | - |

### Child elements

| | |
|---|---|
| <field> | See section *The element <field>* on page 8 |

| | |
|---|---|
| <table> | See the section *The element <table>* |
| <barcode> | See the section *The element <barcode>* |
| <img> | See the section *The element <img>* |
| <label> | See the section *The element <label>* |
| <text> | See the section *The element <text>* |
| <rect> | See the section *The element <rect>* |
| <line> | See the section *The element <line>* |
| <triangle> | See the section *The element <triangle>* |
| <ellipse> | See the section *The element <ellipse>* |
| <polygon> | See the section *The element <polygon>* |

**Examples**

- *2.7"* example of nested fields

## 3.3 The element <table>

**Description:**

The element `table` marks the beginning of a table, followed by table rows and cells. By default the table takes the width from the field it is placed in. The height of the table depends on the height of the rows inside it.

**Attributes**

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |
| **font-family** | The name of the font that is to be selected. | true | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Arial | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **align** | Sets the horizontal alignment of the element/text. | true | left \| right \| center \| justify | Left-aligned | - |
| **padding-bottom** | Specifies the padding in pixels from the bottom. | true | A positive or negative integer value. | 0 | - |
| **padding-left** | Specifies the padding in pixels from the left. | true | A positive or negative integer value. | 0 | - |
| **padding-right** | Specifies the padding in pixels from the right. | true | A positive or negative integer value. | 0 | - |
| **padding-top** | Specifies the padding in pixels from the top. | true | A positive or negative integer value. | 0 | - |
| **background-color** | The background color of the element. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | Transparent | - |
| **text-decoration** | Defines the text decoration for a text (underline, overline, linethrough, or none). | true | underline \| linethrough \| none | None | |
| **font-size** | Sets a range of font sizes, separated by colon. The text is rendered with the largest possible (and fixed) font size. | true | A positive integer specifies the font size in pt, or several integers, separated by a colon as alternate font sizes. | 12 | - |
| **font-type** | Specifies whether a text should be printed italic or normal. | true | normal \| italic | normal | - |
| **font-weight** | Specifies whether a text should be printed bold or normal. | true | normal \| bold | normal | - |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **condense** | If the text is too long, it is automatically compressed using the optimal compression ratio. | true | One or more floating point number(s) separated by a colon. | 1.0 | |
| **line-spacing** | Sets the spacing between text lines in pixels (negative values decrease the line spacing). | true | A positive or negative integer value. | 0 | |
| **wrap** | Changes the line break in a text field.<br>**Normal:**<br>Line break at spaces and hyphens.<br>**None:**<br>No line breaks. | true | normal \| none | normal | - |
| **letter-spacing** | The letter-spacing attribute increases or decreases the spacing between characters within a text. | true | A positive or negative floating point number. | 0.0 | |
| **superscript** | Used to produce superscript or subscript text. | true | superscript \| subscript \| normal | normal | |
| **outline** | Specifies whether the text should be outlined or not. | true | true \| false | false | - |
| **outline-color** | Specifies the color of the text outline. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **outline-thickness** | Specifies the thickness of the text outline. | true | A positive integer value (> 0) | 1 | - |
| **outline-inside** | This value specifies whether the outline should be drawn inside or outside of the text. | true | true \| false | false | - |
| **shadow** | Enables/disables the shadow | true | true \| false | false | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **shadow-alpha** | The alpha value (transparency) of the shadow. | true | A positive floating point number >= 0 and <= 1 | 0.5 | - |
| **shadow-color** | Sets the color of the shadow. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **shadow-size** | The size of the shadow in pixels. | true | A positive integer value (> 0) | 2 | - |
| **shadow-blur** | Specifies whether a shadow is rendered with or without a blur effect. | true | true \| false | false | - |
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |
| **cellpadding** | Specifies the spacing around the text content of a table cell, in pixels. | true | A positive integer value (>=0) | 2 | - |
| **valign** | Sets the vertical alignment of the element/text. | true | top \| center \| bottom | top | - |
| **width** | The width of the element, specified in pixels. | true | The size, value specified in pixels. | No default value | - |
| **height** | The height of the element, specified in pixels. | true | The size, value specified in pixels. | No default value | - |
| **max-height** | Sets the maximum height of the field. This attribute is used in combination with a relative y-position. The height of the | false | A positive integer value (> 0) | No default value | , *4.4"* on page 102*2.7"* on page 68 |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| | rendered field is smaller than or equal to the specified maximum height value, depending on its content. | | | | |
| **table-border** | Sets the border width of the whole table with the exception of the header row or column. The width of the header row/column is set through the attribute 'header-border'. A border of value zero means there are no border. | false | A positive integer value (> 0) | 1 | |
| **header-border** | Determines the border width of the header row and column. This attribute only takes effect if 'header-row' and/or 'header-column' is disabled. | false | A positive integer value (> 0) | 2 | |
| **header-column** | The header column is the first column of a table. The border width of this column can be set with 'header-border', if enabled. | false | true \| false | false | |
| **header-row** | The header row is the first line of a table. The border width of this row can be set with 'header-border', if enabled. | false | true \| false | false | |
| **border-color** | Sets the color of the selected border. | false | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **background-color** | The background color of the element. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | Transparent | - |

### Child elements

<row> See the section *The element <row>*

### Examples

- *4.4"* Example table

## 3.4 The element <row> (line)

### Description:

The element `row` (line) indicates a new row inside a table. The height and width can be specified. No height and width is set, the maximum width and height is used as required by the contents of the row.

### Attributes

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |
| **font-family** | The name of the font that is to be selected. | true | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Arial | - |
| **align** | Sets the horizontal alignment of the element/text. | true | left \| right \| center \| justify | Left-aligned | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **padding-bottom** | Specifies the padding in pixels from the bottom. | true | A positive or negative integer value. | 0 | - |
| **padding-left** | Specifies the padding in pixels from the left. | true | A positive or negative integer value. | 0 | - |
| **padding-right** | Specifies the padding in pixels from the right. | true | A positive or negative integer value. | 0 | - |
| **padding-top** | Specifies the padding in pixels from the top. | true | A positive or negative integer value. | 0 | - |
| **background-color** | The background color of the element. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | Transparent | - |
| **text-decoration** | Defines the text decoration for a text (underline, overline, linethrough, or none). | true | underline \| linethrough \| none | None | |
| **font-size** | Sets a range of font sizes, separated by colon. The text is rendered with the largest possible (and fixed) font size. | true | A positive integer specifies the font size in pt, or several integers, separated by a colon as alternate font sizes. | 12 | - |
| **font-type** | Specifies whether a text should be printed italic or normal. | true | normal \| italic | normal | - |
| **font-weight** | Specifies whether a text should be printed bold or normal. | true | normal \| bold | normal | - |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **condense** | If the text is too long, it is automatically compressed using | true | One or more floating point number(s) separated by a colon. | 1.0 | |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| | the optimal compression ratio. | | | | |
| **line-spacing** | Sets the spacing between text lines in pixels (negative values decrease the line spacing). | true | A positive or negative integer value. | 0 | |
| **wrap** | Changes the line break in a text field.<br>**Normal:**<br>Line break at spaces and hyphens.<br>**None:**<br>No line breaks. | true | normal \| none | normal | - |
| **letter-spacing** | The letter-spacing attribute increases or decreases the spacing between characters within a text. | true | A positive or negative floating point number. | 0.0 | |
| **superscript** | Used to produce superscript or subscript text. | true | superscript \| subscript \| normal | normal | |
| **outline** | Specifies whether the text should be outlined or not. | true | true \| false | false | - |
| **outline-color** | Specifies the color of the text outline. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **outline-thickness** | Specifies the thickness of the text outline. | true | A positive integer value (> 0) | 1 | - |
| **outline-inside** | This value specifies whether the outline should be drawn inside or outside of the text. | true | true \| false | false | - |
| **shadow** | Enables/disables the shadow | true | true \| false | false | - |
| **shadow-alpha** | The alpha value (transparency) of the shadow. | true | A positive floating point number >= 0 and <= 1 | 0.5 | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **shadow-color** | Sets the color of the shadow. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **shadow-size** | The size of the shadow in pixels. | true | A positive integer value (> 0) | 2 | - |
| **shadow-blur** | Specifies whether a shadow is rendered with or without a blur effect. | true | true \| false | false | - |
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |
| **table-height** | Sets the height of the current table element. | false | A positive integer value (>=0) | No default value | |
| **valign** | Sets the vertical alignment of the element/text. | true | top \| center \| bottom | top | - |
| **cellpadding** | Specifies the spacing around the text content of a table cell, in pixels. | true | A positive integer value (>=0) | 2 | - |

### Child elements

<cell> See the section *The element <cell>*

### Examples

- *4.4"* Example table

# 3.5 The element <cell>

**Description:**

The element `cell` stands for a table cell inside a table row. It is possible to set a fixed width. If a fixed width is set, the column size of the whole table will be set to this value. If multiple cells specify a width of this column, the maximum value is applied. If no values are set, the width of the cell is evenly distributed.

**Attributes**

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4" on page 102 Text* |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4" on page 102 Text* |
| **font-family** | The name of the font that is to be selected. | true | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Arial | - |
| **align** | Sets the horizontal alignment of the element/text. | true | left \| right \| center \| justify | Left-aligned | - |
| **padding-bottom** | Specifies the padding in pixels from the bottom. | true | A positive or negative integer value. | 0 | - |
| **padding-left** | Specifies the padding in pixels from the left. | true | A positive or negative integer value. | 0 | - |
| **padding-right** | Specifies the padding in pixels from the right. | true | A positive or negative integer value. | 0 | - |
| **padding-top** | Specifies the padding in pixels from the top. | true | A positive or negative integer value. | 0 | - |
| **background-color** | The background color of the element. | true | black \| white \| transparent \| semicolon-separated | Transparent | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| | | | RGBA string with values between 0 and 255 | | |
| **text-decoration** | Defines the text decoration for a text (underline, overline, linethrough, or none). | true | underline \| linethrough \| none | None | |
| **font-size** | Sets a range of font sizes, separated by colon. The text is rendered with the largest possible (and fixed) font size. | true | A positive integer specifies the font size in pt, or several integers, separated by a colon as alternate font sizes. | 12 | - |
| **font-type** | Specifies whether a text should be printed italic or normal. | true | normal \| italic | normal | - |
| **font-weight** | Specifies whether a text should be printed bold or normal. | true | normal \| bold | normal | - |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **condense** | If the text is too long, it is automatically compressed using the optimal compression ratio. | true | One or more floating point number(s) separated by a colon. | 1.0 | |
| **line-spacing** | Sets the spacing between text lines in pixels (negative values decrease the line spacing). | true | A positive or negative integer value. | 0 | |
| **wrap** | Changes the line break in a text field.<br>**Normal:**<br>Line break at spaces and hyphens.<br>**None:**<br>No line breaks. | true | normal \| none | normal | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **letter-spacing** | The letter-spacing attribute increases or decreases the spacing between characters within a text. | true | A positive or negative floating point number. | 0.0 | |
| **superscript** | Used to produce superscript or subscript text. | true | superscript \| subscript \| normal | normal | |
| **outline** | Specifies whether the text should be outlined or not. | true | true \| false | false | - |
| **outline-color** | Specifies the color of the text outline. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **outline-thickness** | Specifies the thickness of the text outline. | true | A positive integer value (> 0) | 1 | - |
| **outline-inside** | This value specifies whether the outline should be drawn inside or outside of the text. | true | true \| false | false | - |
| **shadow** | Enables/disables the shadow | true | true \| false | false | - |
| **shadow-alpha** | The alpha value (transparency) of the shadow. | true | A positive floating point number >= 0 and <= 1 | 0.5 | - |
| **shadow-color** | Sets the color of the shadow. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **shadow-size** | The size of the shadow in pixels. | true | A positive integer value (> 0) | 2 | - |
| **shadow-blur** | Specifies whether a shadow is rendered with or without a blur effect. | true | true \| false | false | - |
| **table-width** | The width of the clipping area (display area). | false | A positive integer value (>=0) | No default value | |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **cellpadding** | Specifies the spacing around the text content of a table cell, in pixels. | true | A positive integer value (>=0) | 2 | - |
| **valign** | Sets the vertical alignment of the element/text. | true | top \| center \| bottom | top | - |

### Child elements

<label> See the section *The element <label>*

<text> See the section *The element <text>*

### Examples

- *4.4"* Example table

## 3.6 The element <img>

### Description:

This tag is used to insert an image in front of a background. The image is sourced either by means of a relative path from the templates to an image file, or via a Base64-encoded string of an image. Be careful when using a path for an image: Any processing of the template on the server may result in the path changing, or it may be that the image no longer exists on the server.

### Attributes

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102*Text* |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102*Text* |
| **src** | Specifies the path or the URL to an image. The path must be relative to the template file. | false | A path relative to the template file or to a URL (e.g. "../test/image.png or url.to/image.png") | No default value | *4.4"* on page 102 |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **data** | Sets a Base64-encoded string as the image source. | false | A Base64-encoded string | No default value | *4.4"* on page 102 |
| **autoscale** | Enables/disables auto-scaling of the element. The element is scaled to the size of the field. | false | true \| false | false | - |
| **cut-bottom** | Sets the number of pixels to be cut off of this edge of the original image. | false | A positive integer value (>=0) | 0 | - |
| **cut-left** | Sets the number of pixels to be cut off of this edge of the original image. | false | A positive integer value (>=0) | 0 | - |
| **cut-right** | Sets the number of pixels to be cut off of this edge of the original image. | false | A positive integer value (>=0) | 0 | - |
| **cut-top** | Sets the number of pixels to be cut off of this edge of the original image. | false | A positive integer value (>=0) | 0 | - |

### Child elements

No child elements

### Examples

- *4.4"* Example image

## 3.7 The element <barcode>

### Description:

The `barcode` element sets the properties for outputting a barcode on the image.

### Properties

| Name | Description | Inherited | Possible values | Default value | Example: |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The | false | String | No default value | *4.4"* on page 102*Text* |

| Name | Description | Inherited | Possible values | Default value | Example: |
|---|---|---|---|---|---|
| | same class identifiers should be be used in the case of multiple tags. | | | | |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102*Text* |
| **type** | The type of barcode to be generated. | false | codabar \| code128 \| code39 \| ean13 \| ean8 \| ean-128 \| upc-a \| upc-e \| intl2of5 \| itf-14 \| postnet \| royal-mail-cbc \| usps4cb \| pdf417 \| datamatrix \| qr | No default value | - |

### Barcode types

#### Table 1: Codabar

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 10 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **humanReadablePlacement** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **displayStartStop** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |
| **displayStartStop** | Enables/disables the display of start and stop bars for Codabar and Code-39 barcodes. | true \| false | false |

**Table 2: Code 128**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 10 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **humanReadableEnabled** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |
| **codesets** | Sets the character set for Code-128 barcodes. | A string combination for character sets A, B, and C. | abc |

**Table 3: Code39**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 10 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **humanReadableEnabled** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |
| **displayStartStop** | Enables/disables the display of start and stop bars for Codabar and Code-39 barcodes. | true \| false | false |
| **displayChecksum** | Show/hide the checksum | true \| false | false |
| **checksumMode** | Specifies how to handle the checksum. | add \| check \| ignore \| auto | Ignore |
| **intercharGapWidth** | Sets the width of the quiet zone of the Code-39 barcode. | A positive integer value (> 0) | 1 |
| **extendedCharset** | Enables/disables the extended character set for Code-39 barcodes. | true \| false | false |

**Table 4: DataMatrix**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 1 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | true |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **shape** | Sets the shape of the data-matrix barcode. | None\|Rectangle\|Shape | None |

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **padded** | Enables/disables padding | true \| false | false |
| **minSymbolSizeHeight** | Defines the minimum height of the symbol. | A positive integer value (>=0) | 1 |
| **minSymbolSizeWidth** | Defines the minimum width of the symbol. | A positive integer value (>=0) | 1 |
| **maxSymbolSizeHeight** | Defines the maximum height of the symbol. | A positive integer value (>=0) | 1 |
| **maxSymbolSizeWidth** | Defines the maximum width of the symbol. | A positive integer value (>=0) | 0 |

**Table 5: EAN-128**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 10 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **humanReadableEnabled** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |
| **template** | Specifies an optional template for the EAN-128 barcode. | String | No default value |
| **omitBrackets** | Enables/disables the omission of brackets for the EAN-128 barcode. | true \| false | false |

**Table 6: EAN-13**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 10 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **humanReadableEnabled** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |
| **checksumMode** | Specifies how to handle the checksum. | add \| check \| ignore \| auto | Automatic |

**Table 7: ean8**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 10 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **humanReadableEnabled** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |
| **checksumMode** | Specifies how to handle the checksum. | add \| check \| ignore \| auto | Automatic |

**Table 8: intl2of5**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 10 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **humanReadableEnabled** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |
| **wideFactor** | Specifies the factor for calculating the gaps between the barcode stripes. | A positive floating point number (>0) | 3.0 |
| **checksumMode** | Specifies how to handle the checksum. | add \| check \| ignore \| auto | Ignore |
| **displayChecksum** | Show/hide the checksum | true \| false | false |

**Table 9: ITF-14**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 10 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **humanReadableEnabled** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |
| **bearerBarWidth** | Specifies the width (in pixels) of the border surrounding the barcode for ITF-14 barcodes. | A positive integer value (>=0) | 20 |
| **bearerBox** | Specifies the width (in pixels) of the border surrounding the barcode for ITF-14 barcodes. | A positive integer value (>=0) | true |
| **wideFactor** | Specifies the factor for calculating the gaps between the barcode stripes. | A positive floating point number (>0) | 2.0 |
| **checksumMode** | Specifies how to handle the checksum. | add \| check \| ignore \| auto | Automatic |
| **displayChecksum** | Show/hide the checksum | true \| false | false |

**Table 10: PDF417**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **humanReadableEnabled** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |
| **rowHeightRatio** | Defines the height ratio of the lines of the PDF417 barcode. | A positive floating point number (>0) | 3.0 |
| **columns** | Specifies the number of columns. Can be disabled by using a zero value. In this case, it is detected automatically. | A positive integer value (> 0) | 0 |
| **minColumns** | Specifies the minimum number of columns. | A positive integer value (> 0) | 2 |
| **minColumns** | Specifies the maximum number of columns. | A positive integer value (> 0) | 2 |
| **ecLevel** | Specifies the minimum number of rows. | A positive integer value (> 0) | 0 |
| **widthToHeightRatio** | Specifies the ratio between the width and height. | A positive floating point number (>0) | 3.0 |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 2 |
| **verticalQuietZoneFactor** | Factor for calculating the vertical quiet zone. | A positive integer value (> 0) | 6 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | true |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **padded** | Enables/disables padding | true \| false | false |
| **minRows** | Specifies the minimum number of rows. | A positive integer value (> 0) | 3 |
| **maxRows** | Specifies the minimum number of rows. | A positive integer value (> 0) | 90 |
| **autoMode** | Enables/disables the auto mode of the PDF417 barcodes. | true \| false | true |

**Table 11: PostNet**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 10 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **humanReadableEnabled** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |

**Table 12: QR Code**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **padded** | Enables/disables padding | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 2 |
| **errorCorrectionLevel** | Defines the error correction level of a QR code. | l \| m \| q \| h | h |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |

**Table 13: Royal Mail CBC**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 10 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **humanReadableEnabled** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |

**Table 14: UPC-A**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 10 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **humanReadableEnabled** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **checksumMode** | Specifies how to handle the checksum. | add \| check \| ignore \| auto | Automatic |

**Table 15: UPC-E**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 10 |
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **humanReadableEnabled** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |
| **checksumMode** | Specifies how to handle the checksum. | add \| check \| ignore \| auto | Automatic |

**Table 16: USPS4CB**

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **fontSizePt** | Sets a range of font sizes, separated by semicolon. The text is rendered with the largest possible (and fixed) font size. | A positive integer specifies the font size in pt, or several integers, separated by a semicolon as alternate font sizes. | 6 |
| **fontName** | Specifies the font of the text content of the barcode. | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Helvetica |
| **quietZoneFactor** | Factor for calculating the quiet zone of a barcode. | A positive integer value (> 0) | 10 |

| Name | Description | Possible values | Default value |
|---|---|---|---|
| **quietZoneEnabled** | In case a quiet zone should be enabled/disabled. | true \| false | false |
| **scale** | Specification of a scale for the barcode itself. | A positive integer value (>=0) | 1 |
| **humanReadableEnabled** | Enables or disables the display of the text content of the barcode. | true \| false | true |
| **autoscale** | Enables/disables auto-scaling of objects. The object is scaled to the size of the field. | true \| false | false |
| **humanReadablePlacement** | Specifies where to display the contents of the text in relation to the barcode. | bottom \| top \| none | below |

### Child elements

<text> See the section *The element <text>*

### Examples

- *4.4"* example barcode

## 3.8 The element <rect>

### Description:

Defines a rectangle to be drawn on an image. The rectangle is drawn behind the field elements.

### Attributes

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **x** | The position of the element on the x-axis, in pixels. | false | An integer >= 0 and less than the specified image size. | No default value | - |
| **y** | The position of the element on the y-axis, in pixels. | false | An integer >= 0 and less than the specified image size. | No default value | - |
| **width** | The width of the element, specified in pixels. | false | The size, value specified in pixels. | No default value | - |
| **height** | The height of the element, specified in pixels. | false | The size, value specified in pixels. | No default value | - |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **border** | Specifies the thickness of the border around the element. A value of 0 means there is no border. | false | A positive integer value (> 0) | 0 | *4.4"* on page 102 |
| **border-color** | Sets the color of the selected border. | false | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **border-style** | Sets the line type of the border. | false | dotted \| dashed \| solid | solid | - |
| **fill** | Specifies whether the element is to be filled in with the selected color. | false | true \| false | true | - |
| **rounded-corners** | Specifies which corners should be rounded. Several corners can be selected, as separated by a comma (,). | false | top_left \| top_right \| bottom_left \| bottom_right | No default value | *7.4"* on page 116 |
| **corner-radius** | Radius of the rounded corners of the rectangle/field. A radius of value zero means there are | false | The size, value specified in pixels. | 0 | *7.4"* on page 116 |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| | no rounded corners. The radius of the corners is automatically adjusted if it is too large. The maximum radius is equal to (min(width,height) - 1) / 2. | | | | |

### Child elements

No child elements

### Examples

- *4.4"* Drawing example
- *7.4"* Advanced drawing example

## 3.9 The element <line>

### Description:

Specifies the properties of a line that is to be inserted into an image. The line is drawn behind the field elements.

### Attributes

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |
| **x-from** | Sets the starting position on the x-axis, in pixels. | false | An integer >= 0 and less than the specified image size. | No default value | *4.4"* on page 102 |
| **y-from** | Sets the starting position on the y-axis, in pixels. | false | An integer >= 0 and less than the specified image size. | No default value | *4.4"* on page 102 |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **x-to** | Sets the end position on the x-axis, in pixels. | false | An integer >= 0 and less than the specified image size. | No default value | *4.4"* on page 102 |
| **y-to** | Sets the end position on the y-axis, in pixels. | false | An integer >= 0 and less than the specified image size. | No default value | *4.4"* on page 102 |
| **thickness** | Determines the thickness, in pixels. | false | A positive integer value (>=0) | 1 | - |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |

### Child elements

No child elements

### Examples

- *4.4"* Drawing example
- *7.4"* Advanced drawing example

## 3.10 The element <triangle>

### Description:

Defines a triangle to be drawn on an image. The element `triangle` is drawn behind the field elements.

### Attributes

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **x1** | Coordinate to position a corner of a triangle. | false | An integer >= 0 and less than the specified image size. | No default value | *7.4" on page 116* |
| **x2** | Coordinate to position a corner of a triangle. | false | An integer >= 0 and less than the specified image size. | No default value | *7.4" on page 116* |
| **x3** | Coordinate to position a corner of a triangle. | false | An integer >= 0 and less than the specified image size. | No default value | *7.4" on page 116* |
| **y1** | Coordinate to position a corner of a triangle. | false | An integer >= 0 and less than the specified image size. | No default value | *7.4" on page 116* |
| **y2** | Coordinate to position a corner of a triangle. | false | An integer >= 0 and less than the specified image size. | No default value | *7.4" on page 116* |
| **y3** | Coordinate to position a corner of a triangle. | false | An integer >= 0 and less than the specified image size. | No default value | *7.4" on page 116* |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **border** | Specifies the thickness of the border around the element. A value of 0 means there is no border. | false | A positive integer value (> 0) | 0 | *4.4" on page 102* |
| **border-color** | Sets the color of the selected border. | false | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **border-style** | Sets the line type of the border. | false | dotted \| dashed \| solid | solid | - |
| **fill** | Specifies whether the element is to be filled in with the selected color. | false | true \| false | true | - |

### Child elements

No child elements

**Examples**

- *4.4"* Drawing example
- *7.4"* Advanced drawing example

## 3.11 The element <ellipse>

**Description:**

Defines an ellipse to be drawn on an image. The ellipse is placed behind the field element.

**Attributes**

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |
| **x** | The position of the element on the x-axis, in pixels. | false | An integer >= 0 and less than the specified image size. | No default value | - |
| **y** | The position of the element on the y-axis, in pixels. | false | An integer >= 0 and less than the specified image size. | No default value | - |
| **width** | The width of the element, specified in pixels. | false | The size, value specified in pixels. | No default value | - |
| **height** | The height of the element, specified in pixels. | false | The size, value specified in pixels. | No default value | - |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **border** | Specifies the thickness of the border around the element. A value of 0 means there is no border. | false | A positive integer value (> 0) | 0 | *4.4"* on page 102 |
| **border-color** | Sets the color of the selected border. | false | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **border-style** | Sets the line type of the border. | false | dotted \| dashed \| solid | solid | - |
| **fill** | Specifies whether the element is to be filled in with the selected color. | false | true \| false | true | - |

### Child elements

No child elements

### Examples

- *4.4"* Drawing example
- *7.4"* Advanced drawing example

## 3.12 The element <polygon>

### Description:

Defines a polygon to be drawn on an image. The polygon is drawn behind the field elements.

### Attributes

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced | false | String | No default value | *4.4"* on page 102 |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| | using a CSS file. This identifier must be unique. | | | | |
| **points** | A whitespace-separated list of coordinates to be used to draw a polygon. | false | A whitespace-separated list of coordinates. Coordinates are separated by a colon, e.g. 12,12,221,152 | No default value | *7.4"* on page 116 |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **border** | Specifies the thickness of the border around the element. A value of 0 means there is no border. | false | A positive integer value (> 0) | 0 | *4.4"* on page 102 |
| **border-color** | Sets the color of the selected border. | false | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **border-style** | Sets the line type of the border. | false | dotted \| dashed \| solid | solid | - |
| **fill** | Specifies whether the element is to be filled in with the selected color. | false | true \| false | true | - |

### Child elements

No child elements

### Examples

- *4.4"* Drawing example
- *7.4"* Advanced drawing example

# 3.13 The element <text>

**Description:**

The element `text` specifies the properties of the contained text. The field element can contain several text elements. The font properties of the text element are inherited from the root image tag. If several text elements should fit inside a field, the y-position attribute must be omitted. In this case, a text is positioned below the others.

Be careful with the auto-spacing attribute. If you use dynamic positioning, the first text element takes up the entire space of the field.

**Attributes**

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |
| **font-family** | The name of the font that is to be selected. | true | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Arial | - |
| **align** | Sets the horizontal alignment of the element/text. | true | left \| right \| center \| justify | Left-aligned | - |
| **padding-bottom** | Specifies the padding in pixels from the bottom. | true | A positive or negative integer value. | 0 | - |
| **padding-left** | Specifies the padding in pixels from the left. | true | A positive or negative integer value. | 0 | - |
| **padding-right** | Specifies the padding in pixels from the right. | true | A positive or negative integer value. | 0 | - |
| **padding-top** | Specifies the padding in pixels from the top. | true | A positive or negative integer value. | 0 | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **background-color** | The background color of the element. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | Transparent | - |
| **text-decoration** | Defines the text decoration for a text (underline, overline, linethrough, or none). | true | underline \| linethrough \| none | None | |
| **font-size** | Sets a range of font sizes, separated by colon. The text is rendered with the largest possible (and fixed) font size. | true | A positive integer specifies the font size in pt, or several integers, separated by a colon as alternate font sizes. | 12 | - |
| **font-type** | Specifies whether a text should be printed italic or normal. | true | normal \| italic | normal | - |
| **font-weight** | Specifies whether a text should be printed bold or normal. | true | normal \| bold | normal | - |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **condense** | If the text is too long, it is automatically compressed using the optimal compression ratio. | true | One or more floating point number(s) separated by a colon. | 1.0 | |
| **line-spacing** | Sets the spacing between text lines in pixels (negative values decrease the line spacing). | true | A positive or negative integer value. | 0 | |
| **wrap** | Changes the line break in a text field.<br>**Normal:**<br>Line break at spaces and hyphens. | true | normal \| none | normal | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| | **None:**<br>No line breaks. | | | | |
| **letter-spacing** | The letter-spacing attribute increases or decreases the spacing between characters within a text. | true | A positive or negative floating point number. | 0.0 | |
| **superscript** | Used to produce superscript or subscript text. | true | superscript \| subscript \| normal | normal | |
| **outline** | Specifies whether the text should be outlined or not. | true | true \| false | false | - |
| **outline-color** | Specifies the color of the text outline. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **outline-thickness** | Specifies the thickness of the text outline. | true | A positive integer value (> 0) | 1 | - |
| **outline-inside** | This value specifies whether the outline should be drawn inside or outside of the text. | true | true \| false | false | - |
| **shadow** | Enables/disables the shadow. | true | true \| false | false | - |
| **shadow-alpha** | The alpha value (transparency) of the shadow. | true | A positive floating point number >= 0 and <= 1 | 0.5 | - |
| **shadow-color** | Sets the color of the shadow. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **shadow-size** | The size of the shadow in pixels. | true | A positive integer value (> 0) | 2 | - |
| **shadow-blur** | Specifies whether a shadow is rendered with or without a blur effect. | true | true \| false | false | - |

### Child elements

| | |
|---|---|
| <#text> | See the section *The element <#text>* |
| <br> | See the section *The element <br>* |
|  | See the section *The element * |
| <ul> | See the section *The element <ul> (unordered list)* |
| <ol> | See the section *The element <ol>* |

### Examples

- *4.4"* single and multi-line example

## 3.14 The element <label>

### Description:

The element `label` specifies a single line of text. The text is truncated if it is too long. There is no line break.

### Attributes

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |
| **font-family** | The name of the font that is to be selected. | true | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Arial | - |
| **align** | Sets the horizontal alignment of the element/text. | true | left \| right \| center \| justify | Left-aligned | - |
| **padding-bottom** | Specifies the padding in pixels from the bottom. | true | A positive or negative integer value | 0 | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **padding-left** | Specifies the padding in pixels from the left. | true | A positive or negative integer value | 0 | - |
| **padding-right** | Specifies the padding in pixels from the right. | true | A positive or negative integer value | 0 | - |
| **padding-top** | Specifies the padding in pixels from the top. | true | A positive or negative integer value | 0 | - |
| **background-color** | The background color of the element. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | Transparent | - |
| **text-decoration** | Defines the text decoration for a text (underline, overline, linethrough, or none). | true | underline \| linethrough \| none | None | |
| **font-size** | Sets a range of font sizes, separated by colon. The text is rendered with the largest possible (and fixed) font size. | true | A positive integer specifies the font size in pt, or several integers, separated by a colon as alternate font sizes. | 12 | - |
| **font-type** | Specifies whether a text should be printed italic or normal. | true | normal \| italic | normal | - |
| **font-weight** | Specifies whether a text should be printed bold or normal. | true | normal | normal | - |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **condense** | If the text is too long, it is automatically compressed using the optimal compression ratio. | true | One or more floating point numbers separated by a colon. | 1.0 | |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **letter-spacing** | The letter-spacing attribute increases or decreases the spacing between characters within a text. | true | A positive or negative floating point number. | 0.0 | |
| **superscript** | Used to produce superscript or subscript text. | true | superscript \| subscript \| normal | normal | |
| **outline** | Specifies whether the text should be outlined or not. | true | true \| false | false | - |
| **outline-color** | Specifies the color of the text outline. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **outline-thickness** | Specifies the thickness of the text outline. | true | A positive integer value (> 0) | 1 | - |
| **outline-inside** | This value specifies whether the outline should be drawn inside or outside of the text. | true | true \| false | false | - |
| **shadow** | Enables/disables the shadow | true | true \| false | false | - |
| **shadow-alpha** | The alpha value (transparency) of the shadow. | true | A positive floating point number >= 0 and <= 1 | 0.5 | - |
| **shadow-color** | Sets the color of the shadow. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **shadow-size** | The size of the shadow in pixels. | true | A positive integer value (> 0) | 2 | - |
| **shadow-blur** | Specifies whether a shadow is rendered with or without a blur effect. | true | true \| false | false | - |

### Child elements

<#text> See the section *The element <#text>*

 See the section *The element *

**Examples**

- *4.4"* single and multi-line example

## 3.15 The element 

**Description:**

The element span is specified inside the text element in order to format the corresponding text differently.

There is no automatic line wrapping after a span element.

**Attributes**

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |
| **font-family** | The name of the font that is to be selected. | true | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Arial | - |
| **background-color** | The background color of the element. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | Transparent | - |
| **text-decoration** | Defines the text decoration for a text (underline, overline, linethrough, or none). | true | underline \| linethrough \| none | None | |
| **font-size** | Sets a range of font sizes, separated by | true | A positive integer specifies the font | 12 | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| | colon. The text is rendered with the largest possible (and fixed) font size. | | size in pt, or several integers, separated by a colon as alternate font sizes. | | |
| **font-type** | Specifies whether a text should be printed italic or normal. | true | normal \| italic | normal | - |
| **font-weight** | Specifies whether a text should be printed bold or normal. | true | normal \| bold | normal | - |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **condense** | If the text is too long, it is automatically compressed using the optimal compression ratio. | true | One or more floating point number(s) separated by a colon. | 1.0 | |
| **line-spacing** | Sets the spacing between text lines in pixels (negative values decrease the line spacing). | true | A positive or negative integer value. | 0 | |
| **wrap** | Changes the line break in a text field.<br>**Normal:**<br>Line break at spaces and hyphens.<br>**None:**<br>No line breaks. | true | normal \| none | normal | - |
| **letter-spacing** | The letter-spacing attribute increases or decreases the spacing between characters within a text. | true | A positive or negative floating point number. | 0.0 | |
| **superscript** | Used to produce superscript or subscript text. | true | superscript \| subscript \| normal | normal | |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **outline** | Specifies whether the text should be outlined or not. | true | true \| false | false | - |
| **outline-color** | Specifies the color of the text outline. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **outline-thickness** | Specifies the thickness of the text outline. | true | A positive integer value (> 0) | 1 | - |
| **outline-inside** | This value specifies whether the outline should be drawn inside or outside of the text. | true | true \| false | false | - |
| **shadow** | Enables/disables the shadow | true | true \| false | false | - |
| **shadow-alpha** | The alpha value (transparency) of the shadow. | true | A positive floating point number >= 0 and <= 1 | 0.5 | - |
| **shadow-color** | Sets the color of the shadow. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **shadow-size** | The size of the shadow in pixels. | true | A positive integer value (> 0) | 2 | - |
| **shadow-blur** | Specifies whether a shadow is rendered with or without a blur effect. | true | true \| false | false | - |

### Child elements

<#text> See the section *The element <#text>*

### Examples

- *4,4"* Span examples

# 3.16 The element <ul> (unordered list)

**Description:**

The element `ul` specifies an unordered list, comparable with the HTML tag <ul>. The elements specified as "li" elements are then formatted in an unordered list.

**Attributes**

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |
| **font-family** | The name of the font that is to be selected. | true | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Arial | - |
| **background-color** | The background color of the element. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | Transparent | - |
| **text-decoration** | Defines the text decoration for a text (underline, overline, linethrough, or none). | true | underline \| linethrough \| none | None | |
| **font-size** | Sets a range of font sizes, separated by colon. The text is rendered with the largest possible (and fixed) font size. | true | A positive integer specifies the font size in pt, or several integers, separated by a colon as alternate font sizes. | 12 | - |
| **font-type** | Specifies whether a text should be printed italic or normal. | true | normal \| italic | normal | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **font-weight** | Specifies whether a text should be printed bold or normal. | true | normal \| bold | normal | - |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **condense** | If the text is too long, it is automatically compressed using the optimal compression ratio. | true | One or more floating point number(s) separated by a colon. | 1.0 | |
| **line-spacing** | Sets the spacing between text lines in pixels (negative values decrease the line spacing). | true | A positive or negative integer value. | 0 | |
| **wrap** | Changes the line break in a text field.<br>**Normal:**<br>Line break at spaces and hyphens.<br>**None:**<br>No line breaks. | true | normal \| none | normal | - |
| **letter-spacing** | The letter-spacing attribute increases or decreases the spacing between characters within a text. | true | A positive or negative floating point number. | 0.0 | |
| **superscript** | Used to produce superscript or subscript text. | true | superscript \| subscript \| normal | normal | |
| **outline** | Specifies whether the text should be outlined or not. | true | true \| false | false | - |
| **outline-color** | Specifies the color of the text outline. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **outline-thickness** | Specifies the thickness of the text outline. | true | A positive integer value (> 0) | 1 | - |
| **outline-inside** | This value specifies whether the outline should be drawn inside or outside of the text. | true | true \| false | false | - |
| **shadow** | Enables/disables the shadow | true | true \| false | false | - |
| **shadow-alpha** | The alpha value (transparency) of the shadow. | true | A positive floating point number >= 0 and <= 1 | 0.5 | - |
| **shadow-color** | Sets the color of the shadow. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **shadow-size** | The size of the shadow in pixels. | true | A positive integer value (> 0) | 2 | - |
| **shadow-blur** | Specifies whether a shadow is rendered with or without a blur effect. | true | true \| false | false | - |
| **bullet-font-family** | Sets the font for the bullets. | false | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | No default value | |
| **bullet-font-family** | Sets the font for the bullets. | false | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | No default value | |
| **padded** | Manual setting of padding, in pixels | false | A positive integer value (> 0) | No default value | |
| **type** | Specifies the type of bullet to use. | false | disc (•) \| dash (-) \| dot (.) \| colon (:) \| bracket ()) \| dot_bracket (.)) \| circle ( ) \| square ( ) \| common bullet characters | dash | |
| **float** | Used to make elements flow around a list. A floating list is wrapped the same | false | true \| false | false | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| | way as normal text, there is no line break after each element. The line break in a list is not after a bullet and is not before the first letter. | | | | |
| **wrapped-line-spacing** | Determines the line spacing for automatic line breaks within a list (negative values decrease the line spacing). | true | A positive or negative integer value. | 0 | |
| **bullet-image** | Path for an image that is used as a bullet. | false | A path relative to the template file or to a URL (e.g. "../test/image.png or url.to/image.png") | No default value | |

### Child elements

<li> See the section *The element <li>*

### Examples

- *4.4"* Example list

## 3.17 The element <ol>

### Description:

The element `ol` specifies an ordered list, comparable with the HTML tag <ol>. The elements specified as "li" elements are then formatted in an ordered list.

### Attributes

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |
| **font-family** | The name of the font that is to be selected. | true | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Arial | - |
| **background-color** | The background color of the element. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | Transparent | - |
| **text-decoration** | Defines the text decoration for a text (underline, overline, linethrough, or none). | true | underline \| linethrough \| none | None | |
| **font-size** | Sets a range of font sizes, separated by colon. The text is rendered with the largest possible (and fixed) font size. | true | A positive integer specifies the font size in pt, or several integers, separated by a colon as alternate font sizes. | 12 | - |
| **font-type** | Specifies whether a text should be printed italic or normal. | true | normal \| italic | normal | - |
| **font-weight** | Specifies whether a text should be printed bold or normal. | true | normal \| bold | normal | - |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **condense** | If the text is too long, it is automatically compressed using the optimal compression ratio. | true | One or more floating point number(s) separated by a colon. | 1.0 | |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **line-spacing** | Sets the spacing between text lines in pixels (negative values decrease the line spacing). | true | A positive or negative integer value. | 0 | |
| **wrap** | Changes the line break in a text field. Normal: Line break at spaces and hyphens. None: No line breaks. | true | normal \| none | normal | - |
| **letter-spacing** | The letter-spacing attribute increases or decreases the spacing between characters within a text. | true | A positive or negative floating point number. | 0.0 | |
| **superscript** | Used to produce superscript or subscript text. | true | superscript \| subscript \| normal | normal | |
| **outline** | Specifies whether the text should be outlined or not. | true | true \| false | false | - |
| **outline-color** | Specifies the color of the text outline. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **outline-thickness** | Specifies the thickness of the text outline. | true | A positive integer value (> 0) | 1 | - |
| **outline-inside** | This value specifies whether the outline should be drawn inside or outside of the text. | true | true \| false | false | - |
| **shadow** | Enables/disables the shadow | true | true \| false | false | - |
| **shadow-alpha** | The alpha value (transparency) of the shadow. | true | A positive floating point number >= 0 and <= 1 | 0.5 | - |
| **shadow-color** | Sets the color of the shadow. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **shadow-size** | The size of the shadow in pixels. | true | A positive integer value (> 0) | 2 | - |
| **shadow-blur** | Specifies whether a shadow is rendered with or without a blur effect. | true | true \| false | false | - |
| **padded** | Manual setting of padding, in pixels | false | A positive integer value (> 0) | No default value | |
| **type** | Specifies the type of bullet to use. | false | disc (•) \| dash (-) \| dot (.) \| colon (:) \| bracket ()) \| dot_bracket (.)) \| circle ( ) \| square ( ) \| common bullet characters | . | |
| **float** | Used to make elements flow around a list. A floating list is wrapped the same way as normal text, there is no line break after each element. The line break in a list is not after a bullet and is not before the first letter. | false | true \| false | false | - |
| **wrapped-line-spacing** | Determines the line spacing for automatic line breaks within a list (negative values decrease the line spacing). | true | A positive or negative integer value. | 0 | |
| **bullet-image** | Path for an image that is used as a bullet. | false | A path relative to the template file or to a URL (e.g. "../test/image.png or url.to/image.png") | No default value | |

### Child elements

<li> See the section *The element <li>*

### Examples

- *4.4"* Example list

# 3.18 The element <li>

**Description:**

The element `li` specifies a bulleted element, comparable with the HTML tag "li".

**Attributes**

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **class** | Identifier of the CSS class. This identifier can be referenced using a CSS file. The same class identifiers should be be used in the case of multiple tags. | false | String | No default value | *4.4"* on page 102 |
| **id** | The identifier of a tag. This identifier can be referenced using a CSS file. This identifier must be unique. | false | String | No default value | *4.4"* on page 102 |
| **font-family** | The name of the font that is to be selected. | true | A font name (for example, Arial, Tahoma, Times New Roman, etc.) | Arial | - |
| **background-color** | The background color of the element. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | Transparent | - |
| **text-decoration** | Defines the text decoration for a text (underline, overline, linethrough, or none). | true | underline \| linethrough \| none | None | |
| **font-size** | Sets a range of font sizes, separated by colon. The text is rendered with the largest possible (and fixed) font size. | true | A positive integer specifies the font size in pt, or several integers, separated by a colon as alternate font sizes. | 12 | - |
| **font-type** | Specifies whether a text should be printed italic or normal. | true | normal \| italic | normal | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **font-weight** | Specifies whether a text should be printed bold or normal. | true | normal \| bold | normal | - |
| **color** | The color of the element or of the contained text. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **condense** | If the text is too long, it is automatically compressed using the optimal compression ratio. | true | One or more floating point numbers separated by a colon. | 1.0 | |
| **line-spacing** | Sets the spacing between text lines in pixels (negative values decrease the line spacing). | true | A positive or negative integer value. | 0 | |
| **wrap** | Changes the line break in a text field. Normal: Line break at spaces and hyphens. None: No line breaks. | true | normal \| none | normal | - |
| **letter-spacing** | The letter-spacing attribute increases or decreases the spacing between characters within a text. | true | A positive or negative floating point number. | 0.0 | |
| **superscript** | Used to produce superscript or subscript text. | true | superscript \| subscript \| normal | normal | |
| **outline** | Specifies whether the text should be outlined or not. | true | true \| false | false | - |
| **outline-color** | Specifies the color of the text outline. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **outline-thickness** | Specifies the thickness of the text outline. | true | A positive integer value (> 0) | 1 | - |

| Name | Description | Inherited | Possible values | Default value | Examples |
|---|---|---|---|---|---|
| **outline-inside** | This value specifies whether the outline should be drawn inside or outside of the text. | true | true \| false | false | - |
| **shadow** | Enables/disables the shadow | true | true \| false | false | - |
| **shadow-alpha** | The alpha value (transparency) of the shadow. | true | A positive floating point number >= 0 and <= 1 | 0.5 | - |
| **shadow-color** | Sets the color of the shadow. | true | black \| white \| transparent \| semicolon-separated RGBA string with values between 0 and 255 | black | - |
| **shadow-size** | The size of the shadow in pixels. | true | A positive integer value (> 0) | 2 | - |
| **shadow-blur** | Specifies whether a shadow is rendered with or without a blur effect. | true | true \| false | false | - |
| **wrapped-line-spacing** | Determines the line spacing for automatic line breaks within a list (negative values decrease the line spacing). | true | A positive or negative integer value. | 0 | |
| **line-spacing** | Sets the spacing between text lines in pixels (negative values decrease the line spacing). | true | A positive or negative integer value. | 0 | |

### Child elements

<#text> See the section *The element <#text>*

<br> See the section *The element <br>*

 See the section *The element *

### Examples

- *4.4"* Example list

## 3.19 The element <br>

**Description:**

The element `br` is positioned within a text element and forces a line break. It does not need any attributes.

**Multiple occurrences allowed:** true

**Attribute**

None

**Child elements**

None

## 3.20 The element <#text>

**Description:**

Plain text used to generate a barcode or to be displayed in the rendered image.

**Multiple occurrences allowed:** true

**Attribute**

None

**Child elements**

None

# 4 Utils

The Utils template was created to change the text content for generated images.

In order to call this type of method, it must be defined in the template.

```
Utils example
<utils method="toUpperCase">text</utils>
```

As a result, the generated image displays "TEXT" instead of "text".

In the event that the method contains different parameters, these must be defined as an attribute in the Utils tag.

```
Utils example with multiple arguments
<utils method="replace" arg1="x" arg2="s">text</utils>
```

The output of this method is "test" instead of "text".

## 4.1 DefaultTemplateUtils

### replace

Replaces all strings in a text element with another string.

**Table 17: Parameters for replace**

| Name | Type | Description |
|---|---|---|
| Body | String | Search and replace text. |
| arg1 | String | The string to be searched for. |
| arg2 | String | The string to be inserted. |

### contains

Checks whether a text element contains a searched string.

**Table 18: Parameters for contains**

| Name | Type | Description |
|---|---|---|
| Body | String | Text to be checked |
| arg1 | String | Text to be searched for |

### split

Splits a text element.

**Table 19: Parameters for split**

| Name | Type | Description |
|---|---|---|
| Body | String | Text to be split |

| Name | Type | Description |
|---|---|---|
| arg1 | String | The character used as the separator |
| arg2 | String | Text location for the line break |

### toLowerCase

Converts text content to lowercase letters.

**Table 20: Parameters for toLowerCase**

| Name | Type | Description |
|---|---|---|
| Body | String | String to convert to lowercase. |

### toUpperCase

Converts text content to uppercase letters.

**Table 21: Parameters for toUpperCase**

| Name | Type | Description |
|---|---|---|
| Body | String | String to convert to uppercase. |

### format

Formats the given string with the specified format. Formatting uses the Java implementation of the `String.format()` method.

**Table 22: Parameters for format**

| Name | Type | Description |
|---|---|---|
| Body | String | The string to format |
| arg1 | String | The format string |

### lowerCase

Converts text content to lowercase letters.

**Table 23: Parameters for lowerCase**

| Name | Type | Description |
|---|---|---|
| Body | String | The string for lowercase letters |

### capitalize

Converts the first letter of text content to uppercase. No effect on the following letters.

**Table 24: Parameters for capitalize**

| Name | Type | Description |
|---|---|---|
| Body | String | The string for uppercase letters |

### formatNumber

Formats the given number with the specified format. Formatting uses the Java implementation of the `String.format()` method. For further information, see: http://docs.oracle.com/javase/tutorial/java/data/numberformat.html

**Table 25: Parameters for formatNumber**

| Name | Type | Description |
|---|---|---|
| Body | String | Number to be formatted |
| arg1 | String | The format string |

### deleteWhitespace

Removes all spaces from a string.

**Table 26: Parameters for deleteWhitespace**

| Name | Type | Description |
|---|---|---|
| Body | String | The string to delete whitespaces from. |

### abbreviate

Truncates a string by means of omission. The string "Now the time has come for all good people" is abbreviated to "Now the time has come for...".

**Table 27: Parameters for abbreviate**

| Name | Type | Description |
|---|---|---|
| Body | String | Text for truncation |
| arg1 | String | Maximum length of the result string (minimum 4) |

### getCurrentDate

Returns the current date in the specified format.

**Table 28: Parameters for getCurrentDate**

| Name | Type | Description |
|---|---|---|
| Body | String | The description of the desired date format |

### getNonBreakingSpace

Returns a non-breaking space to prevent automatic line breaks (char code: 160).

*No parameters*

### getOperatingSystem

Returns the current operating system installed on the Wireless ePaper Server. (WINDOWS|LINUX|MAC/OTHER)

*No parameters*

### isFontAvailable

Returns whether the specified font exists on the current system (true | false).

**Table 29: Parameters for isFontAvailable**

| Name | Type | Description |
|---|---|---|
| Body | String | The font name as a string |

### getCurrentWorkingDirectory

Returns the current working directory.

*No parameters*

### replaceUmlauts

Replaces all umlauts in the given text content (ä->ae, ö->oe, ü->ue).

**Table 30: Parameters for replaceUmlauts**

| Name | Type | Description |
|---|---|---|
| Body | String | Search and replace text |

### insertUmlauts

Re-inserts the umlauts removed earlier (ae->ä, oe->ö, ue->ü).

**Table 31: Parameters for insertUmlauts**

| Name | Type | Description |
|---|---|---|
| Body | String | Search and replace text |

### formatDate

Parses the date string by replacing the existing input format with the desired output format. y: year, D: day of the year, M: month of the year, d: day of the month, H: hour of the day, m: minute of the hour, s: second of the minute, z: time zone, E: day of the week, w: week of the year

**Table 32: Parameters for formatDate**

| Name | Type | Description |
|---|---|---|
| Body | String | Date string to be formatted |
| arg1 | String | The format for parsing the input date string |
| arg2 | String | The format for the output date string |

# 5 Examples

The following section provides a number of examples of how to generate graphics for Wireless ePaper Displays. The templates used here can be used in practice.

## 5.1 2.7"

**2.7" example image - Coca Cola**


**Figure 1: Generated image**

- Template:

```
2.7" record example - Coca Cola (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image font-family="Arial" height="176" width="264" >
      <line thickness="5" x-from="0" x-to="264" y-from="2" y-to="2" >
      </line>
      <field height="81" width="175" x="5" y="10" >
        <label condense="1.0,0.8,0.6,0.5" font-size="34,33,32,30,28,26,24,22,20"
font-weight="bold" >
          <xsl:value-of select="article/field[@key='name']/@value" >
          </xsl:value-of>
        </label>
        <text condense="1.0,0.8,0.6,0.5" font-size="20,19,18,16,12" padding-top="5" >
          <xsl:value-of select="article/field[@key='description']/@value" >
          </xsl:value-of>
        </text>
        <text condense="1.0,0.8,0.6,0.5" font-size="12" padding-top="3" >
          <ul float="true" type="disc" >
            <li>
              <span font-weight="bold" >
                Unit:
              </span>
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
              <xsl:value-of select="article/field[@key='pricePerUnit']/@value" >
              </xsl:value-of>
            </li>
          </ul>
        </text>
      </field>
      <field height="65" width="65" x="189" y="10" >
```

```
        <barcode autoscale="true" type="qr" >
          <xsl:value-of select="article/@articleNumber" >
          </xsl:value-of>
        </barcode>
      </field>
      <line thickness="5" x-from="0" x-to="264" y-from="92" y-to="92" >
      </line>
      <field height="79" valign="center" width="264" x="0" y="95" >
        <label align="center" condense="1.0,0.8,0.6,0.5"
font-size="70,68,66,64,62,60,58,56,54,52,50,48,46,44,42,40,38" font-style="italic"
font-weight="bold" >
          €
          <xsl:value-of select="article/field[@key='price']/@value" >
          </xsl:value-of>
        </label>
      </field>
      <line thickness="5" x-from="0" x-to="264" y-from="173" y-to="173" >
      </line>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
2.7" record example - Coca Cola (Record)
<article articleNumber="5449000000996" >
  <field key="pricePerUnit" value="1.65" >
  </field>
  <field key="unit" value="1 l" >
  </field>
  <field key="price" value="0.55" >
  </field>
  <field key="description" value="Can 0,33 l" >
  </field>
  <field key="name" value="Coca Cola" >
  </field>
</article>
```

**2.7" example image - Chili con carne**


**Figure 2: Generated image**

- Template:

```
2.7" record example - Chili con carne (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image font-family="Arial" height="176" width="264" >
      <line thickness="5" x-from="0" x-to="264" y-from="2" y-to="2" >
      </line>
      <field height="81" width="175" x="5" y="10" >
        <label condense="1.0,0.8,0.6,0.5" font-size="34,33,32,30,28,26,24,22,20"
font-weight="bold" >
```

```
          <xsl:value-of select="article/field[@key='name']/@value" >
          </xsl:value-of>
        </label>
        <text condense="1.0,0.8,0.6,0.5" font-size="20,19,18,16,12" padding-top="5" >
          <xsl:value-of select="article/field[@key='description']/@value" >
          </xsl:value-of>
        </text>
        <text condense="1.0,0.8,0.6,0.5" font-size="12" padding-top="3" >
          <ul float="true" type="disc" >
            <li>
              <span font-weight="bold" >
                Unit:
              </span>
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
              <xsl:value-of select="article/field[@key='pricePerUnit']/@value" >
              </xsl:value-of>
            </li>
          </ul>
        </text>
      </field>
      <field height="65" width="65" x="189" y="10" >
        <barcode autoscale="true" type="qr" >
          <xsl:value-of select="article/@articleNumber" >
          </xsl:value-of>
        </barcode>
      </field>
      <line thickness="5" x-from="0" x-to="264" y-from="92" y-to="92" >
      </line>
      <field height="79" valign="center" width="264" x="0" y="95" >
        <label align="center" condense="1.0,0.8,0.6,0.5"
font-size="70,68,66,64,62,60,58,56,54,52,50,48,46,44,42,40,38" font-style="italic"
font-weight="bold" >
          €
          <xsl:value-of select="article/field[@key='price']/@value" >
          </xsl:value-of>
        </label>
      </field>
      <line thickness="5" x-from="0" x-to="264" y-from="173" y-to="173" >
      </line>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
2.7" record example - Chili con carne (Record)
<article articleNumber="9000275639319" >
  <field key="pricePerUnit" value="4.98" >
  </field>
  <field key="unit" value="1 kg" >
  </field>
  <field key="price" value="2.49" >
  </field>
  <field key="description" value="Dose 500 g" >
  </field>
  <field key="name" value="Chili con Carne" >
  </field>
</article>
```

**2.7" example image - Red Bull**

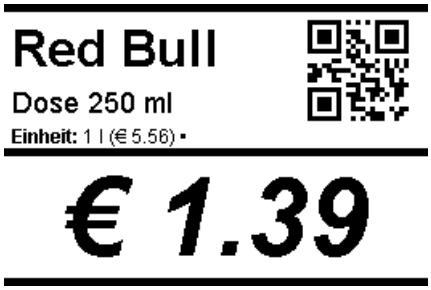



**Figure 3: Generated image**

- Template:

```
2.7" record example - Red Bull (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image font-family="Arial" height="176" width="264" >
      <line thickness="5" x-from="0" x-to="264" y-from="2" y-to="2" >
      </line>
      <field height="81" width="175" x="5" y="10" >
        <label condense="1.0,0.8,0.6,0.5" font-size="34,33,32,30,28,26,24,22,20"
font-weight="bold" >
          <xsl:value-of select="article/field[@key='name']/@value" >
          </xsl:value-of>
        </label>
        <text condense="1.0,0.8,0.6,0.5" font-size="20,19,18,16,12" padding-top="5" >
          <xsl:value-of select="article/field[@key='description']/@value" >
          </xsl:value-of>
        </text>
        <text condense="1.0,0.8,0.6,0.5" font-size="12" padding-top="3" >
          <ul float="true" type="disc" >
            <li>
              <span font-weight="bold" >
                Unit:
              </span>
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
              <xsl:value-of select="article/field[@key='pricePerUnit']/@value" >
              </xsl:value-of>
            </li>
          </ul>
        </text>
      </field>
      <field height="65" width="65" x="189" y="10" >
        <barcode autoscale="true" type="qr" >
          <xsl:value-of select="article/@articleNumber" >
          </xsl:value-of>
        </barcode>
      </field>
      <line thickness="5" x-from="0" x-to="264" y-from="92" y-to="92" >
      </line>
      <field height="79" valign="center" width="264" x="0" y="95" >
        <label align="center" condense="1.0,0.8,0.6,0.5"
font-size="70,68,66,64,62,60,58,56,54,52,50,48,46,44,42,40,38" font-style="italic"
font-weight="bold" >
          €
          <xsl:value-of select="article/field[@key='price']/@value" >
          </xsl:value-of>
        </label>
      </field>
```

```
      <line thickness="5" x-from="0" x-to="264" y-from="173" y-to="173" >
      </line>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
2.7" record example - Red Bull (Record)
<article articleNumber="9002490100070" >
  <field key="pricePerUnit" value="5.56" >
  </field>
  <field key="unit" value="1 l" >
  </field>
  <field key="price" value="1.39" >
  </field>
  <field key="description" value="Dose 250 ml" >
  </field>
  <field key="name" value="Red Bull" >
  </field>
  <field key="sale" value="1" >
  </field>
</article>
```

**2.7" example image - Vöslauer**


**Figure 4: Generated image**

- Template:

```
2.7" record example - Vöslauer (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image font-family="Arial" height="176" width="264" >
      <line thickness="5" x-from="0" x-to="264" y-from="2" y-to="2" >
      </line>
      <field height="81" width="175" x="5" y="10" >
        <label condense="1.0,0.8,0.6,0.5" font-size="34,33,32,30,28,26,24,22,20"
font-weight="bold" >
          <xsl:value-of select="article/field[@key='name']/@value" >
          </xsl:value-of>
        </label>
        <text condense="1.0,0.8,0.6,0.5" font-size="20,19,18,16,12" padding-top="5" >
          <xsl:value-of select="article/field[@key='description']/@value" >
          </xsl:value-of>
        </text>
        <text condense="1.0,0.8,0.6,0.5" font-size="12" padding-top="3" >
          <ul float="true" type="disc" >
            <li>
              <span font-weight="bold" >
                Unit:
              </span>
              <xsl:value-of select="article/field[@key='unit']/@value" >
```

```
              </xsl:value-of>
              <xsl:value-of select="article/field[@key='pricePerUnit']/@value" >
              </xsl:value-of>
            </li>
          </ul>
        </text>
      </field>
      <field height="65" width="65" x="189" y="10" >
        <barcode autoscale="true" type="qr" >
          <xsl:value-of select="article/@articleNumber" >
          </xsl:value-of>
        </barcode>
      </field>
      <line thickness="5" x-from="0" x-to="264" y-from="92" y-to="92" >
      </line>
      <field height="79" valign="center" width="264" x="0" y="95" >
        <label align="center" condense="1.0,0.8,0.6,0.5"
font-size="70,68,66,64,62,60,58,56,54,52,50,48,46,44,42,40,38" font-style="italic"
font-weight="bold" >
          €
          <xsl:value-of select="article/field[@key='price']/@value" >
          </xsl:value-of>
        </label>
      </field>
      <line thickness="5" x-from="0" x-to="264" y-from="173" y-to="173" >
      </line>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
2.7" record example - Vöslauer (Record)
<article articleNumber="9009700145104" >
  <field key="pricePerUnit" value="0.35" >
  </field>
  <field key="unit" value="1 l" >
  </field>
  <field key="price" value="0.35" >
  </field>
  <field key="description" value="Vöslauer sparkling, 1 l PET" >
  </field>
  <field key="name" value="Vöslauer mineral water" >
  </field>
</article>
```

**2.7" example image - Chili con carne**


**Figure 5: Generated image**

- Template:

```
2.7" record example - Chili con carne (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <xsl:variable name="price" >
      <xsl:value-of select="article/field[@key='price']/@value" >
      </xsl:value-of>
    </xsl:variable>
    <xsl:variable name="name" >
      <xsl:value-of select="article/field[@key='name']/@value" >
      </xsl:value-of>
    </xsl:variable>
    <xsl:variable name="description" >
      <xsl:value-of select="article/field[@key='description']/@value" >
      </xsl:value-of>
    </xsl:variable>
    <image height="176" width="264" font-family="Verdana" >
      <field height="80" width="225" x="5" y="7" >
        <text font-size="16" font-weight="bold" condense="1.0, 0.9, 0.8, 0.7"
line-spacing="-1" padding-bottom="5" >
          <xsl:value-of select="$name" >
          </xsl:value-of>
        </text>
        <text font-size="14" font-weight="normal" condense="1.0, 0.9, 0.8, 0.7"
line-spacing="-1" padding-bottom="5" >
          <xsl:value-of select="$description" >
          </xsl:value-of>
        </text>
      </field>
      <field height="100" width="140" x="0" y="75" >
        <text align="right" font-family="Lucida Sans" font-size="80" font-weight="bold"
condense="1.0" >
          <xsl:value-of select="substring-before($price, '.')" >
          </xsl:value-of>
        </text>
      </field>
      <field height="80" width="20" x="135" y="95" >
       <text align="center" font-family="Lucida Sans" font-size="60" font-weight="bold"
condense="1.0" >
          .
        </text>
      </field>
      <field height="90" width="80" x="135" y="79" >
        <label align="left" font-family="Lucida Sans" font-size="52" font-weight="bold"
condense="0.9" >
          <xsl:value-of select="substring-after($price, '.')" >
          </xsl:value-of>
        </label>
      </field>
     <field align="center" valign="center" rotation="270" height="25" width="150" x="230"
 y="17" >
        <barcode scale="1" humanReadableEnabled="false" type="code128" >
          1234567890ABCD
        </barcode>
      </field>
      <field height="14" width="100" y="160" x="5" >
        <label align="left" font-size="10" font-weight="normal" >
          <xsl:value-of select="article/@articleNumber" >
          </xsl:value-of>
        </label>
      </field>
    </image>
```

```
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
2.7" record example - Chili con carne (Record)
<article articleNumber="9000275639319" >
  <field key="pricePerUnit" value="4.98" >
  </field>
  <field key="unit" value="1 kg" >
  </field>
  <field key="price" value="2.49" >
  </field>
  <field key="description" value="Dose 500 g" >
  </field>
  <field key="name" value="Chili con Carne" >
  </field>
</article>
```

**2.7" example image - Red Bull**

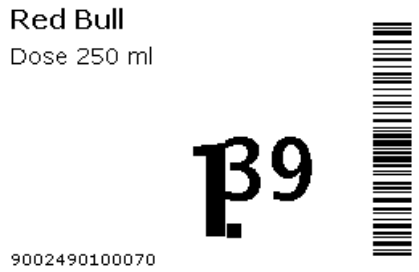



**Figure 6: Generated image**

- Template:

```
2.7" record example - Red Bull (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <xsl:variable name="price" >
      <xsl:value-of select="article/field[@key='price']/@value" >
      </xsl:value-of>
    </xsl:variable>
    <xsl:variable name="name" >
      <xsl:value-of select="article/field[@key='name']/@value" >
      </xsl:value-of>
    </xsl:variable>
    <xsl:variable name="description" >
      <xsl:value-of select="article/field[@key='description']/@value" >
      </xsl:value-of>
    </xsl:variable>
    <image height="176" width="264" font-family="Verdana" >
      <field height="80" width="225" x="5" y="7" >
        <text font-size="16" font-weight="bold" condense="1.0, 0.9, 0.8, 0.7"
line-spacing="-1" padding-bottom="5" >
          <xsl:value-of select="$name" >
          </xsl:value-of>
        </text>
        <text font-size="14" font-weight="normal" condense="1.0, 0.9, 0.8, 0.7"
line-spacing="-1" padding-bottom="5" >
          <xsl:value-of select="$description" >
          </xsl:value-of>
        </text>
      </field>
```

```
      <field height="100" width="140" x="0" y="75" >
        <text align="right" font-family="Lucida Sans" font-size="80" font-weight="bold"
condense="1.0" >
          <xsl:value-of select="substring-before($price, '.')" >
          </xsl:value-of>
        </text>
      </field>
      <field height="80" width="20" x="135" y="95" >
        <text align="center" font-family="Lucida Sans" font-size="60" font-weight="bold"
condense="1.0" >
          .
        </text>
      </field>
      <field height="90" width="80" x="135" y="79" >
        <label align="left" font-family="Lucida Sans" font-size="52" font-weight="bold"
condense="0.9" >
          <xsl:value-of select="substring-after($price, '.')" >
          </xsl:value-of>
        </label>
      </field>
     <field align="center" valign="center" rotation="270" height="25" width="150" x="230"
 y="17" >
        <barcode scale="1" humanReadableEnabled="false" type="code128" >
          1234567890ABCD
        </barcode>
      </field>
      <field height="14" width="100" y="160" x="5" >
        <label align="left" font-size="10" font-weight="normal" >
          <xsl:value-of select="article/@articleNumber" >
          </xsl:value-of>
        </label>
      </field>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
2.7" record example - Red Bull (Record)
<article articleNumber="9002490100070" >
  <field key="pricePerUnit" value="5.56" >
  </field>
  <field key="unit" value="1 l" >
  </field>
  <field key="price" value="1.39" >
  </field>
  <field key="description" value="Dose 250 ml" >
  </field>
  <field key="name" value="Red Bull" >
  </field>
  <field key="sale" value="1" >
  </field>
</article>
```

**2.7" example image - Vöslauer**


**Figure 7: Generated image**

- Template:

```
2.7" record example - Vöslauer (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <xsl:variable name="price" >
      <xsl:value-of select="article/field[@key='price']/@value" >
      </xsl:value-of>
    </xsl:variable>
    <xsl:variable name="name" >
      <xsl:value-of select="article/field[@key='name']/@value" >
      </xsl:value-of>
    </xsl:variable>
    <xsl:variable name="description" >
      <xsl:value-of select="article/field[@key='description']/@value" >
      </xsl:value-of>
    </xsl:variable>
    <image height="176" width="264" font-family="Verdana" >
      <field height="80" width="225" x="5" y="7" >
        <text font-size="16" font-weight="bold" condense="1.0, 0.9, 0.8, 0.7"
line-spacing="-1" padding-bottom="5" >
          <xsl:value-of select="$name" >
          </xsl:value-of>
        </text>
        <text font-size="14" font-weight="normal" condense="1.0, 0.9, 0.8, 0.7"
line-spacing="-1" padding-bottom="5" >
          <xsl:value-of select="$description" >
          </xsl:value-of>
        </text>
      </field>
      <field height="100" width="140" x="0" y="75" >
        <text align="right" font-family="Lucida Sans" font-size="80" font-weight="bold"
condense="1.0" >
          <xsl:value-of select="substring-before($price, '.')" >
          </xsl:value-of>
        </text>
      </field>
      <field height="80" width="20" x="135" y="95" >
        <text align="center" font-family="Lucida Sans" font-size="60" font-weight="bold"
condense="1.0" >
          .
        </text>
      </field>
      <field height="90" width="80" x="135" y="79" >
        <label align="left" font-family="Lucida Sans" font-size="52" font-weight="bold"
condense="0.9" >
          <xsl:value-of select="substring-after($price, '.')" >
          </xsl:value-of>
        </label>
```

```
      </field>
     <field align="center" valign="center" rotation="270" height="25" width="150" x="230"
 y="17" >
       <barcode scale="1" humanReadableEnabled="false" type="code128" >
         1234567890ABCD
       </barcode>
     </field>
     <field height="14" width="100" y="160" x="5" >
       <label align="left" font-size="10" font-weight="normal" >
         <xsl:value-of select="article/@articleNumber" >
         </xsl:value-of>
       </label>
     </field>
   </image>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
2.7" record example - Vöslauer (Record)
<article articleNumber="9009700145104" >
  <field key="pricePerUnit" value="0.35" >
  </field>
  <field key="unit" value="1 l" >
  </field>
  <field key="price" value="0.35" >
  </field>
  <field key="description" value="Vöslauer sparkling, 1 l PET" >
  </field>
  <field key="name" value="Vöslauer mineral water" >
  </field>
</article>
```

### 2.7" example of nested fields

A 2.7" template illustrating the use of nested fields.

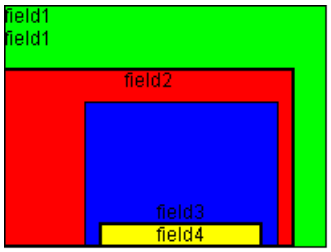



**Figure 8: Generated image**

- Template:

```
2,7" nested field example (Template)
<xsl:stylesheet version="1.0" >
  <xsl:template match="Record" >
    <image height="176" width="264" colors="true" >
      <field x="0" y="0" width="200" height="150" align="left" border="1"
background-color="green" >
       <text>
         field1
       </text>
       <text>
         field1
       </text>
       <field x="0" y="40" width="180" height="110" border="2" align="center"
```

```
background-color="red" >
          <text>
            field2
          </text>
          <field x="50" y="20" width="120" height="90" border="1" align="center"
valign="bottom" background-color="blue" >
            <text>
              field3
            </text>
            <field x="0" width="100" max-height="30" border="2" align="center"
background-color="yellow" >
              <text>
                field4
              </text>
            </field>
          </field>
        </field>
      </field>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

### Rendering of graphics for promotional or non-promotional items

This example illustrates the use of dynamic settings to render different graphics for products that are promotional and/or non-promotional items.


**Figure 9: Generated image**

- Template (non-promotional item):

```
Dynamic Sale/No Sale Image Rendering (Template)
<xsl:stylesheet version="1.0" >
  <xsl:template match="Record" >
    <image height="176" width="264" >
      <xsl:choose>
        <xsl:when test="article/field[@key='Sale']/@value='1'" >
          <rect color="black" height="126" width="264" x="0" y="0" >
          </rect>
          <field height="37" width="90" x="160" y="10" >
           <text color="white" font-family="Comic Sans MS" font-size="26" font-weight="bold"
 >
              Sale!
            </text>
          </field>
        </xsl:when>
        <xsl:otherwise>
          <line thickness="2" x-from="0" x-to="264" y-from="126" y-to="126" >
          </line>
        </xsl:otherwise>
      </xsl:choose>
      <field height="40" width="150" x="9" y="10" >
        <xsl:choose>
```

```
          <xsl:when test="article/field[@key='Sale']/@value='1'" >
            <xsl:attribute name="width" >
              150
            </xsl:attribute>
          </xsl:when>
          <xsl:otherwise>
            <xsl:attribute name="width" >
              240
            </xsl:attribute>
          </xsl:otherwise>
        </xsl:choose>
        <text font-family="Tahoma" font-size="14" >
          <xsl:if test="article/field[@key='Sale']/@value='1'" >
            <xsl:attribute name="color" >
              white
            </xsl:attribute>
          </xsl:if>
          <span font-weight="bold" >
            <utils method="toUpperCase" >
              <xsl:value-of select="article/field[@key='ProductDescription']/@value" >
              </xsl:value-of>
            </utils>
          </span>
        </text>
        <label font-family="Tahoma" font-size="16" padding-top="3" >
          <xsl:if test="article/field[@key='Sale']/@value='1'" >
            <xsl:attribute name="color" >
              white
            </xsl:attribute>
          </xsl:if>
          <xsl:value-of select="article/field[@key='ProductName']/@value" >
          </xsl:value-of>
        </label>
      </field>
      <field height="20" width="50" x="10" y="100" >
        <text align="left" font-family="Verdana" font-size="13" >
          <xsl:if test="article/field[@key='Sale']/@value='1'" >
            <xsl:attribute name="color" >
              white
            </xsl:attribute>
          </xsl:if>
          <xsl:value-of select="article/field[@key='TotalAmount']/@value" >
          </xsl:value-of>
        </text>
      </field>
      <field height="83" width="190" x="60" y="50" >
        <text align="right" condense="1,0.8,0.6,0.4" font-family="Impact"
font-size="68,64,60,58,56" >
          <xsl:if test="article/field[@key='Sale']/@value='1'" >
            <xsl:attribute name="color" >
              white
            </xsl:attribute>
          </xsl:if>
          <xsl:if test="article/field[@key='MainPrice']/@value!=''" >
           <xsl:value-of select="floor(number(article/field[@key='MainPrice']/@value) div
 100)" >
            </xsl:value-of>
           <xsl:value-of select="number(article/field[@key='MainPrice']/@value) mod 100"
 >
            </xsl:value-of>
          </xsl:if>
        </text>
      </field>
      <field height="14" width="70" x="180" y="131" >
```

```
       <text align="right" font-family="Verdana" font-size="11" >
         <xsl:if test="article/field[@key='UsedUnit']/@value!='' and
article/field[@key='UsedUnit']/@value!='0'" >
           <span>
             per
           </span>
           <xsl:if test="article/field[@key='UsedDimension']/@value='2'" >
             <span>
               100
             </span>
           </xsl:if>
         </xsl:if>
         <span>
           <xsl:choose>
             <xsl:when test="article/field[@key='UsedUnit']/@value='1'" >
               g
             </xsl:when>
             <xsl:when test="article/field[@key='UsedUnit']/@value='2'" >
               kg
             </xsl:when>
             <xsl:when test="article/field[@key='UsedUnit']/@value='3'" >
               l
             </xsl:when>
             <xsl:when test="article/field[@key='UsedUnit']/@value='4'" >
               ml
             </xsl:when>
           </xsl:choose>
         </span>
       </text>
     </field>
     <field height="24" width="70" x="180" y="146" >
       <text align="right" font-family="Tahoma" font-size="20" >
         <xsl:if test="article/field[@key='ReferencePrice']/@value!='" >
          <xsl:value-of select="floor(number(article/field[@key='ReferencePrice']/@value)
 div 100)" >
           </xsl:value-of>
          <xsl:value-of select="number(article/field[@key='ReferencePrice']/@value) mod
 100" >
           </xsl:value-of>
         </xsl:if>
       </text>
     </field>
     <field align="center" height="20" width="160" x="10" y="136" >
       <barcode autoscale="false" humanReadableEnabled="false" type="code128" >
         <xsl:value-of select="Label/Id" >
         </xsl:value-of>
       </barcode>
     </field>
     <field height="12" width="160" x="10" y="158" >
       <text align="center" font-family="Tahoma" font-size="10" >
         <span>
           <xsl:value-of select="number(article/field[@key='EAN']/@value)" >
           </xsl:value-of>
         </span>
         <span>
           <xsl:value-of select="article/field[@key='Location']/@value" >
           </xsl:value-of>
         </span>
         <xsl:if test="article/field[@key='Clearance']/@value='1'" >
           <span>
           </span>
           <span font-weight="bold" >
             R
           </span>
```

```
          </xsl:if>
          <span>
          </span>
          <span>
            <xsl:value-of select="article/field[@key='UpdateTime']/@value" >
            </xsl:value-of>
          </span>
        </text>
      </field>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
Dynamic Sale/No Sale Image Rendering (Record)
<article>
  <field key="LineID" value="1234567890" >
  </field>
  <field key="Version" value="1" >
  </field>
  <field key="Operator" value="1" >
  </field>
  <field key="CostCenter" value="1" >
  </field>
  <field key="NAN" value="0000000412342" >
  </field>
  <field key="EAN" value="3330001299874" >
  </field>
  <field key="MainPrice" value="999" >
  </field>
  <field key="UsedUnit" value="1" >
  </field>
  <field key="UsedDimension" value="2" >
  </field>
  <field key="ReferencePrice" value="1998" >
  </field>
  <field key="ProductDescription" value="Extra big schoko" >
  </field>
  <field key="ProductName" value="Corny" >
  </field>
  <field key="TotalAmount" value="50 g" >
  </field>
  <field key="Sale" value="0" >
  </field>
  <field key="Clearance" value="1" >
  </field>
  <field key="Location" value="08L03b" >
  </field>
  <field key="UpdateTime" value="135200" >
  </field>
  <field key="TagSize" value="1" >
  </field>
  <field key="MultiLocation" value="" >
  </field>
  <field key="Reserved" value="" >
```

```
  </field>
</article>
```


**Figure 10: Generated image**

- Template (promotional item):

```
Dynamic Sale/No Sale Image Rendering (Template)
<xsl:stylesheet version="1.0" >
  <xsl:template match="Record" >
    <image height="176" width="264" >
      <xsl:choose>
        <xsl:when test="article/field[@key='Sale']/@value='1'" >
          <rect color="black" height="126" width="264" x="0" y="0" >
          </rect>
          <field height="37" width="90" x="160" y="10" >
           <text color="white" font-family="Comic Sans MS" font-size="26" font-weight="bold"
 >
              Sale!
            </text>
          </field>
        </xsl:when>
        <xsl:otherwise>
          <line thickness="2" x-from="0" x-to="264" y-from="126" y-to="126" >
          </line>
        </xsl:otherwise>
      </xsl:choose>
      <field height="40" width="150" x="9" y="10" >
        <xsl:choose>
          <xsl:when test="article/field[@key='Sale']/@value='1'" >
            <xsl:attribute name="width" >
              150
            </xsl:attribute>
          </xsl:when>
          <xsl:otherwise>
            <xsl:attribute name="width" >
              240
            </xsl:attribute>
          </xsl:otherwise>
        </xsl:choose>
        <text font-family="Tahoma" font-size="14" >
          <xsl:if test="article/field[@key='Sale']/@value='1'" >
            <xsl:attribute name="color" >
              white
            </xsl:attribute>
          </xsl:if>
          <span font-weight="bold" >
            <utils method="toUpperCase" >
              <xsl:value-of select="article/field[@key='ProductDescription']/@value" >
              </xsl:value-of>
            </utils>
          </span>
        </text>
```

```
        <label font-family="Tahoma" font-size="16" padding-top="3" >
          <xsl:if test="article/field[@key='Sale']/@value='1'" >
            <xsl:attribute name="color" >
              white
            </xsl:attribute>
          </xsl:if>
          <xsl:value-of select="article/field[@key='ProductName']/@value" >
          </xsl:value-of>
        </label>
      </field>
      <field height="20" width="50" x="10" y="100" >
        <text align="left" font-family="Verdana" font-size="13" >
          <xsl:if test="article/field[@key='Sale']/@value='1'" >
            <xsl:attribute name="color" >
              white
            </xsl:attribute>
          </xsl:if>
          <xsl:value-of select="article/field[@key='TotalAmount']/@value" >
          </xsl:value-of>
        </text>
      </field>
      <field height="83" width="190" x="60" y="50" >
        <text align="right" condense="1,0.8,0.6,0.4" font-family="Impact"
font-size="68,64,60,58,56" >
          <xsl:if test="article/field[@key='Sale']/@value='1'" >
            <xsl:attribute name="color" >
              white
            </xsl:attribute>
          </xsl:if>
          <xsl:if test="article/field[@key='MainPrice']/@value!=''" >
           <xsl:value-of select="floor(number(article/field[@key='MainPrice']/@value) div
 100)" >
            </xsl:value-of>
           <xsl:value-of select="number(article/field[@key='MainPrice']/@value) mod 100"
 >
            </xsl:value-of>
          </xsl:if>
        </text>
      </field>
      <field height="14" width="70" x="180" y="131" >
        <text align="right" font-family="Verdana" font-size="11" >
          <xsl:if test="article/field[@key='UsedUnit']/@value!='' and
article/field[@key='UsedUnit']/@value!='0'" >
            <span>
              per
            </span>
            <xsl:if test="article/field[@key='UsedDimension']/@value='2'" >
              <span>
                100
              </span>
            </xsl:if>
          </xsl:if>
          <span>
            <xsl:choose>
              <xsl:when test="article/field[@key='UsedUnit']/@value='1'" >
                g
              </xsl:when>
              <xsl:when test="article/field[@key='UsedUnit']/@value='2'" >
                kg
              </xsl:when>
              <xsl:when test="article/field[@key='UsedUnit']/@value='3'" >
                l
              </xsl:when>
              <xsl:when test="article/field[@key='UsedUnit']/@value='4'" >
```

```
              ml
            </xsl:when>
          </xsl:choose>
        </span>
      </text>
    </field>
    <field height="24" width="70" x="180" y="146" >
      <text align="right" font-family="Tahoma" font-size="20" >
        <xsl:if test="article/field[@key='ReferencePrice']/@value!='" >
         <xsl:value-of select="floor(number(article/field[@key='ReferencePrice']/@value)
div 100)" >
          </xsl:value-of>
         <xsl:value-of select="number(article/field[@key='ReferencePrice']/@value) mod
100" >
          </xsl:value-of>
        </xsl:if>
      </text>
    </field>
    <field align="center" height="20" width="160" x="10" y="136" >
      <barcode autoscale="false" humanReadableEnabled="false" type="code128" >
        <xsl:value-of select="Label/Id" >
        </xsl:value-of>
      </barcode>
    </field>
    <field height="12" width="160" x="10" y="158" >
      <text align="center" font-family="Tahoma" font-size="10" >
        <span>
          <xsl:value-of select="number(article/field[@key='EAN']/@value)" >
          </xsl:value-of>
        </span>
        <span>
          <xsl:value-of select="article/field[@key='Location']/@value" >
          </xsl:value-of>
        </span>
        <xsl:if test="article/field[@key='Clearance']/@value='1'" >
          <span>
          </span>
          <span font-weight="bold" >
            R
          </span>
        </xsl:if>
        <span>
        </span>
        <span>
          <xsl:value-of select="article/field[@key='UpdateTime']/@value" >
          </xsl:value-of>
        </span>
      </text>
    </field>
  </image>
 </xsl:template>
</xsl:stylesheet>
```

- Example:

```
Dynamic Sale/No Sale Image Rendering (Record)
<article>
 <field key="LineID" value="1234567890" >
 </field>
 <field key="Version" value="1" >
 </field>
 <field key="Operator" value="1" >
 </field>
 <field key="CostCenter" value="1" >
 </field>
```

```
  <field key="NAN" value="0000000412342" >
  </field>
  <field key="EAN" value="3330001299874" >
  </field>
  <field key="MainPrice" value="799" >
  </field>
  <field key="UsedUnit" value="1" >
  </field>
  <field key="UsedDimension" value="2" >
  </field>
  <field key="ReferencePrice" value="1598" >
  </field>
  <field key="ProductDescription" value="Extra big schoko" >
  </field>
  <field key="ProductName" value="Corny" >
  </field>
  <field key="TotalAmount" value="50 g" >
  </field>
  <field key="Sale" value="1" >
  </field>
  <field key="Clearance" value="0" >
  </field>
  <field key="Location" value="08L03b" >
  </field>
  <field key="UpdateTime" value="135200" >
  </field>
  <field key="TagSize" value="1" >
  </field>
  <field key="MultiLocation" value="" >
  </field>
  <field key="Reserved" value="" >
  </field>
</article>
```

### Dynamic template based on a default graphic – 2.7"

This example is based on a dynamic template and the Coca Cola example. The image size depends on the type of display.

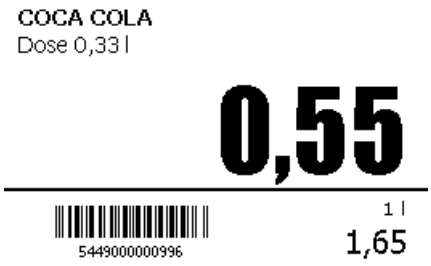



**Figure 11: Generated image**

- Template:

```
Dynamic template using default record - 2.7" (Template)
<xsl:stylesheet version="1.0" >
  <xsl:template match="Record" >
    <xsl:choose>
      <xsl:when test="Label/DisplayHeight = 176 and Label/DisplayWidth = 264" >
        <image height="176" width="264" >
          <line thickness="2" x-from="0" x-to="264" y-from="126" y-to="126" >
          </line>
          <field height="36" width="245" x="9" y="10" >
            <text font-family="Tahoma" font-size="14" >
              <span font-weight="bold" >
```

```
              <utils method="toUpperCase" >
                <xsl:value-of select="article/field[@key='name']/@value" >
                </xsl:value-of>
              </utils>
            </span>
            <br>
            </br>
            <xsl:value-of select="article/field[@key='description']/@value" >
            </xsl:value-of>
          </text>
        </field>
        <field height="83" width="190" x="60" y="47" >
          <text align="right" font-family="Impact" font-size="68" >
            <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
',')" >
            </xsl:value-of>
          </text>
        </field>
        <field height="14" width="70" x="180" y="131" >
          <text align="right" font-family="Verdana" font-size="11" >
            <xsl:value-of select="article/field[@key='unit']/@value" >
            </xsl:value-of>
          </text>
        </field>
        <field height="24" width="70" x="180" y="146" >
          <text align="right" font-family="Tahoma" font-size="20" >
            <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
            </xsl:value-of>
          </text>
        </field>
        <field align="center" height="20" width="140" x="10" y="136" >
          <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
            <xsl:value-of select="article/@articleNumber" >
            </xsl:value-of>
          </barcode>
        </field>
        <field height="12" width="140" x="10" y="158" >
          <text align="center" font-family="Tahoma" font-size="10" >
            <xsl:value-of select="article/@articleNumber" >
            </xsl:value-of>
          </text>
        </field>
      </image>
    </xsl:when>
    <xsl:when test="Label/DisplayHeight = 300 and Label/DisplayWidth = 400" >
      <image height="300" width="400" >
        <line thickness="2" x-from="0" x-to="400" y-from="236" y-to="236" >
        </line>
        <field height="90" width="382" x="9" y="10" >
          <text font-family="Tahoma" font-size="24" >
            <span font-weight="bold" >
              <utils method="toUpperCase" >
                <xsl:value-of select="article/field[@key='name']/@value" >
                </xsl:value-of>
              </utils>
            </span>
          </text>
          <text font-family="Tahoma" font-size="18" >
            <br>
            </br>
            <xsl:value-of select="article/field[@key='description']/@value" >
            </xsl:value-of>
          </text>
```

```
          </field>
          <field height="122" width="350" x="40" y="100" >
            <text align="right" font-family="Impact" font-size="100" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="320" y="248" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="320" y="266" >
            <text align="right" font-family="Tahoma" font-size="20" >
             <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="260" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="282" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 800 and Label/DisplayWidth = 480" >
        <image height="800" width="480" >
          <line thickness="2" x-from="0" x-to="480" y-from="700" y-to="700" >
          </line>
          <field height="400" width="460" x="10" y="10" >
            <text font-family="Tahoma" font-size="48" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="28" >
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="200" width="460" x="10" y="480" >
            <text align="right" font-family="Impact" font-size="120" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="32" width="100" x="370" y="728" >
            <text align="right" font-family="Verdana" font-size="26" >
```

```
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="32" width="100" x="370" y="760" >
            <text align="right" font-family="Tahoma" font-size="26" >
             <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="758" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="780" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
Dynamic template using default record - 2.7" (Record)
<article articleNumber="5449000000996" >
  <field key="pricePerUnit" value="1.65" >
  </field>
  <field key="unit" value="1 l" >
  </field>
  <field key="price" value="0.55" >
  </field>
  <field key="description" value="Dose 0,33 l" >
  </field>
  <field key="name" value="Coca Cola" >
  </field>
</article>
```

### Dynamic template based on a default graphic – 2.7"

This example is based on a dynamic template and the chili con carne example. The image size depends on the type of display.


**Figure 12: Generated image**

- Template:

```
Dynamic template using default record - 2.7" (Template)
<xsl:stylesheet version="1.0" >
  <xsl:template match="Record" >
    <xsl:choose>
      <xsl:when test="Label/DisplayHeight = 176 and Label/DisplayWidth = 264" >
        <image height="176" width="264" >
          <line thickness="2" x-from="0" x-to="264" y-from="126" y-to="126" >
          </line>
          <field height="36" width="245" x="9" y="10" >
            <text font-family="Tahoma" font-size="14" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="83" width="190" x="60" y="47" >
            <text align="right" font-family="Impact" font-size="68" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
',')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="180" y="131" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="180" y="146" >
            <text align="right" font-family="Tahoma" font-size="20" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="136" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="158" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 300 and Label/DisplayWidth = 400" >
        <image height="300" width="400" >
          <line thickness="2" x-from="0" x-to="400" y-from="236" y-to="236" >
          </line>
          <field height="90" width="382" x="9" y="10" >
            <text font-family="Tahoma" font-size="24" >
```

```
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="18" >
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="122" width="350" x="40" y="100" >
            <text align="right" font-family="Impact" font-size="100" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="320" y="248" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="320" y="266" >
            <text align="right" font-family="Tahoma" font-size="20" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="260" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="282" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 800 and Label/DisplayWidth = 480" >
        <image height="800" width="480" >
          <line thickness="2" x-from="0" x-to="480" y-from="700" y-to="700" >
          </line>
          <field height="400" width="460" x="10" y="10" >
            <text font-family="Tahoma" font-size="48" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="28" >
              <br>
              </br>
```

```
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="200" width="460" x="10" y="480" >
            <text align="right" font-family="Impact" font-size="120" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
',')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="32" width="100" x="370" y="728" >
            <text align="right" font-family="Verdana" font-size="26" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="32" width="100" x="370" y="760" >
            <text align="right" font-family="Tahoma" font-size="26" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="758" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="780" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
Dynamic template using default record - 2.7" (Record)
<article articleNumber="9000275639319" >
  <field key="pricePerUnit" value="4.98" >
  </field>
  <field key="unit" value="1 kg" >
  </field>
  <field key="price" value="2.49" >
  </field>
  <field key="description" value="Dose 500 g" >
  </field>
  <field key="name" value="Chili con Carne" >
  </field>
</article>
```

### Dynamic template based on a default graphic – 2.7"

This example is based on a dynamic template and the Red Bull example. The image size depends on the type of display.

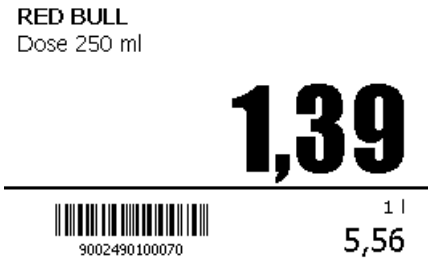



**Figure 13: Generated image**

- Template:

```
Dynamic template using default record - 2.7" (Template)
<xsl:stylesheet version="1.0" >
  <xsl:template match="Record" >
    <xsl:choose>
      <xsl:when test="Label/DisplayHeight = 176 and Label/DisplayWidth = 264" >
        <image height="176" width="264" >
          <line thickness="2" x-from="0" x-to="264" y-from="126" y-to="126" >
          </line>
          <field height="36" width="245" x="9" y="10" >
            <text font-family="Tahoma" font-size="14" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="83" width="190" x="60" y="47" >
            <text align="right" font-family="Impact" font-size="68" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
',')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="180" y="131" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="180" y="146" >
            <text align="right" font-family="Tahoma" font-size="20" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="136" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
```

```
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="158" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 300 and Label/DisplayWidth = 400" >
        <image height="300" width="400" >
          <line thickness="2" x-from="0" x-to="400" y-from="236" y-to="236" >
          </line>
          <field height="90" width="382" x="9" y="10" >
            <text font-family="Tahoma" font-size="24" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="18" >
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="122" width="350" x="40" y="100" >
            <text align="right" font-family="Impact" font-size="100" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
',')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="320" y="248" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="320" y="266" >
            <text align="right" font-family="Tahoma" font-size="20" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="260" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="282" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
```

```
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 800 and Label/DisplayWidth = 480" >
        <image height="800" width="480" >
          <line thickness="2" x-from="0" x-to="480" y-from="700" y-to="700" >
          </line>
          <field height="400" width="460" x="10" y="10" >
            <text font-family="Tahoma" font-size="48" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="28" >
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="200" width="460" x="10" y="480" >
            <text align="right" font-family="Impact" font-size="120" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
',')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="32" width="100" x="370" y="728" >
            <text align="right" font-family="Verdana" font-size="26" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="32" width="100" x="370" y="760" >
            <text align="right" font-family="Tahoma" font-size="26" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="758" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="780" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
Dynamic template using default record - 2.7" (Record)
<article articleNumber="9002490100070" >
  <field key="pricePerUnit" value="5.56" >
  </field>
```

```
  <field key="unit" value="1 l" >
  </field>
  <field key="price" value="1.39" >
  </field>
  <field key="description" value="Dose 250 ml" >
  </field>
  <field key="name" value="Red Bull" >
  </field>
  <field key="sale" value="1" >
  </field>
</article>
```

### Dynamic template based on a default graphic – 2.7"

This example is based on a dynamic template and the Vöslauer example. The image size depends on the type of display.


**Figure 14: Generated image**

- Template:

```
Dynamic template using default record - 2.7" (Template)
<xsl:stylesheet version="1.0" >
  <xsl:template match="Record" >
    <xsl:choose>
      <xsl:when test="Label/DisplayHeight = 176 and Label/DisplayWidth = 264" >
        <image height="176" width="264" >
          <line thickness="2" x-from="0" x-to="264" y-from="126" y-to="126" >
          </line>
          <field height="36" width="245" x="9" y="10" >
            <text font-family="Tahoma" font-size="14" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="83" width="190" x="60" y="47" >
            <text align="right" font-family="Impact" font-size="68" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
',')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="180" y="131" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
```

```
            </text>
          </field>
          <field height="24" width="70" x="180" y="146" >
            <text align="right" font-family="Tahoma" font-size="20" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="136" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="158" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 300 and Label/DisplayWidth = 400" >
        <image height="300" width="400" >
          <line thickness="2" x-from="0" x-to="400" y-from="236" y-to="236" >
          </line>
          <field height="90" width="382" x="9" y="10" >
            <text font-family="Tahoma" font-size="24" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="18" >
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="122" width="350" x="40" y="100" >
            <text align="right" font-family="Impact" font-size="100" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
',')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="320" y="248" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="320" y="266" >
            <text align="right" font-family="Tahoma" font-size="20" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="260" >
```

```
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="282" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 800 and Label/DisplayWidth = 480" >
        <image height="800" width="480" >
          <line thickness="2" x-from="0" x-to="480" y-from="700" y-to="700" >
          </line>
          <field height="400" width="460" x="10" y="10" >
            <text font-family="Tahoma" font-size="48" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="28" >
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="200" width="460" x="10" y="480" >
            <text align="right" font-family="Impact" font-size="120" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
',')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="32" width="100" x="370" y="728" >
            <text align="right" font-family="Verdana" font-size="26" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="32" width="100" x="370" y="760" >
            <text align="right" font-family="Tahoma" font-size="26" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="758" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="780" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
```

```
          </field>
        </image>
      </xsl:when>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
Dynamic template using default record - 2.7" (Record)
<article articleNumber="9009700145104" >
  <field key="pricePerUnit" value="0.35" >
  </field>
  <field key="unit" value="1 l" >
  </field>
  <field key="price" value="0.35" >
  </field>
  <field key="description" value="Vöslauer sparkling, 1 l PET" >
  </field>
  <field key="name" value="Vöslauer mineral water" >
  </field>
</article>
```

### Customized graphic – 2.7"


**Figure 15: Generated image**

- Template:

```
Custom Record 2.7" Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <xsl:variable name="price" >
      <xsl:value-of select="translate(Article/Price,'.','')" >
      </xsl:value-of>
    </xsl:variable>
    <image height="176" width="264" >
      <field height="20" width="256" x="8" y="8" >
        <text font-family="Verdana" font-size="10" text-decoration="underline" >
          <xsl:value-of select="Article/Category" >
          </xsl:value-of>
        </text>
      </field>
      <field height="19" width="256" x="8" y="28" >
        <text font-family="Verdana" font-size="16" font-weight="bold" >
          <utils method="toUpperCase" >
            <xsl:value-of select="Article/Manufacturer" >
            </xsl:value-of>
          </utils>
        </text>
      </field>
      <field height="19" width="256" x="8" y="48" >
        <text font-family="Verdana" font-size="16" >
```

```
          <xsl:value-of select="Article/Name" >
          </xsl:value-of>
        </text>
      </field>
      <xsl:choose>
        <xsl:when test="substring-after($price,',') = '00'" >
          <field height="78" width="250" x="0" y="83" >
            <text align="right" font-family="Tahoma" font-size="65" font-style="italic"
font-weight="bold" >
              <xsl:value-of select="substring-before($price,',')" >
              </xsl:value-of>
            </text>
          </field>
        </xsl:when>
        <xsl:otherwise>
          <field height="78" width="210" x="0" y="83" >
            <text align="right" font-family="Tahoma" font-size="65" font-style="italic"
font-weight="bold" >
              <xsl:value-of select="substring-before($price,',')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="78" width="64" x="200" y="83" >
            <text align="left" font-family="Tahoma" font-size="35" font-style="italic"
font-weight="bold" >
              <xsl:value-of select="substring-after($price,',')" >
              </xsl:value-of>
            </text>
          </field>
        </xsl:otherwise>
      </xsl:choose>
      <field height="8" width="105" x="4" y="165" >
        <barcode humanReadableEnabled="false" type="code128" >
          <xsl:value-of select="Article/ArticleNumber" >
          </xsl:value-of>
        </barcode>
      </field>
      <field height="13" width="60" x="3" y="153" >
        <text font-family="Tahoma" font-size="11" >
          <xsl:value-of select="Article/ArticleNumber" >
          </xsl:value-of>
        </text>
      </field>
      <field align="right" height="8" width="146" x="114" y="165" >
        <barcode humanReadableEnabled="false" type="code128" >
          <xsl:value-of select="Label/Id" >
          </xsl:value-of>
        </barcode>
      </field>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
Custom Record 2.7" Example (Record)
<Article>
  <ArticleNumber>
    1183821
  </ArticleNumber>
  <Name>
    Coolpix S9200 Silber
  </Name>
  <Manufacturer>
    Nikon
```

```
  </Manufacturer>
  <Category>
    Digital camera
  </Category>
  <Price>
    129,90
  </Price>
  <Specification>
    <Field>
      Photo resolution: 16 Megapixel max.
    </Field>
    <Field>
      Zoom range: 25 to 450 mm
    </Field>
    <Field>
      Storage medium: SD/SDHC/SDXC
    </Field>
    <Field>
      Available file formats: JPEG, MOV, MPEG4
    </Field>
    <Field>
      Video resolution: 1980 x 1080 pixels
    </Field>
    <Field>
      Zoomfaktor: 18x
    </Field>
    <Field>
      Optical image stabilizer
    </Field>
    <Field>
      Rel. aperture: F 1:3,5 to 5,9
    </Field>
    <Field>
      Film speed: 3200 ISO
    </Field>
  </Specification>
  <Packaging>
    <Item>
      Carrying strap
    </Item>
    <Item>
      USB cable
    </Item>
    <Item>
      Power adapter
    </Item>
  </Packaging>
  <Accessories>
    <Item>
      Rechargeable battery EN-EL12
    </Item>
    <Item>
      SDHC card
    </Item>
  </Accessories>
</Article>
```

# 5.2 4.4"

### List example

An example that shows the various uses of the list tag.

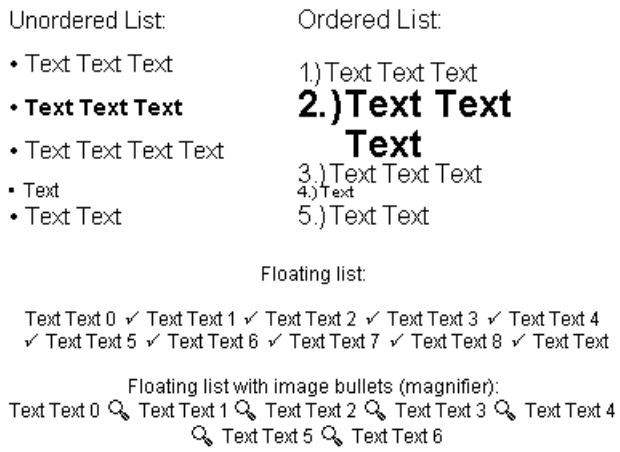



**Figure 16: Generated image**

- Template:

```
List Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image height="300" width="400" >
      <field height="150" width="150" x="10" y="10" >
        <text font-family="Arial" font-size="15" line-spacing="10" >
          Unordered List:
          <ul line-spacing="10" spacing="5" type="disc" wrapped-line-spacing="1" >
            <li>
              Text Text Text
            </li>
            <li>
              <span font-size="18,16,14" font-weight="bold" >
                Text Text Text
              </span>
            </li>
            <li wrapped-line-spacing="15" >
              Text Text Text Text
            </li>
            <li font-size="12" line-spacing="0" >
              Text
            </li>
            <li>
              Text Text
            </li>
          </ul>
        </text>
      </field>
      <field height="150" width="150" x="190" y="10" >
        <text font-family="Arial" font-size="16" line-spacing="15" >
          Ordered List:
          <ol line-spacing="-4" spacing="2" type="DOT_BRACKET" wrapped-line-spacing="1" >
```

```
          <li>
            Text Text Text
          </li>
          <li font-size="24,22,20,18" font-weight="bold" wrapped-line-spacing="-3" >
            Text Text Text
          </li>
          <li wrapped-line-spacing="20" >
            Text Text Text
          </li>
          <li font-size="10" line-spacing="0" >
            Text
          </li>
          <li>
            Text Text
          </li>
        </ol>
      </text>
    </field>
    <field height="56" width="380" x="10" y="170" >
      <text align="center" font-size="12" >
        Floating list:
        <br>
        </br>
        <ul float="true" bullet-font-family="Wingdings" spacing="5" type="ü" >
          <li>
            Text Text 0
          </li>
          <li>
            Text Text 1
          </li>
          <li>
            Text Text 2
          </li>
          <li>
            Text Text 3
          </li>
          <li>
            Text Text 4
          </li>
          <li>
            Text Text 5
          </li>
          <li>
            Text Text 6
          </li>
          <li>
            Text Text 7
          </li>
          <li>
            Text Text 8
          </li>
          <li>
            Text Text
          </li>
        </ul>
      </text>
    </field>
    <field height="56" width="380" x="10" y="240" >
      <text align="center" font-size="12" >
        Floating list with image bullets (magnifier):
        <ul float="true" spacing="4" bullet-image="../images/magnifier.png" >
          <li>
            Text Text 0
          </li>
```

```
            <li>
              Text Text 1
            </li>
            <li>
              Text Text 2
            </li>
            <li>
              Text Text 3
            </li>
            <li>
              Text Text 4
            </li>
            <li>
              Text Text 5
            </li>
            <li>
              Text Text 6
            </li>
          </ul>
        </text>
      </field>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

**Relative positioning example**

An example that shows the relative positioning of fields and the use of the attribute `max-height`.


**Figure 17: Generated image**

- Template:

```
Relative Positioning Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image font-size="16" height="300" width="400" >
      <field max-height="200" width="200" x="10" y="20" >
        <text>
          Fixed Position
        </text>
      </field>
      <field max-height="100" width="200" x="10" >
        <label>
          Relative Position
        </label>
```

```
      </field>
      <field max-height="100" width="200" x="10" y="200" >
        <text>
          Second fixed position
        </text>
      </field>
      <field background-color="black" max-height="50" width="100" x="10" >
        <text color="white" >
          Relative 1
        </text>
      </field>
      <field background-color="black" max-height="50" width="100" x="10" >
        <text color="white" >
          Relative 2
        </text>
      </field>
      <field max-height="50" width="100" x="10" >
        <text>
          Relative 3
        </text>
      </field>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

### Single and multi-line example

This example shows the difference between the label tag (single lines only) and the text tag (multiple lines with automatic line break). It also shows how the letter spacing changes the text output.

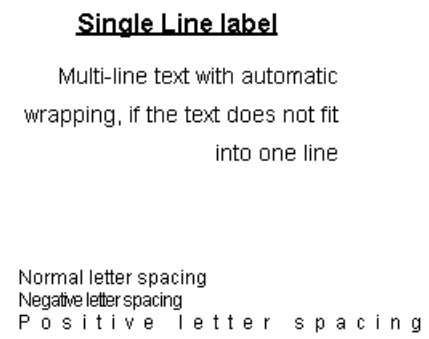



**Figure 18: Generated image**

- Template:

```
Single and Multi Line Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image height="300" width="400" >
      <field x="10" y="20" width="200" height="105" >
       <label font-size="16" font-weight="bold" align="center" text-decoration="underline"
 padding-bottom="15" >
          Single Line label
        </label>
        <text font-size="14" align="right" line-spacing="8" >
          Multi-line text with automatic wrapping, if the text does not fit into one line
        </text>
```

```
      </field>
      <field x="10" y="180" width="300" height="100" >
        <text letter-spacing="0" >
          Normal letter spacing
        </text>
        <text letter-spacing="-0.08" >
          Negative letter spacing
        </text>
        <text letter-spacing="0.5" >
          Positive letter spacing
        </text>
      </field>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

### Drawing example

This example shows different types of pictures with different borders, background colors and thicknesses. It also shows how fields are positioned on the elements, such as lines or rectangles.

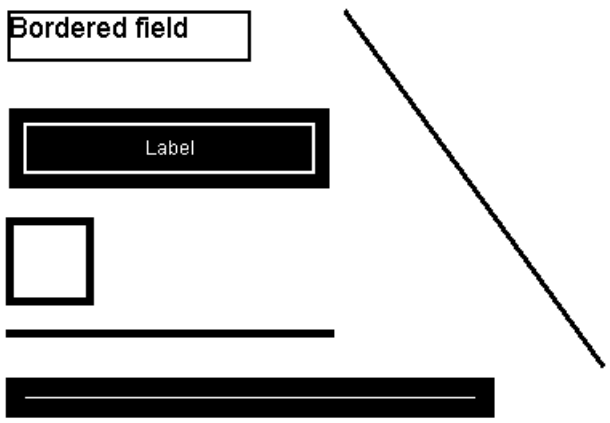



**Figure 19: Generated image**

- Template:

```
Drawing Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image height="300" width="400" >
      <field border="2" height="30" width="150" x="10" y="20" >
        <label font-size="18" >
          Bordered field
        </label>
      </field>
      <rect height="50" width="200" x="10" y="80" >
      </rect>
      <field background-color="black" border="2" border-color="white" height="30"
valign="center" width="180" x="20" y="90" >
        <label align="center" color="white" >
          Label
        </label>
      </field>
      <rect border="5" fill="false" height="50" width="50" x="10" y="150" >
      </rect>
      <line thickness="5" x-from="10" x-to="210" y-from="220" y-to="220" >
      </line>
```

```
      <rect border="5" height="20" width="300" x="10" y="250" >
      </rect>
      <line color="white" x-from="20" x-to="300" y-from="260" y-to="260" >
      </line>
      <line thickness="3" x-from="220" x-to="380" y-from="20" y-to="240" >
      </line>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

### Image example

This example shows the use of the image tag to insert images into the template, either by using a relative path or by inserting a Base64 string into the template.


**Figure 20: Generated image**

- Template:

```
Image Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image height="300" width="400" >
      <field align="center" height="80" width="380" x="10" y="10" >
        <img autoscale="true" src="../images/imagotag.png" >
        </img>
      </field>
      <field align="center" height="190" width="380" x="10" y="100" >
        <img
data="iVBORw0KGgoAAAANSUhEUgAAAQgAAACwAQAAAADbY83+AAALVElEQVR42u2YfVxTVR/ABByIISpvx
gIrplmWImvj4miJe0seQLCaZYvLBqWm+O58UMWBcFA0RX8iXtHhwhBaCAVYKKcJEUCzC+VYzFSYK
DaWx4Rhju7vnPOecy4Ah0efT3+6P+zm7n+/9vZ9zfufw4N/9eM+J50TXnwJCU8+EDsJN1CLW+wN
bqAiYp4lzGN5g/eTUQOER7ojLvOhOZKMmO4ttYZCV+hM4aGpey0GIZ0J9Ro8ZLu3dFcspVBXu3eE
pCthSVWqb0BjDY1EWBBBP0PUt4aH6yCQSSAE4d3KSG3updyYVxe3Eo0LIDz3DAGSh4bkPVXU0qno
j7o7S1un+e+F2dNXMHrOW8MzRO2HVR5Mjk/N2ic08Vb5DPGVXBczCQotwnwhEaK2J1gI8mobslRq
vk6iyUMvokAXX/TQnJIqb6SNavMQ8y0cr67Ebrp5hVgGi8E0kDbjMXpR1SUeYDhdszDA2qSDqRGH
I5rRm8YuRKsD/cP8zxWatXcLqzz1LI53kX08Wl6mxxTPZwZYUhNVoBagAmDS7Imyl+OSPxjEKlRQ
ri43Q+Quy9oTKdOAF98RxvxOh+Qd8ZLgStTaE4GnrboLiYdbIyATNya9CtUxbLIjQGiGZcvlUCu8
IJFtUI42PepUQBxh5l1s0k52rKpiPSM/91tnwt+r7bzVOx9q2B6dWR2h82nO1Z1U/Ag7CpkjWnoH
BjWnZ4CJmyOa1jf+QmJhsstc/YFX15atW7zVb2uRkf8wJGwxCpkBCdnWTlydtu0FUUHBn48ZYWWQ
9UPTEUSYTRDcpW3E9TPveH6XFmDhGcRDrgFxQSoqZisDGW+5jdi+i/fqIotkkeogv/Ji0ZqAUmwD
DbW/+tmIjB938VJ/Wr1BdItvroAxitJy9LIY3og7YiMuGkcMv/2+v/FQqVIxs8x34hHsqppufnjK
RgR/1ccreXd21JVBLVLXfG/FnSgcEEVCtNFGvP1Q4Drh7LqdZwbTMA4my+fh2WRQx/aeYCPcbgr6
OX2jkPluU+vo6CmT+2fj4m7sO3aBjZizo3/gsOBIreYRmxdn2TA84gQKBID+99fRbUSvS5Gim+ac
aL5Bu7/6k7rL9Y44beKdjeE2Yk+f953c97mupszWKwCIm1Y+wsE8NghGthHAIThsxqCP1x6lmABK
cxPUic142l1xjiqi2og9G+/49q30eqBgztItfuZ0YRPWontNWG8jRIKTovzCneWQOXs4v8EYqmgQ
```

```
4/znVShtxIyl5plzNl+KlFkUTzT+QQ/o1jBceXlzrgg5gp3tMDA4OezfWvio6JyiaFM2vQStuLBZ
OfmJnCOsriVbHCTqeUrm/LWD8tyhGbL3cNhbyiljG9H8mkPfF2//nEPBmJ9hIS15uGnRRLw2hrv8
0UbUHztQ+q8RoyQWyjAgjbkRvgBkRGPl58QP93PEpSHh1CzFhVwIQ+k6mTZpNBs4FSuPKNW2EV/F
Op/asezKMasi4etNzOD6N0D9WFzL4oaaNmImWFCRvktX/Ov0L4roqD8D7hj7jEOE0afBICYEcNn6
iTCjKPouoB9diNFm556rSvJCQQXfqw18TobXGb+88/kQ6GOPu7iDkqcTrUkmvIScK2jiCHbok6P8
+Cl00d7Ipg/gvaL3Ci4PN6P5YJ5pYjnCEqsPleROaY66ajixUAfls+/dEeSgQFhpKeQsbZ6nXUvv
HwAtZWCjm3HP/d+jnjq9gz5mAyJaVIR4Wu76qS5D7R9y07i68beYUOmAh4JVYuzBMUs5Iawjkubn
qRaV16f4359QmRPZpMl3jg5CHrwoXeZCiAeO/XLgnXDleVHM+v5LVMCou+49EG8yu0yaNoK3KIuW
BEkztkt/mT+VrqhWXh8hwnt6CVNTQIir3y16DN3fbclylT641DdEeh3+1lvA7fralYQoaRxJR4wO
8U5/derpJY4/NDVITjo4rCYroCGREAaLlN4ycNYWQbrssGCX3Cituu30xctExiucHZllQsX9cZ5U
PDWhpWS2+7wzivujB/oS4hsuYiXGXAs0L93cC9LLY5asiL1FnXYULefscFFj4mC6B1Ndv3EgHwaE
lS5zKV8uWzPt2wuE+CmTEC4puxfTD76/CmFclDV72MhP6Ks1wYvJjn2b1Ac4LAhLjEu49AqE46Ww
8mbaN1UC55e8CXGPIz7rv0aYf2oGSkGcFDJbeRZ45sCnYtJknCC+gP5L0qNuN7zuA+F0MbTClPUt
Ih+PWYTwJJayHq0XlX8ox6BcxSDnhsoiwMlJbu/ipRRG8jFhfWPu9jkrRvdBAmm0fuqezNMG9u/1
C15SwZQhRMto54itsxZjgsJ78ebExvhxK8pwci3LhyoQYfF4afrJbXNC0feHhcR+36SPcj49joZ1
ekK0upkjL/a+OxbVnQIvLBoNWyMKTB+PhpXfCmksY9l7zKHj9bOR6+OweazJ4Dt41fY30NDbKsV2
WEJrjW47GnhIxnhinlL1tihLOZSGQLWQEODFivcPDBoW6IdWULJ1/W/LAyf3F/wSoebsURMmmvod
HO749ZuxfGhlfTChZpdc186lxvKmm4YRovmD1+sXaZP2oYhznd7v7EifMXUBVPm9uzUGTOjd5u8O
PD38v3xoZJMxUXy61CkpTTiq6M/CBIZoWTc12Ts9LV6MJPiTZs5SFvmft2AFMEnLCaF/c/WcDQmy
8ygn00k7AG5s9hrWGO1Z6SUVczIE37n1d2d7URDIMdHasiclZbmj9NxKJl9Cdg/9cb13ShQ7E6X7
JzNqsSosJdcnHthrPKX32UxxhOvIVW/uh4E44LSZgkJ2bv6+wI/YU4V5N7n9Vp+QcHIvZfanADQt
hvGAhbwRN7d9SY8qpD/j9mzD12/pz0BmtARtwY3Ilh2SuX5Oy64FZcPJCW3EnBkzMqExnjJBSyMy
dKgs0Fsg8JySCze6cIT19I6FqLtaKpSjzY/MEoHjXAfepBP0fIojgPQyKhxQK6EhW0wipq+qHtZ7
7r4Qka27WIrLSb0QfWAlud2fQ43f6bZzzc9Bdr1lYXwceuaQ3GbQtasFosDblB1RdZG1tVjqOvS6
n5Noin1/Kg1l8AwiZwbcSrXyeJQ90YIbCfY4af113Z092OJMVD0syb4mpjuCsfLxM5HUWHR3hMkq
RLKZlYXYUrW8Oxl6bFfzbJ4jhNfUhq4E8lP3MU26ZdiKtKgBXUbby+BDUIe7b/w6EY5SQB1dbk+g
Db8EEVZcpK0KWoHmxSN7ohSSNzhmQHEsH4mqZe2JJkAnooaUGChhryFC7dnVF8BHBDkfxawMxsZf
60Iw1TgE5CwxcxmeeGB5F0L3BD9JoN5dR6rT2oW49z2NXCV9r4Mv4HdD4FjRkKQ7YdWCxG4IcSx+
EuLqiAPxdOdzEtdrw8doAeK06D12fIvVaDoRUvh53i0IaMjlorW32wMKLUSdZVgbYSZWbMtW/DQ+
Ojp0OldqLGoUCksHgfy2KkgO2og6JhNizZ2I6rFybJiNMKmiTEK0RIEOQuWgaP8AEcyXF13kSG0n
og74tB/osKWnXMrlyP/qjsozRk5uP59gIgKa1ShczR1E09YCUlhtBPgB1SGyy9jpqCuQtkceEwWQ
1aEAyToItii8c16AFAId8r4QAsp2kt1SbJc5MWzRKTHBtqfLyy5zQIwLFWezpf3DcrsZBSTcBcF9
XljnuUZ30sJp16p1mi4XHF0JCTT81d0FVwpZso7TZudrg45TdNbKwD/+SgaDjTJmUX99Q2LBRhlP
9HCHYsbEWdeeCJymsx49EThNh3x6IKyLkW9H5T0QYMt+yOzp8bZn0j2oH9wj4aWSLVP1SGy/v3mm
uEfizvrXBlI93zo5Jwf/zb1Uazz9/DbwOfGPiP8D6F7mPMdSRz4AAAAASUVORK5CYII=" >
        </img>
      </field>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

### Barcode example

This example shows how different types of bar code can be used.

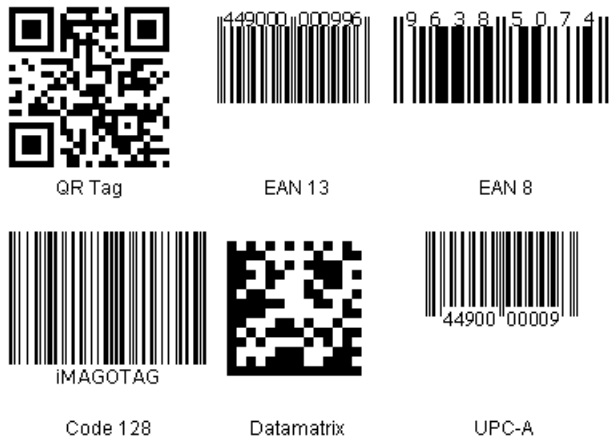



**Figure 21: Generated image**

- Template:

```
Barcode Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image height="300" width="400" >
      <field height="100" width="100" x="10" y="10" >
        <barcode autoscale="true" type="qr" >
          <xsl:value-of select="Barcodes/Qr" >
          </xsl:value-of>
        </barcode>
      </field>
      <field height="20" valign="bottom" width="100" x="10" y="110" >
        <label align="center" >
          QR Tag
        </label>
      </field>
      <field height="60" width="100" x="140" y="10" >
       <barcode fontName="Tahoma" humanReadableEnabled="true" humanReadablePlacement="top"
type="ean13" >
          <xsl:value-of select="Barcodes/Ean13" >
          </xsl:value-of>
        </barcode>
      </field>
      <field height="20" valign="bottom" width="100" x="140" y="110" >
        <label align="center" >
          EAN 13
        </label>
      </field>
      <field height="60" width="140" x="250" y="10" >
       <barcode fontName="Tahoma" humanReadableEnabled="true" humanReadablePlacement="top"
scale="2" type="ean8" >
          <xsl:value-of select="Barcodes/Ean8" >
          </xsl:value-of>
        </barcode>
      </field>
      <field height="20" valign="bottom" width="100" x="270" y="110" >
        <label align="center" >
          EAN 8
        </label>
```

```
      </field>
      <field height="100" width="125" x="10" y="150" >
        <barcode autoscale="true" type="code128" >
          <xsl:value-of select="Barcodes/Code128" >
          </xsl:value-of>
        </barcode>
      </field>
      <field height="20" valign="bottom" width="125" x="10" y="260" >
        <label align="center" >
          Code 128
        </label>
      </field>
      <field height="100" width="100" x="140" y="150" >
        <barcode autoscale="true" type="datamatrix" >
          <xsl:value-of select="Barcodes/DataMatrix" >
          </xsl:value-of>
        </barcode>
      </field>
      <field height="20" valign="bottom" width="100" x="140" y="260" >
        <label align="center" >
          Datamatrix
        </label>
      </field>
      <field height="60" width="100" x="270" y="150" >
        <barcode fontName="Times New Roman" humanReadableEnabled="true"
humanReadablePlacement="bottom" type="upc-a" >
          <xsl:value-of select="Barcodes/Upca" >
          </xsl:value-of>
        </barcode>
      </field>
      <field height="20" valign="bottom" width="100" x="270" y="260" >
        <label align="center" >
          UPC-A
        </label>
      </field>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
Barcode Example (Record)
<Barcodes>
  <Qr>
    iMAGOTAG
  </Qr>
  <Ean13>
    5449000000996
  </Ean13>
  <Ean8>
    96385074
  </Ean8>
  <Code128>
    iMAGOTAG
  </Code128>
  <DataMatrix>
    iMAGOTAG
  </DataMatrix>
  <Upca>
    544900000093
  </Upca>
</Barcodes>
```

### Span examples

This example shows how to use the span tags for text formatting, including compression, different font sizes and text decorations.

- Template:

```
Span Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image height="300" width="400" >
      <field height="71" width="380" x="10" y="10" >
        <text font-family="Times New Roman" font-size="30" >
          Text
          <span font-family="Tahoma" font-size="20" font-weight="bold"
text-decoration="overline" >
            Span
          </span>
          <span superscript="superscript" font-size="24" >
            super
          </span>
          <span font-size="24" >
            normal
          </span>
          <span font-size="24" superscript="subscript" >
            subscript
          </span>
        </text>
        <label font-family="Verdana" font-size="18" >
          Label with
          <span font-size="26" font-style="italic" text-decoration="underline" >
            Span
          </span>
          <span font-family="Arial" font-size="14" text-decoration="linethrough" >
            Span
          </span>
        </label>
      </field>
      <field height="60" width="380" x="10" y="90" >
        <text condense="1,0.8,0.6,0.4" >
          Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text Text
 Text
        </text>
        <label font-size="16,12" >
          Label Label Label Label Label Label Label Label Label
        </label>
        <label condense="1,0.8,0.6,0.4" font-size="16" >
          Label Label Label Label Label Label Label Label Label
        </label>
      </field>
      <field height="140" width="180" x="10" y="150" >
        <text condense="1,0.8,0.6,0.4" font-size="24" >
          <ul>
            <li>
              Text Text Text Text
            </li>
            <li condense="1" >
              Text Text Text Text
            </li>
            <li>
              Text Text Text Text
            </li>
            <li condense="1" font-size="24,22,20,18,16,14,12" >
              Text Text Text Text
            </li>
```

```
          </ul>
        </text>
      </field>
      <field height="140" width="190" x="200" y="150" >
        <text condense="1,0.8,0.6,0.4" font-size="24" >
          <ol>
            <li>
              Text Text Text Text
            </li>
            <li condense="1" >
              Text Text Text Text
            </li>
            <li>
              Text Text Text Text
            </li>
            <li condense="1" font-size="24,22,20,18,16,14,12" >
              Text Text Text Text
            </li>
          </ol>
        </text>
      </field>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

### CSS example

This example uses a CSS file to format and specify the template.

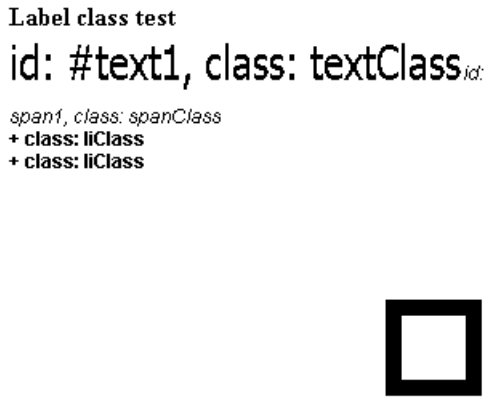



**Figure 22: Generated image**

- Template:

```
CSS Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image height="300" width="400" >
      <css href="../css/example.css" >
      </css>
      <field max-height="290" width="300" x="10" y="10" >
        <label class="labelClass" font-size="16" >
          Label class test
        </label>
        <text class="textClass" id="text1" >
          id: #text1, class: textClass
          <span class="spanClass" id="span1" >
```

```
            id: span1, class: spanClass
          </span>
          <ul class="listClass" id="ul1" >
            <li id="liClass" >
              class: liClass
            </li>
            <li id="liClass" >
              class: liClass
            </li>
          </ul>
        </text>
      </field>
      <rect class="rectClass" id="rect1" >
      </rect>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

### Table example

This example shows how a table can be used.

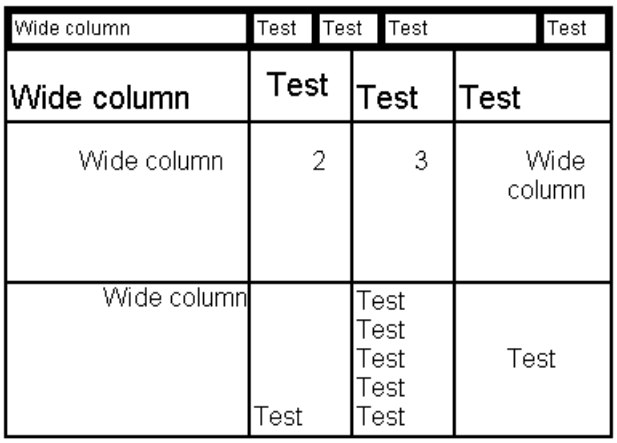



**Figure 23: Generated image**

- Template:

```
Table Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image height="300" width="400" >
      <field max-height="280" width="380" x="10" y="10" >
        <table table-border="2" header-border="5" header-row="true" valign="bottom" >
          <row>
            <cell table-width="40%" >
              <text>
                Wide column
              </text>
            </cell>
            <cell>
              <text>
                Test
              </text>
            </cell>
            <cell>
              <label>
                Test
```

```
            </label>
          </cell>
          <cell>
            <label>
              Test
            </label>
          </cell>
          <cell>
            <label>
              Test
            </label>
          </cell>
        </row>
        <row>
          <cell>
            <text font-size="20" >
              Wide column
            </text>
          </cell>
          <cell cellpadding="10" >
            <label font-size="20" >
              Test
            </label>
          </cell>
          <cell>
            <text font-size="20" >
              Test
            </text>
          </cell>
          <cell>
            <label font-size="20" >
              Test
            </label>
          </cell>
        </row>
        <row cellpadding="15" table-height="100px" valign="top" align="right" >
          <cell>
            <text font-size="16" >
              Wide column
            </text>
          </cell>
          <cell>
            <label font-size="16" >
              2
            </label>
          </cell>
          <cell>
            <text font-size="16" >
              3
            </text>
          </cell>
          <cell table-width="100px" >
            <text font-size="16" >
              Wide column
            </text>
          </cell>
        </row>
        <row>
          <cell align="right" cellpadding="0" valign="top" >
            <text font-size="16" >
              Wide column
            </text>
          </cell>
          <cell>
```

```
            <label font-size="16" >
              Test
            </label>
          </cell>
          <cell>
            <text font-size="16" >
              Test Test Test Test Test
            </text>
          </cell>
          <cell align="center" table-width="100px" valign="center" >
            <label font-size="16" >
              Test
            </label>
          </cell>
        </row>
      </table>
    </field>
  </image>
 </xsl:template>
</xsl:stylesheet>
```

### Rotation example

This example shows how the rotation attribute can be used.


**Figure 24: Generated image**

- Template:

```
Rotation Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image height="300" width="400" >
      <field height="80" rotation="-45" width="80" x="0" y="0" >
        <label font-size="20" font-weight="bold" >
          Sale!
        </label>
      </field>
      <field height="100" rotation="-90" width="100" x="150" y="0" >
        <barcode type="ean13" >
          5449000000996
        </barcode>
      </field>
      <field height="80" rotation="180" width="200" x="0" y="100" >
        <label font-size="16" font-weight="bold" >
          Test Test Test Test Test
```

```
        </label>
      </field>
      <field height="50" rotation="270" width="100" x="300" y="0" >
        <label font-size="16" font-weight="bold" >
          Test Test
        </label>
      </field>
      <field height="70" rotation="-10" width="200" x="100" y="190" >
        <img autoscale="true" src="../images/imagotag.png" >
        </img>
      </field>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

## 5.3 7.4"

### Advanced drawing example

This example shows the use of drawing elements like triangles, ellipses, and polygons.

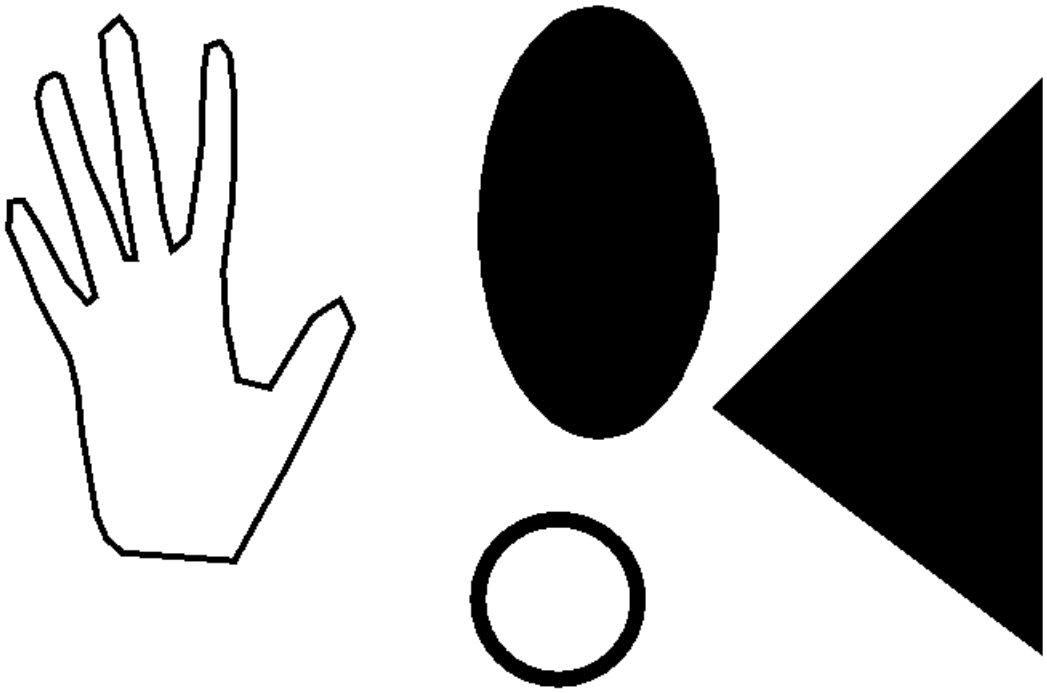


**Figure 25: Generated image**

- Template:

```
Advanced Drawing Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image height="800" rotation="270" width="480" >
      <polygon border="4" fill="false" points="104,248 160,279 208,303 250,322 267,314
256,297 235,284 212,270 217,250 251,243 285,241 332,247 367,248 392,248 420,245 428,239
```

```
425,231 411,229 385,228 368,228 339,224 306,219 298,209 321,203 357,198 387,191 431,185
443,176 433,165 409,167 372,174 344,177 293,186 293,180 321,170 351,157 406,140 408,136
404,128 393,125 377,128 328,144 311,149 269,161 265,156 279,144 313,126 329,116 328,108
311,107 286,118 268,125 231,145 210,150 182,153 160,157 132,162 118,168 109,178" >
      </polygon>
      <triangle border="5" x1="50" x2="200" x3="400" y1="750" y2="550" y3="750" >
      </triangle>
      <ellipse height="150" width="270" x="180" y="400" >
      </ellipse>
      <ellipse border="10" fill="false" height="100" width="100" x="30" y="400" >
      </ellipse>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

### Rounded rectangle/field example

This example shows the use of rounded corners on rectangles and text fields.

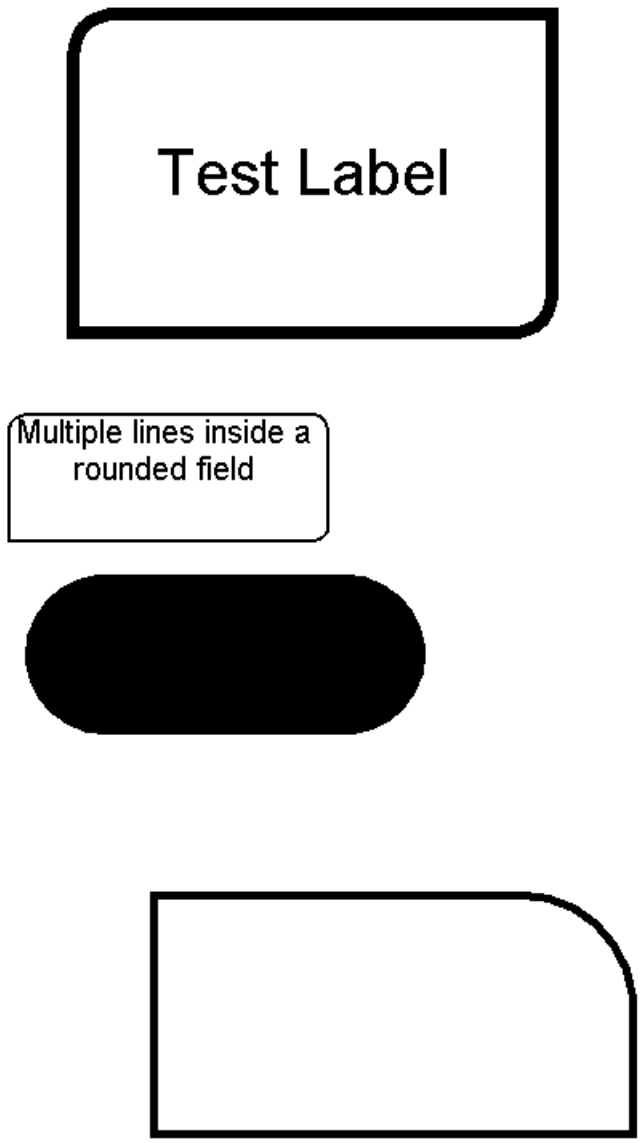



**Figure 26: Generated image**

- Template:

```
Rounded Rectangle/Field Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <image height="800" width="480" >
      <field border="8" corner-radius="25" height="200"
rounded-corners="TOP_LEFT,BOTTOM_RIGHT" valign="center" width="300" x="50" y="50" >
        <label align="center" font-size="40" >
          Test Label
        </label>
```

```
      </field>
      <field border="2" corner-radius="10" height="80"
rounded-corners="TOP_RIGHT,TOP_LEFT,BOTTOM_RIGHT" width="200" x="10" y="300" >
        <text align="center" font-size="20" >
          Multiple lines inside a rounded field
        </text>
      </field>
      <rect corner-radius="50" height="100"
rounded-corners="TOP_RIGHT,TOP_LEFT,BOTTOM_RIGHT,BOTTOM_LEFT" width="250" x="20" y="400"
>
      </rect>
      <rect border="5" corner-radius="70" fill="false" height="150"
rounded-corners="TOP_RIGHT" width="300" x="100" y="600" >
      </rect>
    </image>
  </xsl:template>
</xsl:stylesheet>
```

### Dynamic template based on a default graphic – 7.4"

This example is based on a dynamic template and the Coca Cola example. The image size depends on the type of display.


**Figure 27: Generated image**

- Template:

```
Dynamic template using default record - 7.4" (Template)
<xsl:stylesheet version="1.0" >
  <xsl:template match="Record" >
    <xsl:choose>
      <xsl:when test="Label/DisplayHeight = 176 and Label/DisplayWidth = 264" >
        <image height="176" width="264" >
          <line thickness="2" x-from="0" x-to="264" y-from="126" y-to="126" >
          </line>
          <field height="36" width="245" x="9" y="10" >
```

```
            <text font-family="Tahoma" font-size="14" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="83" width="190" x="60" y="47" >
            <text align="right" font-family="Impact" font-size="68" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="180" y="131" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="180" y="146" >
            <text align="right" font-family="Tahoma" font-size="20" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="136" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="158" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 300 and Label/DisplayWidth = 400" >
        <image height="300" width="400" >
          <line thickness="2" x-from="0" x-to="400" y-from="236" y-to="236" >
          </line>
          <field height="90" width="382" x="9" y="10" >
            <text font-family="Tahoma" font-size="24" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="18" >
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
```

```
              </xsl:value-of>
            </text>
          </field>
          <field height="122" width="350" x="40" y="100" >
            <text align="right" font-family="Impact" font-size="100" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="320" y="248" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="320" y="266" >
            <text align="right" font-family="Tahoma" font-size="20" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="260" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="282" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 800 and Label/DisplayWidth = 480" >
        <image height="800" width="480" >
          <line thickness="2" x-from="0" x-to="480" y-from="700" y-to="700" >
          </line>
          <field height="400" width="460" x="10" y="10" >
            <text font-family="Tahoma" font-size="48" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="28" >
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="200" width="460" x="10" y="480" >
            <text align="right" font-family="Impact" font-size="120" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
```

```
          <field height="32" width="100" x="370" y="728" >
            <text align="right" font-family="Verdana" font-size="26" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="32" width="100" x="370" y="760" >
            <text align="right" font-family="Tahoma" font-size="26" >
             <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="758" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="780" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
Dynamic template using default record - 7.4" (Record)
<article articleNumber="5449000000996" >
  <field key="pricePerUnit" value="1.65" >
  </field>
  <field key="unit" value="1 l" >
  </field>
  <field key="price" value="0.55" >
  </field>
  <field key="description" value="Can 0.33 l" >
  </field>
  <field key="name" value="Coca Cola" >
  </field>
</article>
```

### Dynamic template based on a default graphic – 7.4"

This example is based on a dynamic template and the chili con carne example. The image size depends on the type of display.


**Figure 28: Generated image**

- Template:

```
Dynamic template using default record - 7.4" (Template)
<xsl:stylesheet version="1.0" >
  <xsl:template match="Record" >
    <xsl:choose>
      <xsl:when test="Label/DisplayHeight = 176 and Label/DisplayWidth = 264" >
        <image height="176" width="264" >
          <line thickness="2" x-from="0" x-to="264" y-from="126" y-to="126" >
          </line>
```

```
          <field height="36" width="245" x="9" y="10" >
            <text font-family="Tahoma" font-size="14" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="83" width="190" x="60" y="47" >
            <text align="right" font-family="Impact" font-size="68" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="180" y="131" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="180" y="146" >
            <text align="right" font-family="Tahoma" font-size="20" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="136" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="158" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 300 and Label/DisplayWidth = 400" >
        <image height="300" width="400" >
          <line thickness="2" x-from="0" x-to="400" y-from="236" y-to="236" >
          </line>
          <field height="90" width="382" x="9" y="10" >
            <text font-family="Tahoma" font-size="24" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="18" >
              <br>
              </br>
```

```
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="122" width="350" x="40" y="100" >
            <text align="right" font-family="Impact" font-size="100" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="320" y="248" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="320" y="266" >
            <text align="right" font-family="Tahoma" font-size="20" >
             <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="260" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="282" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 800 and Label/DisplayWidth = 480" >
        <image height="800" width="480" >
          <line thickness="2" x-from="0" x-to="480" y-from="700" y-to="700" >
          </line>
          <field height="400" width="460" x="10" y="10" >
            <text font-family="Tahoma" font-size="48" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="28" >
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="200" width="460" x="10" y="480" >
            <text align="right" font-family="Impact" font-size="120" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
```

```
          </field>
          <field height="32" width="100" x="370" y="728" >
            <text align="right" font-family="Verdana" font-size="26" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="32" width="100" x="370" y="760" >
            <text align="right" font-family="Tahoma" font-size="26" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="758" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="780" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
Dynamic template using default record - 7.4" (Record)
<article articleNumber="9000275639319" >
  <field key="pricePerUnit" value="4.98" >
  </field>
  <field key="unit" value="1 kg" >
  </field>
  <field key="price" value="2.49" >
  </field>
  <field key="description" value="Can 500 g" >
  </field>
  <field key="name" value="Chili con Carne" >
  </field>
</article>
```

### Dynamic template based on a default graphic – 7.4"

This example is based on a dynamic template and the Red Bull example. The image size depends on the type of display.


**Figure 29: Generated image**

- Template:

```
Dynamic template using default record - 7.4" (Template)
<xsl:stylesheet version="1.0" >
  <xsl:template match="Record" >
    <xsl:choose>
      <xsl:when test="Label/DisplayHeight = 176 and Label/DisplayWidth = 264" >
        <image height="176" width="264" >
          <line thickness="2" x-from="0" x-to="264" y-from="126" y-to="126" >
          </line>
          <field height="36" width="245" x="9" y="10" >
```

```
            <text font-family="Tahoma" font-size="14" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="83" width="190" x="60" y="47" >
            <text align="right" font-family="Impact" font-size="68" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="180" y="131" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="180" y="146" >
            <text align="right" font-family="Tahoma" font-size="20" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="136" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="158" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 300 and Label/DisplayWidth = 400" >
        <image height="300" width="400" >
          <line thickness="2" x-from="0" x-to="400" y-from="236" y-to="236" >
          </line>
          <field height="90" width="382" x="9" y="10" >
            <text font-family="Tahoma" font-size="24" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="18" >
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
```

```
              </xsl:value-of>
            </text>
          </field>
          <field height="122" width="350" x="40" y="100" >
            <text align="right" font-family="Impact" font-size="100" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="320" y="248" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="320" y="266" >
            <text align="right" font-family="Tahoma" font-size="20" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="260" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="282" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 800 and Label/DisplayWidth = 480" >
        <image height="800" width="480" >
          <line thickness="2" x-from="0" x-to="480" y-from="700" y-to="700" >
          </line>
          <field height="400" width="460" x="10" y="10" >
            <text font-family="Tahoma" font-size="48" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="28" >
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="200" width="460" x="10" y="480" >
            <text align="right" font-family="Impact" font-size="120" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
```

```
          <field height="32" width="100" x="370" y="728" >
            <text align="right" font-family="Verdana" font-size="26" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="32" width="100" x="370" y="760" >
            <text align="right" font-family="Tahoma" font-size="26" >
             <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="758" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="780" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
Dynamic template using default record - 7.4" (Record)
<article articleNumber="9002490100070" >
  <field key="pricePerUnit" value="5.56" >
  </field>
  <field key="unit" value="1 l" >
  </field>
  <field key="price" value="1.39" >
  </field>
  <field key="description" value="Can 250 ml" >
  </field>
  <field key="name" value="Red Bull" >
  </field>
  <field key="sale" value="1" >
  </field>
</article>
```

**Dynamic template based on a default graphic – 7.4"**

This example is based on a dynamic template and the Vöslauer example. The image size depends on the type of display.


**Figure 30: Generated image**

- Template:

```
Dynamic template using default record - 7.4" (Template)
<xsl:stylesheet version="1.0" >
  <xsl:template match="Record" >
    <xsl:choose>
      <xsl:when test="Label/DisplayHeight = 176 and Label/DisplayWidth = 264" >
        <image height="176" width="264" >
          <line thickness="2" x-from="0" x-to="264" y-from="126" y-to="126" >
          </line>
          <field height="36" width="245" x="9" y="10" >
```

```
            <text font-family="Tahoma" font-size="14" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="83" width="190" x="60" y="47" >
            <text align="right" font-family="Impact" font-size="68" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="180" y="131" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="180" y="146" >
            <text align="right" font-family="Tahoma" font-size="20" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="136" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="158" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 300 and Label/DisplayWidth = 400" >
        <image height="300" width="400" >
          <line thickness="2" x-from="0" x-to="400" y-from="236" y-to="236" >
          </line>
          <field height="90" width="382" x="9" y="10" >
            <text font-family="Tahoma" font-size="24" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="18" >
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
```

```
              </xsl:value-of>
            </text>
          </field>
          <field height="122" width="350" x="40" y="100" >
            <text align="right" font-family="Impact" font-size="100" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
          <field height="14" width="70" x="320" y="248" >
            <text align="right" font-family="Verdana" font-size="11" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="24" width="70" x="320" y="266" >
            <text align="right" font-family="Tahoma" font-size="20" >
              <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="260" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="282" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
      <xsl:when test="Label/DisplayHeight = 800 and Label/DisplayWidth = 480" >
        <image height="800" width="480" >
          <line thickness="2" x-from="0" x-to="480" y-from="700" y-to="700" >
          </line>
          <field height="400" width="460" x="10" y="10" >
            <text font-family="Tahoma" font-size="48" >
              <span font-weight="bold" >
                <utils method="toUpperCase" >
                  <xsl:value-of select="article/field[@key='name']/@value" >
                  </xsl:value-of>
                </utils>
              </span>
            </text>
            <text font-family="Tahoma" font-size="28" >
              <br>
              </br>
              <xsl:value-of select="article/field[@key='description']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="200" width="460" x="10" y="480" >
            <text align="right" font-family="Impact" font-size="120" >
              <xsl:value-of select="translate(article/field[@key='price']/@value, '.',
','')" >
              </xsl:value-of>
            </text>
          </field>
```

```
          <field height="32" width="100" x="370" y="728" >
            <text align="right" font-family="Verdana" font-size="26" >
              <xsl:value-of select="article/field[@key='unit']/@value" >
              </xsl:value-of>
            </text>
          </field>
          <field height="32" width="100" x="370" y="760" >
            <text align="right" font-family="Tahoma" font-size="26" >
             <xsl:value-of select="translate(article/field[@key='pricePerUnit']/@value,
'.', ',')" >
              </xsl:value-of>
            </text>
          </field>
          <field align="center" height="20" width="140" x="10" y="758" >
            <barcode autoscale="false" humanReadableEnabled="false" type="ean13" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </barcode>
          </field>
          <field height="12" width="140" x="10" y="780" >
            <text align="center" font-family="Tahoma" font-size="10" >
              <xsl:value-of select="article/@articleNumber" >
              </xsl:value-of>
            </text>
          </field>
        </image>
      </xsl:when>
    </xsl:choose>
  </xsl:template>
</xsl:stylesheet>
```

- Example:

```
Dynamic template using default record - 7.4" (Record)
<article articleNumber="9009700145104" >
  <field key="pricePerUnit" value="0.35" >
  </field>
  <field key="unit" value="1 l" >
  </field>
  <field key="price" value="0.35" >
  </field>
  <field key="description" value="Vöslauer sparkling, 1 l PET" >
  </field>
  <field key="name" value="Vöslauer mineral water" >
  </field>
</article>
```

**Customized graphic 7.4"**

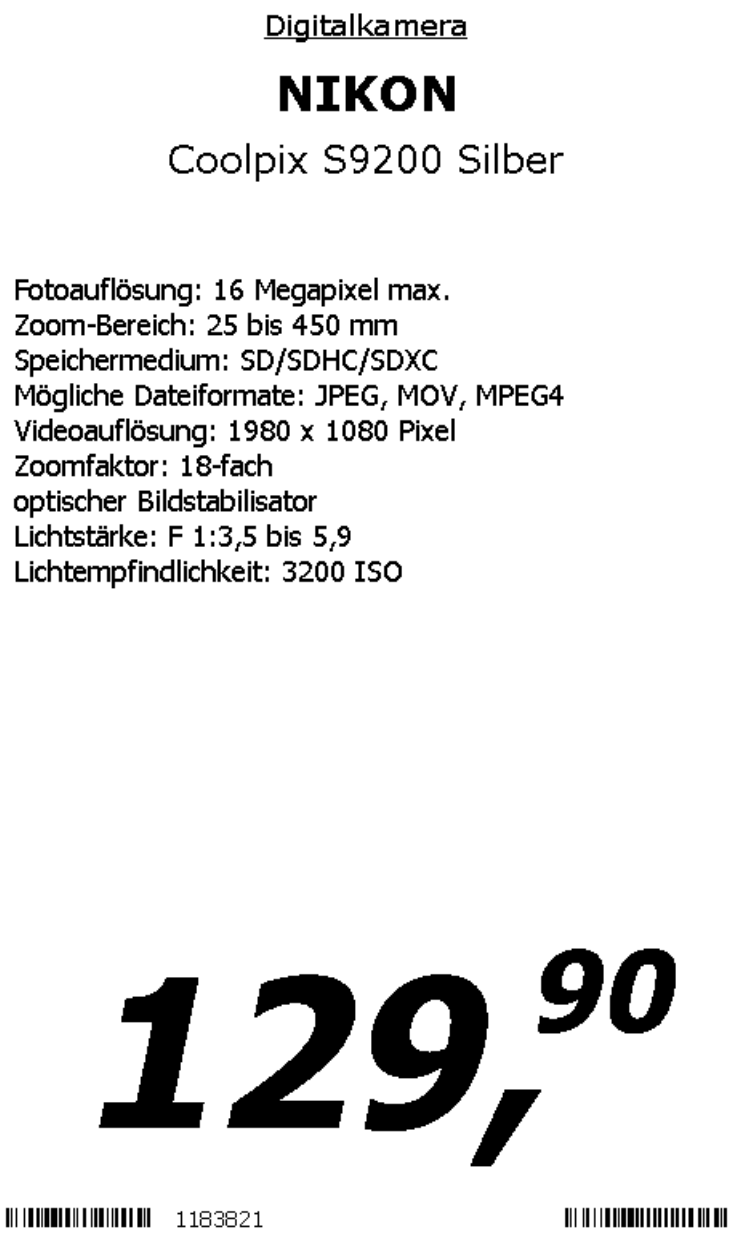



**Figure 31: Generated image**

- Template:

```
Custom Record 7.4" Example (Template)
<xsl:stylesheet version="2.0" >
  <xsl:template match="Record" >
    <xsl:variable name="price" >
      <xsl:value-of select="translate(Article/Price,'.','')" >
      </xsl:value-of>
    </xsl:variable>
    <xsl:variable name="maxProperties" select="8" >
    </xsl:variable>
    <xsl:variable name="maxSpecifications" select="7" >
    </xsl:variable>
```

```
   <image height="800" width="480" >
     <field height="22" width="480" x="0" y="30" >
      <text align="center" font-family="Verdana" font-size="18" text-decoration="underline"
>
        <xsl:value-of select="Article/Category" >
        </xsl:value-of>
      </text>
     </field>
     <field height="36" width="480" x="0" y="65" >
       <text align="center" font-family="Verdana" font-size="30" font-weight="bold" >
         <utils method="toUpperCase" >
           <xsl:value-of select="Article/Manufacturer" >
           </xsl:value-of>
         </utils>
       </text>
     </field>
     <field height="60" width="450" x="15" y="110" >
       <text align="center" font-family="Verdana" font-size="24" >
         <xsl:value-of select="Article/Name" >
         </xsl:value-of>
       </text>
     </field>
     <field height="400" width="450" x="15" y="175" >
       <text align="center" font-family="Tahoma" font-size="15" font-weight="bold" >
         <ul spacing="0" type="" >
           <xsl:for-each select="Article/Properties/Field" >
             <li>
               <xsl:value-of select="." >
               </xsl:value-of>
             </li>
           </xsl:for-each>
         </ul>
       </text>
       <text align="left" font-family="Tahoma" font-size="18" font-weight="normal"
padding-left="5" padding-top="20" >
         <ul spacing="0" type="" >
           <xsl:for-each select="Article/Specification/Field" >
             <li>
               <xsl:value-of select="." >
               </xsl:value-of>
             </li>
           </xsl:for-each>
         </ul>
       </text>
     </field>
     <xsl:choose>
       <xsl:when test="number(substring-before($price, ',')) < 100" >
         <xsl:choose>
           <xsl:when test="substring-after($price,',') = '00'" >
             <field height="205" width="410" x="10" y="570" >
             <text align="right" font-family="Tahoma" font-size="170" font-style="italic"
 font-weight="bold" >
                 <xsl:value-of select="substring-before($price,',')" >
                 </xsl:value-of>
               </text>
             </field>
           </xsl:when>
           <xsl:otherwise>
             <field height="205" width="330" x="10" y="570" >
             <text align="right" font-family="Tahoma" font-size="170" font-style="italic"
 font-weight="bold" >
                 <xsl:value-of select="substring-before($price,',')" >
                 </xsl:value-of>
               </text>
```

```
              </field>
              <field height="157" width="120" x="320" y="570" >
                <text font-family="Tahoma" font-size="85" font-style="italic"
font-weight="bold" >
                  <xsl:value-of select="substring-after($price,',')" >
                  </xsl:value-of>
                </text>
              </field>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:when>
        <xsl:when test="number(substring-before($price, ',')) < 1000" >
          <xsl:choose>
            <xsl:when test="substring-after($price,',') = '00'" >
              <field height="157" width="415" x="10" y="600" >
              <text align="right" font-family="Tahoma" font-size="130" font-style="italic"
 font-weight="bold" >
                  <xsl:value-of select="substring-before($price,',')" >
                  </xsl:value-of>
                </text>
              </field>
            </xsl:when>
            <xsl:otherwise>
              <field height="157" width="340" x="10" y="600" >
              <text align="right" font-family="Tahoma" font-size="130" font-style="italic"
 font-weight="bold" >
                  <xsl:value-of select="substring-before($price,',')" >
                  </xsl:value-of>
                </text>
              </field>
              <field height="157" width="100" x="340" y="600" >
                <text font-family="Tahoma" font-size="70" font-style="italic"
font-weight="bold" >
                  <xsl:value-of select="substring-after($price,',')" >
                  </xsl:value-of>
                </text>
              </field>
            </xsl:otherwise>
          </xsl:choose>
        </xsl:when>
        <xsl:when test="number(substring-before($price, ',')) < 10000" >
          <xsl:choose>
            <xsl:when test="substring-after($price,',') = '00'" >
              <field height="157" width="435" x="10" y="600" >
              <text align="right" font-family="Tahoma" font-size="110" font-style="italic"
 font-weight="bold" >
                  <xsl:value-of select="substring-before($price,',')" >
                  </xsl:value-of>
                </text>
              </field>
            </xsl:when>
            <xsl:otherwise>
              <field height="157" width="370" x="00" y="600" >
              <text align="right" font-family="Tahoma" font-size="110" font-style="italic"
 font-weight="bold" >
                  <xsl:value-of select="substring-before($price,',')" >
                  </xsl:value-of>
                </text>
              </field>
              <field height="157" width="100" x="360" y="600" >
                <text font-family="Tahoma" font-size="55" font-style="italic"
font-weight="bold" >
                  <xsl:value-of select="substring-after($price,',')" >
                  </xsl:value-of>
```

```
              </text>
            </field>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:when>
    </xsl:choose>
    <field align="left" height="12" width="100" x="15" y="780" >
      <barcode humanReadableEnabled="false" type="code128" >
        <xsl:value-of select="Article/ArticleNumber" >
        </xsl:value-of>
      </barcode>
    </field>
    <field height="17" width="80" x="120" y="778" >
      <text font-family="Tahoma" font-size="14" >
        <xsl:value-of select="Article/ArticleNumber" >
        </xsl:value-of>
      </text>
    </field>
    <field align="right" height="12" width="150" x="315" y="780" >
      <barcode humanReadableEnabled="false" type="code128" >
        <xsl:value-of select="Label/Id" >
        </xsl:value-of>
      </barcode>
    </field>
  </image>
</xsl:template>
</xsl:stylesheet>
```

- Example:

```
Custom Record 7.4" Example (Record)
<Article>
  <ArticleNumber>
    1183821
  </ArticleNumber>
  <Name>
    Coolpix S9200 Silber
  </Name>
  <Manufacturer>
    Nikon
  </Manufacturer>
  <Category>
    Digital camera
  </Category>
  <Price>
    129,90
  </Price>
  <Specification>
    <Field>
      Photo resolution: 16 Megapixel max.
    </Field>
    <Field>
      Zoom range: 25 to 450 mm
    </Field>
    <Field>
      Storage medium: SD/SDHC/SDXC
    </Field>
    <Field>
      Available file formats: JPEG, MOV, MPEG4
    </Field>
    <Field>
      Video resolution: 1980 x 1080 pixels
    </Field>
    <Field>
      Zoom factor: 18x
```

```
    </Field>
    <Field>
      Optical image stabilizer
    </Field>
    <Field>
      Rel. aperture: F 1:3,5 to 5,9
    </Field>
    <Field>
      Film speed: 3200 ISO
    </Field>
  </Specification>
  <Packaging>
    <Item>
      Carrying strap
    </Item>
    <Item>
      USB cable
    </Item>
    <Item>
      Power adapter
    </Item>
  </Packaging>
  <Accessories>
    <Item>
      Rechargeable battery EN-EL12
    </Item>
    <Item>
      SDHC card
    </Item>
  </Accessories>
</Article>
```