XML fileformat (18/01/2018)

The internal laser firmware accepts a xml file format that defines the message's contents that is to be marked. Any external device or software can , in principle, use this kind of file format to produce a valid message readable by the laser's firmware and/or by the PC laser software.

File structure

General structure of a xml file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<laserfile version="0x1">
  <layers>
    <layer name="New Layer" id="0" printable="1" editable="1" visible="1" power="100.000000"
speed="1000000" resolution="10" frequency="50.000000" zpos="0" zdefocus="0" color="0xff0000"
mask="0x0" delay="0" repeat="0" signalmask="0x0" signalstatus="0x0" scannerset="0" />
  </layers>
  <objects>
    <line sx="12900" sy="23800" ex="16300" ey="19800" id="0">
      <generic layer_id="0" printable="1" editable="1" linewidth="0" />
    </line>
    <rectangle x="22900" y="19800" x2="22900" y2="22600" x3="27300" y3="22600" x4="27300"
y4="19800" id="1">
      <generic layer_id="0" printable="1" editable="1" linewidth="0" />
    </rectangle>
    <polyline type="closed" points=" 34300 25600, 35500 22500, 41500 21900, 43000 24900, 40300
26000, 34300 25600" id="2">
      <generic layer_id="0" printable="1" editable="1" linewidth="0" />
    </polyline>
    <polyline type="open" points=" 48700 23500, 51700 20800, 55100 20300, 57800 21600, 57200 22700,
53800 24600" id="3">
      <generic layer_id="0" printable="1" editable="1" linewidth="0" />
    </polyline>
    <arc cx="12458" cy="34282" smallaxis="2293" largeaxis="2293" startangle="0.890469"
endangle="4.131884" flip="0" id="4">
      <generic layer_id="0" printable="1" editable="1" linewidth="0" />
    </arc>
    <ellipse cx="21350" cy="34250" rx="2150" ry="1750" id="5">
      <generic layer_id="0" printable="1" editable="1" linewidth="0" />
    </ellipse>
....
....
  </objects>
</laserfile>
```

Details:

<?xml version="1.0" encoding="UTF-8" ?> Generic xml header. Actually, the only supported encoding type is UTF-8.

Tags:

In general any tag may have attributes or embedded tags/entities. Not all of the attributes of the elements must be present in the xml file unless the value of the attributes differs from its default value. Thus, in general, the xml file can be very tiny and small as long as only the necessary attributes are explicitly given and set inside the xml file. In the following detailed description mandatory attributes or entities are explicitly named as mandatory.
All other attributes or entities must not be necessarily present as long as their value is equal to the default value.

<laserfile version="0x1">

The laserfile entity is mandatory. All components within the laserfile entity compose the message.

Attributes :

- version: <numerical value> in hexadecimal. Default value is 0x1. Actually, the version value is not read by the xml parser
- delay: <numerical value> . Default value is 0. When printing mode is static the 'delay' attribute value represents a time delay in milliseconds that is applied before the message is printed. When printing mode is any of the dynamic modes the 'delay' attribute value defines an offset in ideal coordinates along the moving direction (1 unit equals 1/100 000 of the scanfield size). A negative offset shifts the message towards the moving direction, a positive value shifts it to the opposite direction.
- comment: <string>. Is just used for storing and comment that the application would like to store within this file.
- config: <string>. The string can contain one or more of the following parameters as described in the documentation of the configuration file. These configuration parameters are valid while the file is loaded (is the default printing file). With this string you can change certain configuration parameters without uploading a new configuration file. Each <parameter> <value> pair stored within this string must be seperated by a linefeed (LF, hex 0x0a) ! The actual parameters that can be set within the config string are (see the documentation of the configuration file):
    - Dynamicmode
    - Autodistance
    - Velocity

    - Sortobjects
    - CYLINDER_RADIAN
    - CYLINDER_A0
    - CYLINDER_ZDIST
    - CYLINDER_INVERT
    - AUTOBORDER
    - LEFTBORDER
    - RIGHTBORDER
    - SERVOBORDERLEFT
    - SERVOBORDERRIGHT

  One additional parameter can be set within this string that is not part of the configuration file parameters:

    - SERVOSCALE: <value> : <value> can be a value from 0.0 to 1.0. This parameter scales the

SERVOMAX value of the configuration file, which represents the max. DAC value for the servomotor control. A value of 1.0 would set the max. DAC value to half of the configuration settings thus reducing the max. velocity of the servomotor to half of its default settings. This parameter is very helpful to reduce/increment the max. speed of the servomotor depending on the marking design.

- axisflags: <hexadecimal> [0x0..0xffffff] default="0x00000000": According the bits set in this variable the laser sends out an axis command to the optional XYZ-motor when the file is loaded or going to be printed.
    - 0x00000001 the X axis flag (move the x axis according 'xaxis')
    - 0x00000002 the Y axis flag (move the y axis according 'yaxis')
    - 0x00000004 the Z axis flag (move the z axis according 'zaxis')

    - 0x00000010 the X axis home flag (move to the home position of the x axis)
    - 0x00000020 the Y axis home flag (move to the home position of the y axis)
    - 0x00000040 the Z axis home flag (move to the home position of the z axis)

- xaxis:<float>: the x position to move to in units of [mm].
- yaxis:<float>: the y position to move to in units of [mm].
- zaxis:<float>: the z position to move to in units of [mm].
- zpos: <integer>[-50000..+50000] default="0". The z-position of the optional z-axis in ideal units with respect to the nominal focal plane. Objects in the layer are shifted in their z-position and are placed in a plane at z = zpos. If the system is equipped with a focus-shifter, the focus will be adjusted to this new plane.
    - note that the layer's 'zpos' value will be added to this value.

- zdefocus: <integer>[-50000,+50000] default="0". The z-defocus value in ideal units shifts the focus of the laser spot in z-direction.
    - note that the layer's 'zdefocus' value will be added to this value.

```
<layers>

    ...
</layers>
```

The layers entity is mandatory. It contains for each used layer an additional layer entity.

```
<layer name="New Layer" id="0" printable="1" editable="1" visible="1" power="100.000000"
speed="1000000" resolution="10" frequency="50.000000" zpos="0" zdefocus="0" color="0xff0000"
mask="0x0" delay="0" repeat="0" signalmask="0x0" signalstatus="0x0" scannerset="0" />
```

At least one layer entity is mandatory in the xml file. The layer defines some main parameters used for the marking process of the objects. Each object to be marked is assigned a layer, usually different objects are assigned to the same layer. Parameters like marking speed and others are set to values corresponding to the layer to which the object is assigned just before the object is being marked. Note that the order of the layers (occurrence within the xml-file) and it's ID (consecutive number starting from 0 – 255) is important as it determines the marking sequence.

'Command layer': A command layer is a special type of layer entity. A layer becomes a command layer when the 'command layer'-flag of the layer's mask is set. Command layers may not contain any figures and only some attributes are used for them. Attributes that do not apply to command layer are simply ignored. The command layer is simply used for introducing xyz-axis movements of an optional stepper motor axis. Also, the 'delay' parameter and the 'signalmask' , 'signalstate' parameters apply to a command layer.

'Pixmap layer': A pixmap layer is a layer that has the pixmap-attribute set to 1. In this case also a pixmap entity has to be stored inside the layer entity. A pixmap is a kind of bitmap to where the underlying objects are rendered. the pixmap entity is then printed as if it was an image entity. Only image entities and any text entities with a fonttype = '2', '5' or '6' can be rendered into a pixmap.

Attributes :

- name: <string> default="". The name of the layer. Layer have a name-attribute that may be used for referencing when the eventhandler is used for script programming.
- id: <integer> [0,...255] default="0", mandatory. The id of the layer must be unic for each layer. The objects defined within the xml file has also an id attribute that assigns the layer to this object. The layer's id must be a consecutive number for each used layer, starting with '0' and the layers must be stored inside the xml file in an ordered manner, as the order determines the marking sequence in static printing mode.
- printable: <integer> [0..1] default="1", mandatory. A value of "0" indicates that this layer and all objects assigned to this layer will not be printed. A value of "1" indicates that the objects of this layer will be printed.
- editable: <integer> [0..1] default="1". A value of "0" indicates that the layer's objects are not editable when loaded into the PC laser marking software. A value of "1" indicates that the objects are editable.
- visible: <integer> [0..1] default="1". A value of "0" indicates that the layer's objects will not be displayed in the PC laser marking software. A value of "1" makes the objects visible.
- power: <float> [0..100] default="100", mandatory. The laser power used to mark the objects of this layer. Units are in percent of the maximum laser power.
- speed: <integer>[10...60000000] default="1000000", mandatory. The vectorial marking speed used to mark the objects of this layer. The units are ideal-units per second, where the total scanfield size is 100 000 ideal units wide in x and y.
- resolution: <integer>[1...40] default = "10", mandatory. The resolution of vectorial marking to mark objects of this layer. Units are points per 1000 ideal units. Each vector is split into microvectors whose length corresponds to the resolution. A resolution of "10" means, that a line of length 1000 ideal units is split into 10 microvectors, or, in other words, the length of a microvector is 1000 ideal units devided by the resolution.
- frequency: <float>[5...200] default = "50.0", mandatory. The modulation frequency in kHz of the laser signal (pulse frequency for YAG or fiber lasers, duty cycle frequency for CO2 lasers).
- delay:<integer>[0...3000]. Defines a delay in milliseconds. The marking process is stopped during this time. The maximum allowed value depends on the system's scanner card.
- zpos: <integer>[-50000..+50000] default="0". The z-position of the optional z-axis in ideal units with respect to the nominal focal plane. Objects in the layer are shifted in their z-position and are placed in a plane at z = zpos. If the system is equipped with a focus-shifter, the focus will be adjusted to this new plane.
- zdefocus: <integer>[-50000,+50000] default="0". The zdefocus value in ideal units shifts the focus of the laser spot in z-direction.
- color: <hexadecimal> [0x0..0xffffff] default="0x0000ff". The RGB-color of the layer. The RGB-color is used for drawing the vectors of the layer's objects in the PC marking software (0xrrggbb).
- mask:<hexadecimal>[0x0...0xffffffff] default ="0x0". With the mask parameter you can bit-wise turn on/off some options for the layer.
  - 0x0004 Turns on the use of the signalmask of the layer. With the signalmask and signalstatus you can control with external signals if the objects of the layer should be printed or not.
  - 0x0002 Disables power scaling of the objects of this layer. With the TCP/IP and/or RS232 protocol the power settings can be globally scaled. When setting the power scale bit of the layer is enabled the scaling of the power for objects of this layer is disabled.

    - 0x0001 Disables speed scaling of the objects of this layer. With the TCP/IP and/or RS232 protocol the speed settings can be globally scaled. When setting the speed scale bit of the layer is enabled the scaling of the speed for objects of this layer is disabled.
    - 0x0010 The command flag. When this flag is enabled the layer is simply treated as a command layer. A command layer is a special layer that may not contain any figures. It is just used for 'delay' and 'axis' commands. The only used attributes of a command layer are 'id' , 'name', 'delay', 'axisflags', 'xaxis', 'yaxis', 'zaxis', 'signalmask' and 'signalstatus'.
- delay:<integer> [0..1000] default="0". A delay in milliseconds applied after settings internally all power, speed and frequency values of the layer. The marking process is suspended during the delay time.
- repeat:<integer>[0..1000] default="0". Defines how many times the marking of the objects of this layer is to be repeated.
- signalmask:<hexadecimal> default="0x0". The signalmask is used to mask out the input bits that determine if the layer is to be printed or not. The mask is 32-bit wide with the following mapping:
    - Bitmask 32-bit: bit31,bit30,.....,photocell2(bit18), photocell(bit17), PLC input(bit16),bit15, bit,14,........bit9,bit8,external selection7,..external selection0.
- signalstatus:<hexadecimal> default="0x0". The signalstatus is used to define the input bits state that determine if the layer is to be printed or not. This value is masked with the statusmask
    - <result> = <signalstatus> & <signalmask>. The result is compared with the real masked input state taken when the system is triggered (<real masked state> = <real input state> & <signalmask>). If <result> is equal to <real masked state> then the layer is printed, else it is not printed.
- scannerset:<integer> [0..3] default="0". Defines which scannerset parameters should be used for the marking of the objects of this layer. Scannersets and its parameters are defined in the configuration of the laser.

- axisflags: <hexadecimal> [0x0..0xffffff] default="0x00000000": According the bits set in this variable the laser sends out an axis command to the optional XYZ-stepper motor. This attribute is only used for command layers.
    - 0x00000001 the X axis flag (move the x axis according 'xaxis')
    - 0x00000002 the Y axis flag (move the y axis according 'yaxis')
    - 0x00000004 the Z axis flag (move the z axis according 'zaxis')

    - 0x00000010 the X axis home flag (move to the home position of the x axis)
    - 0x00000020 the Y axis home flag (move to the home position of the y axis)
    - 0x00000040 the Z axis home flag (move to the home position of the z axis)

- xaxis:<float>: the x position to move to in units of [mm]. Only used for command layers.
- yaxis:<float>: the y position to move to in units of [mm]. Only used for command layers.
- zaxis:<float>: the z position to move to in units of [mm]. Only used for command layers.
- pixmap:<integer>[0,1] default="0": when set to '1' then this layer becomes a pixmap layer and objects are rendered inside the pixmap. In this case a pixmap entity must also be present inside this layer entity.

<pixmap> , optional entity of a layer entitiy.

The pixmap entity defines a pixmap for the layer to which it belongs. A pixmap is a kind of bitmap to where the underlying objects are rendered. the pixmap entity is then printed as if it was an image entity. Only image entities and any text entities with a fonttype = '2', '5' or '6' can be rendered into a pixmap.

Attributes :

- x: <integer> default="25000": Defines the x-coordinate of the TopLeft point of the pixmap in ideal coordinates.
- y: <integer> default="25000": Defines the y-coordinate of the TopLeft point of the pixmap in ideal coordinates.
- sizex:<integer>default="50000": Defines the x size of the pixmap in ideal coordinates.
- sizey:<integer>default="50000": Defines the y size of the pixmap in ideal coordinates.
- width:<integer>default="0": Defines the number of pixels in x.
- height:<integer>default="0": Defines the number of pixels in y.

The objects

<objects>
...
</objects>

Within the objects entities all graphical elements must be placed. The main objects entity is unique and follows the layers entity although it could appear also before the layers entity.
The following elements/entities may be embedded within the objects entity.

<line> <polyline> <rectangle> <arc> <ellipse> <text> <day> <week> <month> <year> <hour> <minutes> <seconds><counter><shift><groupedtext><usermessagetext><vectorgroup><image>

Note that <groupedtext> and <vectorgroup> are entities that embedd other entities , so they have also an object entity which embedds other elements.
Each element type has some attributes defined and embedd some further entities necessary to "construct" those elements graphically correctly. The embedded entities are usually shared between the different elements.

In the following we give a description of the basic embedded entities shared over different element types:

<generic > , mandatory

The generic entity is mandatory for all elements and has the following attributes :

- layer_id:<integer> [0..99] default="0", mandatory. Defines the layer to which the object belongs. The behaviour is undefined if the corresponding layer does not exist within the <layers>.
- printable:<integer>[0..1] default="1", mandatory. Defines if the object is printable (1) or not (0). Note that when the object's layer is not printable this parameter is ignored and the object is marked as not printable, too.
- editable:<integer>[0..1] default="1". Defines if the object is editable (1) or not (0). Note that when the object's layer is not editable this parameter is ignored and the object is marked as not editable, too.
- linewidth: <integer>[0..1000] default="0". Defines the marking linewidth for this object. A linewidth of "0" means that the object is marked with straight vectors. Linewidths of > 0 will start a "wobble" curve along the vector path of the object. A "wobble" curve is a circular movement of the x-y scanner with a diameter of approximate (linewidth x 100) ideal units. Note that the marking time will be drastically increased for linewidhts > 0.
- render:<integer>[0..1] default="1". When set to "0" the outline of the object is not marked, when set to "1" the oultine is marked.

- repoutline:<integer>[0..99] default="0". Defines the mark-repetitions of the outlines of an object. Note that neither hatches (fill lines) nor bitmap arrays are considered as outlines so that this parameter does not have any effect on either. It applies thus on lines, polylines, arcs, circles, text and 1-dimensional barcodes.

<transformation> , optional

The transformation is optional for all elements except for <line>,<rectangle>, <polyline> and <vectorgroup>. The attributes define a 2x2 matrix that is used to implement a linear matrix transformation on the graphical object
with respect to the insertation point of the object (the insertation point of an object is the reference point that is invariant for the transformation).

- m11:<float> default="1.0". The m11 element of the 2x2 matrix.
- m12:<float> default="0.0". The m12 element of the 2x2 matrix.
- m21:<float> default="0.0". The m21 element of the 2x2 matrix.
- m22:<float> default="1.0". The m22 element of the 2x2 matrix.

  Example: For a rotation of angle phi the matrix elements would be

  m11 = cos(phi) m12 = -sin(phi)

  m21 = sin(phi) m22 = cos(phi)

<mask> , optional

The mask entity is optional and has just one attribute "hexvalue".

- hexvalue:<hexadecimal> default="0x00000000". The hexvalue has different flags that may be usefull for some types of elements.
  - 0x00000001 Non-printable flag. When set the element will not be printed (same effect as the printable attribute in the generic entity)
  - 0x00000002 Unicode flag. Has actually only an effect in combination with the <year> element when arabic digits should be displayed (must be set in this case).
  - 0x00000004 Realpointer flag. When this flag is set the red diode pointer (optional) indicates the real vector path of the element instead of an enclosing square. For text objects this flag is internally set automatically.
  - 0x00000008 Signalmask actualization flag. When set and the element is an element that actualizes its content automatically after each mark (e.g. <counter>), the actualization is supressed in case that the signalmask of the element's layer has prevented the object from beeing marked.
  - 0x00000010: For vectorgroup entities; when it is set then the vectorgroup will not be exploded into its elements for dynamic printing and is treated as a single entity in dynamic print. For small dimensioned vectorgroups this is the preferred setting. For large vectorgroups (e.g. > 30-40% of the scanfield) this bit should be cleared in case of a dynamic printing mode.

<fill> , optional

The fill entity defines a hatch of the object. Note that only closed vectorial objects can have a hatch. A <line> <arc> or <image> cannot have a fill entitiy.

- type: <integer> [0,1,2] default="0". A value of "0" deactivates filling. A value of "1" uses a single hatch (parallel lines), a value of "2" uses a crossed hatch.
- separation:<integer>[0..32167] default="0". Sets the separation between the hatch lines in ideal units. A value of "0" will be internally set to 100 ideal units.
- edge:<integer> default = "0". Defines a distance of the start and endpoint of the hatchlines with respect to the outline.
- mask:<hexadecimal> default="0x0". Defines some flags for the hatching.
    - 0x00000001 Unidirectional. When set the hatchlines are marked with an uni-directional strategy else they are marked with a bi-directional strategy.
    - 0x00000002 Connect lines. When set, the hatchlines are going to be connected to from polylines. The connections depend on some conditions and on the 'condist' parameter. When 'condist' is set to '0' hatchlines are connected when their end/start-points are closer than 2-times their actual separation, else the 'condist' defines the distance for the connections.
- condist:<integer> default="0". Sets the 'connection distance' for the hatchlines in ideal units.
- angle:<integer>[-180,+180] default="0". Sets an angle for the hatchlines in degrees.
- finclude:<integer>[0..1] default="0". When set the hatching is done taking into account figures/elements that are "inside" this element. This is only used for the graphical PC software.
- repeat:<integer> default="0". Defines the marking repeats of the hatch.

<rasterscan> , optional

The rasterscan entity is only used for <image> entities or for '2dcode' types of <text> <hour> <month>.....entities. Attribute:

- maxpower: <integer> [0,100] mandatory. Defines the power value used for a black pixel (color value = 0).
- minpower: <integer> [0,100] mandatory. Defines the power value used for a white pixel (color value = 255). Any color value in between black and white is obtained using a linear interpolation unless the complete palette table is given within the rasterscan entity.
- maxtime: <integer> [0,100] default = 100. Defines the time ni microseconds that the laser will be turned on for a pixel of black color.
- mintime: <integer> [0,100] default = 0. Defines the time in microseconds that the laser will be turned on for a pixel of white color. Values for other colors are obtained using a linear interpolation unless the complete palette table is given.
- mask: <hexadecimal> default = "0x0". A bit-mask to enable/disable specific print options.
    - 0x0001 GateOn: When set, the laser is turned on at the beginning of a scan and turned off at the end of each scan.
    - 0x0002 Prescan: When set, some prescan pixels of blank pixels are added at the beginning of each scan. The number of those pixels depend on the value of the attibute 'prescanpixel'.
    - 0x0004 When set, the horizontal scan direction is reversed.
    - 0x0008 When set, each pixel consumes exactly the same amount of time.
    - 0x0020 When set, at each pixel a small vertical slash is drawn.
    - 0x0040 When set, each pixel consumes the same time and a syncronized scan method is used.
    - 0x0080 When set, the power is modulated at each pixel according to the max/minpower settings, else the power is fixed at the value of the layer. Note that some lasertypes do not allow a power modulation.
    - 0x0100 When set, the time value applied to a pixel is not modifiable with the corresponding TCP/IP command.
    - 0x0200 When set, the power value applied to a pixel is not modifiable with the corresponding TCP/IP command.
    - 0x0400 When set, the vertical scan direction is reversed.
    - 0x00010000 When set, blank pixels are added at the end of each scan (postscan). The number of those pixels are half the number of the 'prescanpixel'.

  - 0x00020000 When set, the laser is kept on between non-white pixels.
  - 0x00040000 When set, the 'laseroffdelay' , if set in the configuration, is used for this image.
  - 0x00080000 When set, the 'laserondelay' , if set in the configuration, is used for this image.
  - 0x00100000 When set, the frequency of the PWM signal for powermodulated CO2 lasers is set according the maximum pixeltime and the allowed duty cycle.
- optimize:<integer>[0,1] default = 0. When set to 1, the bitmap is optimized to start/end each scan at the first/last non-white pixel.
- xsubpixel:<integer>[1,2,..] default = 1. Defines the number of subpixels in a horizontal direction.
- ysubpixel:<integer>[1,2,..] default = 1. Defines the number of subpixels in a vertical direction.
- passes:<integer>[1,2,...] default = 1. Defines how many time the image is to be printed.
- delay:<integer> mandatory. Defines a wait delay in microseconds at each non-white pixel before the laser is turned on.
- blankdelay:<integer>. Defines a wait delay in microseconds at each white pixel.
- startdelay:<integer> default = 0. Defines an initial delay in microseconds before the first scan of the image is started.
- vertical: <integer>[0,1] default = 0. When set to 1, the scan direction is vertical , else it is horizontal.
- bidirectional:<integer>[0,1] default = 0. When set to 1, the scans are performed bidirectional.
- invert:<integer>[0,1] default = 0. When set to 1, the color table is inverted.
- shift:<integer> default = 0. Defines an offset in deal coordinates for each scan. Positive values shift the scan towards the scan direction.
- syncrodelay: <integer> default = 0. Defines a time delay in microseconds between the position signal and the laser on/off signal. This delay is used only when the 'syncronized' bit is set.
- slashsize: <integer> . Defines the size of the vertical slash in percent of the vertical 'pixelsize'. The vertical 'pixelsize' is the height/number of subpixels.
- prescanpixel:<integer>. Defines the number of pixels used for a prescan.
- shape: <integer> [0,1,...6] default=0. Defines the mode of the image printing.
  - 0: pixelmode. The bitmap is printed with scans, scanning from left to right or up to down pixel by pixel. At each pixel the laser is turned on according the maxtime and the color value of the pixel. This is the default mode.
  - 1: square. The bitmap is scanned from left to right or up to down but the scanners are only positioned to pixels whose color is not white. At each pixel a square or multiple squares, each within the other, are printed according to the parameters set in the layer. The number of squares is defined with the 'nshapes' attribute and the size of the maximum square is determined by the 'shapex' and 'shapeaspect' attributes.
  - 2: circle. The bitmap is scanned from left to right or up to down but the scanners are only positioned to pixels whose color is not white. At each pixel a circle or multiple circles, each within the other, are printed according to the parameters set in the layer. The number of circles is defined with the 'nshapes' attribute and the size of the maximum circle is determined by the 'shapex' and 'shapeaspect' attributes.
  - 3: bidirectional lines. The bitmap is scanned from left to right or up to down but the scanners are only positioned to pixels whose color is not white. At each pixel a line or multiple bidirectional lines are printed according to the parameters set in the layer. The number of lines is defined with the 'nshapes' attribute and the size of the lines is determined by the 'shapex' and 'shapeaspect' attributes.
  - 4: lines. The bitmap is scanned from left to right or up to down but the scanners are only positioned to pixels whose color is not white. At each pixel a line or multiple lines are printed according to the parameters set in the layer. The number of lines is defined with the 'nshapes' attribute and the size of the lines is determined by the 'shapex' and 'shapeaspect' attributes.
  - 5: dotmatrix. The bitmap is scanned from left to right or up to down but the scanners are only positioned to pixels whose color is not white. At each pixel a uniform grid of points is created depending on the 'nshapes' attribute and the 'shapex' and 'shapeaspect' attributes. At each point the laser is turned on according the maxtime and the color value of the pixel.
  - 6: semispiral. The bitmap is scanned from left to right or up to down but the scanners are only

positioned to pixels whose color is not white. At each pixel a semispiral is printed according to the parameters set in the layer. The number of turns of the spiral is defined with the 'nshapes' attribute and the size of the maximum square is determined by the 'shapex' and 'shapeaspect' attributes.

- nshapes:<integer>. Defines the number of subshapes in case that the 'shape' attribute is not set to 0.
- shapex: <float>. Define the size of the maximum subshape. A value of 1.0 means that the size of the maximum subshape occupies a complete pixel cell.
- shapeaspect: <float>. Defines the aspect ratio of the subshapes. The default value is 1.0.

The <rasterscan> entity can have an optional <palette> entity.

<palette> , optional

The palette entity is only used within the <rasterscan> entity and defines the color table, the relationship between color of a pixel and power or laser on-time values. If no palette entity is given, the power and laser on-time values for a certain pixel of a certain color is calculated using a linear interpolation between the minpower/maxpoer and mintime/maxtime attributes of the rasterscan entity. This is the usual case where no palette entity is used. Attributes:

- colortable: <string>. The colortable is a string containing 255 data triples, each element of the triple separated by a <space> and each triple separated by a ','. A data triple consists thus of three numbers <power> <freq> <time>. The first triple applies to color value 0 (black), the next triple to color value 1, and so on until the color value 255 (white) is reached. The <power> value must be given in permille and is used to calculate the applied power value for the color. The <freq> value must be given but is not used actually (any value is valid). The <time> value must be given in permille and is used to calculate the applied laser on-time for the color. The applied power is calculated as power = <power> - (color*1000)/255. The applied laser on-time is calculated as time = <time> - (color*1000)/255. Note that the palette and colortable is only used when you do want to have a linear colortable.

Entities

<line ...attributes>
..
</line>

The line entity represents a single straight line. Attributes:

- sx: <integer> , mandatory. The x position of the start point of the line in ideal coordinates.
- sy: <integer> , mandatory. The y position of the start point of the line in ideal coordinates.
- ex: <integer> , mandatory. The x position of the end point of the line in ideal coordinates.
- ey: <integer> , mandatory. The y position of the start point of the line in ideal coordinates.
- id: <positive integer> . A unique identifier for this element.
- name <string>, default="". An optional name for this object that may be used as a reference for eventhandler actions.

Mandatory entities within a <line> entity:

- <generic>

Optional entities within a <line> entity:

- <mask>

<rectangle ...attributes>
..
</rectangle>

The rectangle entity represents a rectangle. It can be either defined by its 4 edge points in cw or ccw sense, or it can be defined by its top-left position, its width and height.
If only the 4 edge points are given you should not set a <transformation> entity as the 4 edge points are supposed to be the final points of the rectangle. If you would add a <transformation> entity to the rectangle defined by its 4 edge points, the parser uses the <transformation> entity to invert the transformation and calculate the original untransformed rectangle and set the width and height. Note that the object defined by 4 edge points has to be at least a parallelogram, else the representation may fail !
Optionally, you can define a rectangle by its top-left position and the width and height. In this case the <transformation> entity would have the desired effect of being applied to the rectangle.

Attributes:

defined by 4 points

- x: <integer> , mandatory . The x position of the first point of the rectangle in ideal coordinates.
- y: <integer> , mandatory . The y position of the first point of the rectangle in ideal coordinates.
- x2: <integer> , mandatory . The x position of the second point of the rectangle in ideal coordinates.
- y2: <integer> , mandatory . The y position of the second point of the rectangle in ideal coordinates.
- x3: <integer> , mandatory . The x position of the third point of the rectangle in ideal coordinates.
- y3: <integer> , mandatory . The y position of the third point of the rectangle in ideal coordinates.
- x4: <integer> , mandatory . The x position of the forth point of the rectangle in ideal coordinates.
- y4: <integer> , mandatory . The y position of the forth point of the rectangle in ideal coordinates.

defined by width and height

- x: <integer> , mandatory . The x position of the first point of the rectangle in ideal coordinates.
- y: <integer> , mandatory . The y position of the first point of the rectangle in ideal coordinates
- width: <integer> , mandatory . The width of the rectangle in ideal coordinates.
- height: <integer> , mandatory . The height of the rectangle in ideal coordinates.

both cases

- rx: <integer> , optional. The radius in X of the corner of the rectangle in ideal coordinates ("0" for sharp corners) .
- ry: <integer> , optional . The radius in Y of the corner of the rectangle in ideal coordinates ("0" for sharp corners) .
- id: <positive integer> . A unique identifier for this element.
- name <string>, default="". An optional name for this object that may be used as a reference for eventhandler actions.

Mandatory entities within a <line> entity:

- <generic>

Optional entities within a <line> entity:

- <mask>

- <transformation>

<polyline ...attributes>
..
</polyline>

The polyline entity represents a sequence of lines. Attributes:

- type: <string> , default="". When set to "closed" the polyline will be forecely closed if the end point does not match exactly the start point.
- points: < integer integer,.....> , mandatory. Defines the xy points of the polyline segments. " integer integer,...." is a sequence of x y pairs. An x y pair is seperated from the next pair by a "," and a space. The y coordinate is spearated from the x coordinate by a space. Example: points=" 34300 25600, 35500 22500, 41500 21900, 43000 24900, 40300 26000, 34300 25600".
- id <positive integer>. A unique identifier for this element.
- name <string> default="". An optional name for this object that may be used as a reference for eventhandler actions.

Mandatory entities within a <polyline > entity:

- <generic>

Optional entities within a <polyline > entity:

- <mask>
- <fill> when polyline is of "closed" type

<arc...attributes>
..
</arc>

The arc entity represents an arc defined by the center point, a small axis, large axis, the start angle and the end angle. Attributes:

- cx: <integer> , mandatory. The x position of the center point of the arc in ideal units.
- cy: <integer> , mandatory. The y position of the center point of the arc in ideal units.
- smallaxis: <integer> , mandatory. The length of the small axis of the arc in ideal units.
- largeaxis: <integer> , mandatory. The length of the large axis of the arc in ideal units.
- startangle: <float> , mandatory. The start angle in radian.
- endangle: <float> , mandatory. The end angle in radian.
- flip: <integer> , default="0". Per default an arc is done clockwise from the start angle to the end

angle. Setting flip= "1" reverts the direction the arc is done. Example: startangle=0 endangle=1.57 flip=0 will do 90 degree arc. Setting flip="1" will result in a 360-90=270 degree arc.

- id <positive integer>. A unique identifier for this element.
- name <string> default="". An optional name for this object that may be used as a reference for eventhandler actions.

Mandatory entities within a <arc> entity:

- <generic>

Optional entities within a <arc> entity:

- <mask>
- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the arc. The reference or invariant point for the transformation is the center point of the arc.

<ellipse...attributes>
..
</ellipse>

The ellipse entity represents an ellipse defined by the center point, a small axis and a large axis. Attributes:

- cx: <integer> , mandatory. The x position of the center point of the ellipse in ideal units.
- cy: <integer> , mandatory. The y position of the center point of the ellipse in ideal units.
- rx: <integer> , mandatory. The length of the small axis of the ellipse in ideal units.
- ry: <integer> , mandatory. The length of the large axis of the ellipse in ideal units.
- id <positive integer>. A unique identifier for this element.
- name <string> default="". An optional name for this object that may be used as a reference for eventhandler actions.

Mandatory entities within a <ellipse> entity:

- <generic>

Optional entities within a <ellipse> entity:

- <mask>
- <fill>
- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the ellipse. The reference or invariant point for the transformation is the center point of the ellipse.

<text...attributes>
..
</text>

The text entity represents a static text whose string is not actualized automatically. Attributes:

- x: <integer> , mandatory. The x position of the reference point in ideal units. If this <text> entity is member of a <groupedtext> entity this attribute may be omitted.
- y: <integer> , mandatory. The y position of the reference point in ideal units. If this <text> entity is

member of a <groupedtext> entity this attribute may be omitted.

  - The reference point of a text object is the point where the text is placed to. Drawing of the characters, bars or bitmap pixels is "started" from the reference point. For barcode and 2D barcode text types the reference point is usually (left aligned) the top-left point. For human readable characters the reference point is the bottom-left point.
- texttype: <string> , mandatory. Sets the type of text. "chars" means human readable characters, "barcode" means a 1-dimensional barcode (bars and spaces) and "2dcode" means a 2-dimensional barcode type (bitmap image). If this <text> entity is member of a <groupedtext> entity this attribute may be omitted.
- spotreduction:<integer> default="300". Optional attribute for "chars" and "barcode" type in case that a special laserfont is used for marking the characters. It defines the theoretical spotsize in ideal coordinates that will be used to correct the marking path for not having "crossing" lines. If this <text> entity is member of a <groupedtext> entity this attribute may be omitted.
- width:<positive integer>, mandatory for "2dcode" texttype. Defines the width in ideal coordinates of the desired 2D code before any possible <transformation> will be applied. The 2D code pixels of the created barcode will be stretched and fitted to match the whole width.
- height:<positive integer>, mandatory for "2dcode" texttype. Defines the height in ideal coordinates of the desired 2D code before any possible <transformation> will be applied. The 2D code pixels of the created barcode will be stretched and fitted to match the whole height.
- fixedsize:<integer>[0,1] default="0": optional for "2dcode" texttype. When set the size of the 2D code will be adjusted such that the cell size (pixel) equals the value defined with the 'pixelsize' attribute.

- pixelsize:<integer>: Defines the cell size in case that the 'fixedsize' attribute has been set to '1'.

- id <positive integer>. A unique identifier for this element.
- name <string> default="". An optional name for this object that may be used as a reference for eventhandler actions.

For firmware version > 5.3.1 the following attributes can be set if the texttype is "barcode" or "2dcode":

- text_visible: <integer> [0,1] default="0". For barcode or 2dcode types; when set to "1" the human readable characters will be visible and marked, too.
- text_position: <integer> [0,1] default="0". For barcode or 2dcode types; determines the position of the human readable characters when beeing marked. "0" means at the bottom of the barcode, "1" means on top of the barcode. If the barcode is a 2D code the text_position can be also: "2" for displaying the text at the topleft of the code; "3" for displaying the text at the topright of the code.
- text_separation: <integer> default="0". For barcode or 2dcode types; determines the separation of the human readable characters from the top or bottom extent of the barcode. Units are ideal units.

These attributes are also attributes of the <barcode_1d> entity but are added also to the <text> entitiy in case that the type is a "2dcode". If the type is a "barcode" type , you can set these attributes either in the <text> entity or in the <barcode_1d> entity or in both, in which case the values should be the same in both entities.

Mandatory entities within a <text> entity:

- <generic>
- <font> when the texttype is "char" or "barcode" with human readable characters displayed.
- <barcode_1d> when texttype is "barcode" (or "2dcode" AND the pattern results in just a linear barcode pattern).
- <barcode> when texttype is "barcode" or "2dcode".
- <rasterscan> when texttype is "2dcode" (or "barcode" AND the pattern results in having 2D components).

Optional entities within a <text> entity:

- <mask>
- <fill>
- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the entity.
- <rasterscan> Applies only for 2code texttypes.

The string of the <text> entity is encoded as the content of the <text> tag. This content is in any case a CDATA section and UTF-8 coded. The string should not contain any LF character ! If your string needs to have a newline character, just use one CR character for each newline. Typically, for Window systems a newline is programmed as CR + LF, for Unix-like systems a newline is programmed as LF only. Thus, within the CDATA section our convention is not to used any LF character and use as a newline character the single CR character.

<font...attributes>
..
</font>

The font entity is a mandatory entity within an entity that may represent human readable characters (<text> <counter> <year>....). It basically defines the type of font , size, etc... used to render the characters.
Attributes:

- type: <integer>, mandatory. Defines the type of font used to render the characters.
    - "0" mfs-fonts. These are fonts that may contain up to 255 characters. The <text> string read from the xml file (UTF-8 encoded) is internally treated as a Unicode string. For this font type only the lower byte of the Unicode char is evaluated to extract the corresponding glyph out of the font file. Usually, the font data in these files are stored according the Windows-1252 codepage.
    - "1" Obsolete type. Was using some "TrueType"-similar font to render the characters. Should not be used in the xml-format.
    - "2" mfs-fonts. This kind of font is named also "dotmatrix" or "crystall" font. The format is exactly the same as in the "0" case, but the characters are not rendered using line segments but only dots at each segment point.
    - "3" Obsolete type. Can only be used inside the PC software where it then uses the rendering mechanisms of Windows to render the characters. Denominated also "Opentype" in the PC software.
    - "4" unx or unf-fonts. These fonts are unicode fonts encoding up to 65535 characters according the unicode charts.
    - "5" unx or unf-fonts. This is the unicode variant of the "dotmatrix" or "crystall" font. Characters are rendered as dots using the segment points stored inside the font file.
    - "6" Freetype fonts. Uses the same library in the PC software as well as in the embedded laser program to render the fonts. The font files are standard ttf (truetype), otf (opentype) or ttc (truetype collection) files. Characters are always rendered with "double" lines which normally requires much longer marking times. This type of font must be used when you want to use the Truetype fonts of your system. Please note that there is a limited memory space inside the laser for storing truetype fonts (about 60 MBytes), so you should always keep track of the truetype fonts you may want to work with.
- facename:<string>. mandatory only for PC software. Must be set to the facename as it appears inside the ttf, otf or ttc file when the fonttype is "6". For the fonttypes 0,2,4,5 the facename is the real filename of the font file.

- synonym:<string> mandatory. Must be set to the real filename of the font in any case.
- faceindex: <integer> default="0", mandatory for "6" fonts when the font is a ttc font (truetype collection). The faceindex in a truetype collection defines which font of the collection should be selected for rendering.

<cirular...attributes>
..
</circular>

The circular entity is an optional entity of the font entity when the text or 1-dimensional barcode should be rendered following a circle. Attributes:

- radian: <integer> mandatory. Defines the radius of the circle in ideal coordinates on which the text or barcode should be mapped to.
- startangle: <float>. The starting angle in radian of the circle from where on the mapping starts. The reference point of the text is mapped to the starting angle.
- centrate: <integer>. [0,1] default="0". When set to "1" the text is centered with respect to the startangle.

<circle-aligned...attributes>
..
</circle-aligned>

The circle-aligned entity is an optional entity of the font entity of "chars" type only. It makes sense only for text objects with newline characters. The starting character of each new line will be aligned at a circle. Attributes:

- radian: <integer> mandatory. Defines the radius of the circle in ideal coordinates on which the text or barcode should be aligned.
- startangle: <float>. The starting angle in radian of the circle from where on the alignment starts.

<metric...attributes>
..
</metric>

The metric entity is mandatory for each font entity. It defines the size of the font and other font relevant parameters that may change the rendering characteristics. Attributes:

- height: <integer> mandatory. The height represents the font "size" that should be used for rendering and is given in <ideal units/100>. The "real" character height depends on this parameter as well as on the font itself (the way the font was drawn when it was created). The height attribute matches the real size of the virtual box in which the designer of the font has drawn the characters. Therefore, it roughly matches the size of the "largest" character.
- ratio: <float> default="1.0". The aspect ratio of the characters. The x-size (width) of the characters are affected directly by the ratio. A ratio > 1.0 makes the character wider a ratio < 1.0 makes it narrower.
- spacesize: <float> default="1.0". A scaling factor for the size of the space character. The intrinsic size according to the font is multiplied with this factor to obtain the size reserved for the space character. For 'freetype' fonts the 'kerning' flag must be set if you want to make the spacesize effective.
- separation: <integer> [0..65535] default="100". The additional sepearation between consecutive

characters in <ideal units/10>. Character rendering is started with the first character beginning with the coordinate system's origin placed in the reference point of the text entity. After rendering a character the drawing origin for the next character is set by moving horizontally (in positive x direction) the separation distance measured from the right position of the enclosing rectangle of the preceeding character. If kerning is enabled (only for fonttype "5") the next drawing origin is determined from the kerning table of the font and the separation distance is always added to the value obtained from the kerning table.

- linespacing: <integer>[-32767..32767] default="1000". Defines the spacing between new lines within a text object in ideal coordinates. A new line starts with a CR character in the text string. The drawing origin of a new line with respect to the previous line is calculated as (height x 125) + linespacing.
- fontflags: <hexadecimal> default = "0x00000000". Flags that determine the behaviour of rendering.
  - 0x00000001: RTL flag. Characters are drawn right-to-left ordered.
  - 0x00000004: fixed space size. A space occupies a fixed width. The fixed width for a space is 1/4 th of the virtual width of a character box (which in turn is the same as the height of the metric). If the flag is not set a space corresponds to an additional advance of 'separation'.
  - 0x00000008: kerning. When set and fonttype is "5" the character spacing is determined from the kerning table + 'separation'.
  - 0x00000010: max X: When set, the rendered text will be stretched to a maximum width according the 'stretchleft' and 'stretchright' attributes only if the width of the text is greater than given by the 'stretchleft' and 'stretchright' attributes.
  - 0x00000020: max Y: When set, the rendered text will be stretched to a maximum height according the 'stretchtop' and 'stretchbottom' attributes only if the height of the text is greater than given by the 'stretchtop' and 'stretchbottom' attributes.
  - 0x00000040: stretchX: When set, the rendered text will always be stretched to a maximum width according the 'stretchleft' and 'stretchright' attributes.
  - 0x00000080: stretchY: When set, the rendered text will always be stretched to a maximum height according the "stretchtop' ' and "stretchbottom' ' attributes.
  - 0x00000100: vertical: When set, characters are rotated 90 degrees ccw before rendered. They are still drawn from left to right. If you want to draw them vertically you would have to use the <transformation> entity and set the 2x2 matrix elements accordingly.
  - 0x00000400: paragraph. When set, the whole text is treated as a paragraph not exceeding the width defined with 'stretchleft' and 'stretchright'. If necessary, newline characters are automatically inserted.
  - 0x00001000: underline. When set, the text is underlined.
  - 0x00002000: chars alignment. When set, the characters of a circular text are forced to be vertically aligned.
  - 0x00010000: arabic: When set, a bidirectional algorithm automatically determines RTL or LTR ordering (arabic or hebreic).
  - 0x00020000: swap arabic: When set, the parentesis characters for RTL languages are swapped too.
  - 0x00040000: multiscript: When set, the string is parsed for language scripts and all typical script rules are applied. This flag works onlyfor 'freetype' fonts.
  - 0x00100000: single-stroke: When set, all outlines of the selected 'freetype' font are treated as single-stroke lines/splines. This flag basically removes the start point of any outline, that was added as the last point for closing the outline.
  - 0x00200000: grayscaled colored: When set and the object is rendered into a layer with enabled pixmap and the font is a freetype font, then the text is rendered as a 8-bit grayscale image, else it is a monocrome bitmap.
- align:<integer> [0,1,2] default="0": Defines the final text alignment with respect to the reference point of the text entity. "0" left-aligned "1" center-aligned "2" right-aligned.
- stretchleft: <integer> : Defines the left position in ideal units if maxX or stretchX fontflags are turned on.
- stretchright: <integer> : Defines the right position in ideal units if maxX or stretchX fontflags are

turned on.

- stretchtop: <integer> : Defines the top position in ideal units if maxY or stretchY fontflags are turned on.
- stretchbottom: <integer> : Defines the bottom position in ideal units if maxY or stretchY fontflags are turned on.

- underlineoffset: <integer> : Defines an offset for the underline position in font design units.

<dotmatrix...attributes>
..
</dotmatrix>

The dotmatrix entity is mandatory for each font entity if the fonttype is "2" or "5". Attributes:

- speed:<integer> mandatory. Defines the scanner speed used to move between the dots. The units are ideal-units per second (see also <layer> entity).
- resolution:<integer> mandatory. Defines the scanner resolution for the movment between the dots. Units are points per 1000 ideal units (see also <layer> entity).
- time:<integer> mandatory. Defines the laser ON-time for marking a single dot. Units are microseconds.
- gateon: <integer> [0,1] default="0". When "1", the laser is kept ON during the movement between subsequent dots of a character.
- wait: <integer> mandatory. Defines the wait time at each single dot of a character before the laser is turned ON. Units are microseconds.

<barcode_1d...attributes>
..
</barcode_1d>

The barcode_1d entity is mandatory for each text type entity (<text> <counter>.....) when the texttype is "barcode" or when the type is "2dcode" but may have just a linear pattern. Attributes:

- height: <integer> mandatory. Defines the height in ideal units of the barcode's bars.
- subbars: <integer> [1,2...] mandatory. Defines the number of lines used to draw one barcode bar.
- separation: <integer> mandatory. Defines the separation in ideal units between each line.
- spotcorrection: <integer> default="0". Sets a correction due to the spotsize of the laser. The value is given in ideal units. See the PC software help for more details on this parameter.
- reduction: <integer> default="0". Sets a correction due to the spotsize of the laser. The value is given in ideal units. See the PC software help for more details on this parameter.
- flags: <integer> default="0". Sets some flags for barcode printing. Note that this value must be given as an integer and not in hexadecimal.
    - 1 bidir. Enables bi-directional marking of the barcode.
    - 2 RTL. Enables right to left marking of the barcode.
    - 4 explode. Enables exploding of the barcode's lines for marking. For dynamic printing this flag should be enabled as marking time is decreased when the lines are exploded. If not set the barcode is seen as a single entity in dynamic printing and marking is not started unless the complete barcode is within the scanfield.
    - 8 short quietzone bars: Draws the bars for inverted 1D-codes, that contain synro-bars, slightly shorter.
- text_visible: <integer> [0,1] default="0". When set to "1" the human readable characters will be visible and marked, too.

- text_position: <integer> [0,1] default="0". Determines the position of the human readbale characters when beeing marked. "0" means at the bottom of the barcode, "1" means on top of the barcode.
- text_separation: <integer> default="0". Determines the separation of the human readbale characters from the top or bottom extent of the barcode. Units are ideal units.

<barcode...attributes>
..
</barcode>

The barcode entity is mandatory for each text type entity (<text> <counter>.....) when the texttype is "barcode" or "2dcode". Refer to the PC software help and barcode reference for further details of each attribute. Attributes:

- invert: <integer> [0,1] default="0". When set to "1" the barcode or 2D code is inverted (bars become spaces and spaces become bars, black pixels become white pixels and viceversa).
- type: <integer> mandatory. A numerical value that defines the exact type of the barcode.
    - list of all possible barcodes, 2D codes are marked green.

        - 1 Code11
        - 2 Code 2 of 5
        - 3 Interleaved 2 of 5
        - 4 Code 2 of 5 IATA
        - 5 Code 2 of 5 Matrix
        - 6 Code 2 of 5 Data Logic
        - 7 Code 2 of 5 Industrial
        - 8 Code 3 of 9
        - 9 Code 3 of 9 ASCII
        - 10 EAN-8
        - 11 EAN-8 with 2 digits add on
        - 12 EAN-8 with 5 digits add on
        - 13 EAN-13
        - 14 EAN-13 with 2 digits add on
        - 15 EAN-13 with 5 digits add on
        - 16 EAN-128 (GS1-128)
        - 17 UPC-12 digits (UPC-A)
        - 18 CodaBar (2 widths)
        - 19 CodaBar (18 widths)
        - 20 Code 128
        - 21 Deutsche Post Leitcode
        - 22 Deutsche Post Identcode
        - 23 Code 16K
        - 24 Code 49
        - 25 Code 93
        - 26 UPC 25
        - 27 UPCD1
        - 28 Flattermarken
        - 29 GS1 Databar (RSS-14)
        - 30 GS1 Databar Limited (RSS Limited)
        - 31 GS1 Databar Expanded
        - 32 Telepen Alpha
        - 33 UCC-128 (GS1-128/EAN-128)
        - 34 UPC A

- 35 UPC A with 2 digits add on
- 36 UPC A with 5 digits add on
- 37 UPC E
- 38 UPC E with 2 digits add on
- 39 UPC E with 5 digits add on
- 40 USPS PostNet ZIP 5 digits
- 41 USPS PostNet ZIP 5 digits + checkdigit
- 42 USPS PostNet ZIP+4 (5 digits+ 4 digits)
- 43 USPS PostNet ZIP +4 (5 digits + 4 digits + checkdigit)
- 44 USPS PostNet DPBC (5 + 4 + 2 digits)
- 45 USPS PostNet DPBC (5 + 4 + 2 digits + CD)
- 46 Plessey Code
- 47 MSI Plessey Code
- 48 Serial Shipping Container Code 18
- 49 FIM
- 50 LOGMARS
- 51 Pharmacode one-track
- 52 PZN
- 53 Pharmacode two-track
- 54 General Parcel
- 55 PDF417
- 56 PDF417 truncated
- 57 Maxicode
- 58 QR-code
- 59 Code128 Subset A
- 60 Code128 Subset B
- 61 Code128 Subset C
- 62 Code 93 ASCII
- 63 Australian Post standard customer
- 64 Australian Post standard customer 2
- 65 Australian Post standard customer 3
- 66 Australian Post reply paid
- 67 Australian Post routing
- 68 Australian Post redirection
- 69 ISBN
- 70 Royal Mail 4 state customer
- 71 DataMatrix
- 72 EAN-14
- 73 Codablock-E
- 74 Codablock-F
- 75 NVE-18
- 76 Japanese Postal customer code
- 77 Korean Postal authority code
- 78 GS1 Databar truncated
- 79 GS1 Databar stacked
- 80 GS1 Databar stacked omnidirectional
- 81 GS1 Databar expanded stacked
- 82 Planet Code 12
- 83 Planet Code 14
- 84 MicroPDF417
- 85 USPS OnCode
- 86 Plessey Code bidirectional
- 87 Telepen

  - 88 GS1-128
  - 89 ITF-14
  - 90 KIX
  - 91 BC412
  - 92 Aztec code
  - 93 DAFT code
  - 94 Italian Postal 2 of 5
  - 95 Italian Postal 3 of 9
  - 96 DPD code

- checkdigit:<integer> default="0". A numerical value indicating the type of check digit calculation.

  - 0: No check digit calculation is performed
  - 1: Standard check digit of barcode type will be used
  - 2: Modulo 10 (usually used with Interleaved 2of5)
  - 3: Modulo 43 (suggested for Code39 and Logmars, consist of 1 digit)
  - 4: Modulo 47 (2 digits)
  - 5: Method for DP Leitcode
  - 6: Method for DP Identcode
  - 7: Method for Code11 (1 digit)
  - 8: Method for Code11 (2 digits)
  - 9: Method for USPS Postnet
  - 10: Method for MSI (1 digit)
  - 11: Method for MSI (2 digits)
  - 12: Method for Plessey
  - 13: Method for EAN 8
  - 14: Method for EAN 13
  - 15: Method for UPC A
  - 16: Method for UPC E
  - 17: EAN 128 internal method (Modulo 103)
  - 18: Code 128 internal method (Modulo 103)
  - 19: Method for Royal Mail 4 State
  - 20: Mod 11 Method for PZN
  - 21: Mod 11 (using weight 7)
  - 22: Method for EAN 14
  - 23: Method for Korean Postal Authority (Modulo 10 based)
  - 24: Method for Planet (Modulo 10 based)
  - 25: Method for Italian Postal 2/5 (Modulo 10 based)
  - 26: Mod 36 (ISO/IES 7064) for DPD Barcode
  - 27: Mod 16 for Codabar Barcode
  - 1000: OEM type 0 (mod 10, position weight)

- encodemode: <integer> default="1". A numerical value that specifies the encoding mode of the string data.
  - 1: low byte. Only the lower byte of the unicode character is encoded.
  - 2: byte stream. The unicode characters are encoded as a byte stream, where first the low byte then the high byte is encoded.
  - 3: highlow stream. Same as '2' but the high byte is encoded before the low byte.
- codepage: <integer> default="1". Actually not used.
- compositecomponent: <integer>. Actuall not used.
- esc_sequence: <integer> [0,1] default="0". When set to "1", ESC-sequence is enabled so that some

non-printable control characters can be entered manually (<GS> <FUNC0>..). See the barcode reference in the help manual for details on ESC sequencing.

- quietzone: <integer> default="0". A numerical value used to define a quietzone at the left/right side of a linear barcode or used to define a frame around a 2D code.
- format: <integer> default="0". A numerical value defining a specific format of some barcode types.
    - Datamatrix
        - 0: Default mode : The standard format of Data Matrix without a special header
        - 1: UCC/EAN mode : Special format defined by UCC and EAN for encoding Application Identifiers. This format adds the function character FNC1 at 1st position in the symbol.
        - 2: Industry mode: supports peculiar industry formats (adds FNC1 at 2nd position)
        - 3: Macro 05 mode: [)>Rs05Gs is encoded at the beginning of the code
        - 4: Macro 06 mode: [)>Rs06Gs is encoded at the beginning of the code
    - QR code
        - 0: Default mode
        - 1: UCC/EAN
        - 2: Industry
    - Codablock_F
        - 0: Default mode
        - 1: UCC/EAN
- size:<integer> default="0". A numerical value defining the size of a specific barcode type.
    - Aztec Code
        - 0: default. Size is determined automatically.
        - 1: 15x15
        - 2: 19x19
        - 3: 23x23
        - 4: 27x27
        - 5: 31x31
        - 6: 37x37
        - 7: 41x41
        - 8: 45x45
        - 9: 49x49
        - 10: 53x53
        - 11: 57x57
        - 12: 61x61
        - 13: 67x67
        - 14: 71x71
        - 15: 75x75
        - 16: 79x79
        - 17: 79x79
        - 18: 83x83
        - 19: 87x87
        - 20: 91x91
        - 21: 95x95
        - 22: 101x101
        - 23: 105x105
        - 24: 109x109
        - 25: 113x113
        - 26: 117x117
        - 27: 121x121
        - 28: 125x125
        - 29: 131x131
        - 30: 135x135

    - 31: 139x139
    - 32: 143x143
    - 33: 147x147
    - 34: 151x151
    - 35: 19x19Rd
    - 36: 23x23Rd
    - 37: 27x27Rd
    - 38: Rune 11x11
- QRCode
    - 0: Default
    - 1: 21x21
    - 2: 25x25
    - 3: 29x29
    - 4: 33x33
    - 5: 37x37
    - 6: 41x41
    - 7: 45x45
    - 8: 49x49
    - 9: 53x53
    - 10: 57x57
    - 11: 61x61
    - 12: 65x65
    - 13: 69x69
    - 14: 73x73
    - 15: 77x77
    - 16: 81x81
    - 17: 85x85
    - 18: 89x89
    - 19: 93x93
    - 20: 101x101
    - 21: 105x105
    - 22: 109x109
    - 23: 113x113
    - 24: 117x117
    - 25: 121x121
    - 26: 125x125
    - 27: 129x129
    - 28: 133x133
    - 29: 137x137
    - 30: 141x141
    - 31: 145x145
    - 32: 149x149
    - 33: 153x153
    - 34: 157x157
    - 35: 161x161
    - 36: 165x165
    - 37: 169x169
    - 38: 173x173
    - 39: 177x177
- MicroQRCode
    - 0: Default
    - 1: 11x11
    - 2: 13x13

- 3: 15x15
- 4: 17x17
- DataMatrix
  - 0: Default
  - 1: 10x10
  - 2: 12x12
  - 3: 14x14
  - 4: 16x16
  - 5: 18x18
  - 6: 20x20
  - 7: 22x22
  - 8: 24x24
  - 9: 26x26
  - 10: 32x32
  - 11: 36x36
  - 12: 40x40
  - 13: 44x44
  - 14: 48x48
  - 15: 52x52
  - 16: 64x64
  - 17: 72x72
  - 18: 80x80
  - 19: 88x88
  - 20: 96x96
  - 21: 104x104
  - 22: 120x120
  - 23: 132x132
  - 24: 144x144
  - 25: 8x18
  - 26: 8x32
  - 27: 12x26
  - 28: 12x36
  - 29: 16x36
  - 30: 16x48

- mode:<integer> default="0". Numerical value that determines the mode for some specific barcode type.
  - PDF417
    - 0: Default encoding
    - 1: Binary encoding
  - Maxicode
    - 0: Reserved 0
    - 1: Reserved 1
    - 2: SCM Numeric
    - 3: SCM Alphanumeric
    - 4: Mode 4(default)
    - 5: Full ECC
  - Datamatrix:
    - 0: Square
    - 1: Rectangular
- error_correction: <integer> default="0". Numerical value that determines the error correction behavious for certain barcode.
  - AztecCode

        - values from 0-99
    - PDF417
        - values from -1 (default) to 8
    - QRCode
        - values from 0 to 3
- mask: <integer> default ="-1". Only used for QRcode or MicroQRCode.
    - QRcode/MicroQRCode
        - -1: default
        - 0: Fixed 0
        - 1: Fixed 1
        - 2: Fixed 2
        - 3: Fixed 3
- compact: <integer> default="0". Actually not used.
- rows:<integer> default="-1". Numerical value defining the row numbers for specific barcode types.
    - GS1DatabarExpandedStacked
        - values from 2,4,6,....22
    - Codablock F
        - values from -1, 2...44
    - PDF417
        - Values from -1, 3...90
- columns: <integer> default="-1". Numerical value defining the column numbers for specific barcode types.
    - Codablock F
        - values from -1, 4...62
    - PDF417
        - values from -1, 1...30
- binary: <integer> default="0". When set to "1" the encoding mode for Datamatrix and Aztec code is forced to be in binary mode.
- undercut: <integer>. Actually not used.
- modulewidth: <integer> default="1". Numerical value defining the modulewidth for the 2D barcode types.
- moduleheight: <integer> default="1". Numerical value defining the modulewidth for the 2D barcode types.
- humantext_view: <integer> default="0". Sets the display format of the optional human readable text for linear barcodes.
    - 0: default display
    - 1: show also the checkdigit
    - 2: formated display (automatic)
- stringformat: <string> default="". Please, refer to the barcode reference in the help for more details.
- ratio: <string> default="". Please, refer to the barcode reference in the help for more details.
- pdfratio: <string> default="". Please, refer to the barcode reference in the help for more details.
- formatspecifier: <string> default="". Please, refer to the barcode reference in the help for more details.

Expiry or automated date entities

```
<dbfile...attributes>
..
</dbfile>
```

The <dbfile> entity represents a remote textfile that can serve as a small database file for the automated date entities <day> <year> <month> <week><minutes> <hour> <seconds>. It can only be used as an embedded entity for these date entities. Attributes:

- remote: <string>. mandatory. Defines the filename from where the data to be displayed are read from. The automated date entity "creates" a standard text string according to all of its settings and according to the actual date automatically. The database file serves to replace the actual text by a new string. Please, refer to the help for more details about the database file for expiry entities.

<day...attributes>
..
</day>

The day entity represents an automatic text displaying the day. Attributes: The <day> entity heritates all attributes from the <text> entity. Additionally it has the following attributes:

- type: <integer> mandatory. Defines the type of string to be displayed.
  - 0: 1..31 two digits
  - 1: 1...7 one digit, day of the week
  - 2: MON, TUE, ...three characters
  - 3: 1..366 Julian day
  - 4: NNNN Absolute day, number of days since 1.1. 1970
- offset: <integer> default="0". The offset in days or months (depending on the format) added/subtracted to the actual date.
- minutes: <integer> [-32767,32767] default="0". An offset in minutes that is added to the actual date.
- expiry: <integer> [0,1,2] default="0". Sets a special format for calculating the day to be displayed.
  - 0: standard format. Offset is given in days.
  - 1: date always rounded to monday.
  - 2: offset interpretated in months.
- format: <integer> default="0". Sets some flags for the date format. Note that this value must be given as an integer and not in hexadecimal.
  - 0x00000001: Hijri. Calculates the day according the hijri calendar.
  - 0x00010000: Displays arabic digits.
  - 0x00020000: Displays east-arabic digits.
  - 0x00030000: Displays indic digits.
  - 0x00000010: displays day "366" for leap year 29th of february when julian day is selected.
- daynames:<string> default="". Defines the 'daynames' for type '2'. A typical string is "MONTUEWED.....". For each day the same amount of characters must be set (can be different of '3' characters) starting with the characters for Monday and ending with the characters for Sunday.

Mandatory entities within a <day> entity:

- <generic>
- <font> when the texttype is "char" or "barcode" with human readable characters displayed.
- <barcode_1d> when texttype is "barcode"
- <barcode> when texttype is "barcode" or "2dcode".
- <rasterscan> when texttype is "2dcode".

Optional entities within a <day> entity:

- <mask>

- <fill>
- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the entity.
- <dbfile> An optional entity for reading the date characters from a file.

<month...attributes>
..
</month>

The month entity represents an automatic text displaying the month . Attributes: The <month > entity heritates all attributes from the <text> entity. Additionally it has the following attributes:

- type: <integer> mandatory. Defines the type of string to be displayed.
  - 0: 01..12 two digits
  - 1: JAN, FEB, .. three characters
  - 2: 1,2..9,A,B,C hexadecimal
- offset: <integer> default="0". The offset in days or months (depending on the format) added/subtracted to the actual date.
- minutes: <integer> [-32767,32767] default="0". An offset in minutes that is added to the actual date.
- expiry: <integer> [0,1,2] default="0". Sets a special format for calculating the month to be displayed.
  - 0: standard format. Offset is given in days.
  - 1: date always rounded to monday.
  - 2: offset interpretated in months.
- format: <integer> default="0". Sets some flags for the date format. Note that this value must be given as an integer and not in hexadecimal.
  - 0x00000001: Hijri. Calculates the day according the hijri calendar.
  - 0x00010000: Displays arabic digits.
  - 0x00020000: Displays east-arabic digits.
  - 0x00030000: Displays indic digits.
- monthnames:<string> default="". Defines the 'monthnames' for type '1'. A typical string is "JANFEB....". For each month the same amount of characters must be set (can be different of '3' characters) starting with the characters for January and ending with the characters for December.

Mandatory entities within a <month> entity:

- <generic>
- <font> when the texttype is "char" or "barcode" with human readable characters displayed.
- <barcode_1d> when texttype is "barcode"
- <barcode> when texttype is "barcode" or "2dcode".
- <rasterscan> when texttype is "2dcode".

Optional entities within a <month> entity:

- <mask>
- <fill>
- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the entity.
- <dbfile> An optional entity for reading the date characters from a file.

<year...attributes>
..

</year>

The year entity represents an automatic text displaying the year. Attributes: The <year> entity heritates all attributes from the <text> entity. Additionally it has the following attributes:

- type: <integer> mandatory. Defines the type of string to be displayed.
    - 0: 1986,...2013 four digits
    - 1: 13,14 .. last two digits only
    - 2: 0,1,2...9 last digit only
    - 3: 1,2..9,A,B,C hexadecimal according ISO3779
- offset: <integer> default="0". The offset in days or months (depending on the format) added/subtracted to the actual date.
- minutes: <integer> [-32767,32767] default="0". An offset in minutes that is added to the actual date.
- expiry: <integer> [0,1,2] default="0". Sets a special format for calculating the month to be displayed.
    - 0: standard format. Offset is given in days.
    - 1: date always rounded to monday.
    - 2: offset interpretated in months.
- format: <integer> default="0". Sets some flags for the date format. Note that this value must be given as an integer and not in hexadecimal.
    - 0x00000001: Hijri. Calculates the day according the hijri calendar.
    - 0x00010000: Displays arabic digits.
    - 0x00020000: Displays east-arabic digits.
    - 0x00030000: Displays indic digits.

Mandatory entities within a <year> entity:

- <generic>
- <font> when the texttype is "char" or "barcode" with human readable characters displayed.
- <barcode_1d> when texttype is "barcode"
- <barcode> when texttype is "barcode" or "2dcode".
- <rasterscan> when texttype is "2dcode".

Optional entities within a <year> entity:

- <mask>
- <fill>
- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the entity.
- <dbfile> An optional entity for reading the date characters from a file.

<week...attributes>
..
</week>

The week entity represents an automatic text displaying the week of the year. Attributes: The <week> entity heritates all attributes from the <text> entity. Additionally it has the following attributes:

- type: <integer> mandatory. Defines the type of string to be displayed.
    - 0: Displays the week of the year according the the USA standard.
    - 1: Displays the week of the year according ISO standard.
- offset: <integer> default="0". The offset in days or months (depending on the format) added/subtracted to the actual date.
- minutes: <integer> [-32767,32767] default="0". An offset in minutes that is added to the actual date.

- expiry: <integer> [0,1,2] default="0". Sets a special format for calculating the month to be displayed.
    - 0: standard format. Offset is given in days.
    - 1: date always rounded to monday.
    - 2: offset interpretated in months.
- format: <integer> default="0". Sets some flags for the date format. Note that this value must be given as an integer and not in hexadecimal.
    - 0x00000001: Hijri. Calculates the day according the hijri calendar.
    - 0x00010000: Displays arabic digits.
    - 0x00020000: Displays east-arabic digits.
    - 0x00030000: Displays indic digits.

Mandatory entities within a <week> entity:

- <generic>
- <font> when the texttype is "char" or "barcode" with human readable characters displayed.
- <barcode_1d> when texttype is "barcode"
- <barcode> when texttype is "barcode" or "2dcode".
- <rasterscan> when texttype is "2dcode".

Optional entities within a <week> entity:

- <mask>
- <fill>
- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the entity.
- <dbfile> An optional entity for reading the date characters from a file.

<hour...attributes>
..
</hour>

The hour entity represents an automatic text displaying the hour. Attributes: The <hour> entity heritates all attributes from the <text> entity. Additionally it has the following attributes:

- type: <integer> mandatory. Defines the type of string to be displayed.
    - 0: 0..23
    - 1: 1..12
    - 2: AM/PM
- minutes: <integer> [-32767,32767] default="0". An offset in minutes that is added to the actual date.

Mandatory entities within a <hour> entity:

- <generic>
- <font> when the texttype is "char" or "barcode" with human readable characters displayed.
- <barcode_1d> when texttype is "barcode"
- <barcode> when texttype is "barcode" or "2dcode".
- <rasterscan> when texttype is "2dcode".

Optional entities within a <hour> entity:

- <mask>
- <fill>

- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the entity.
- <dbfile> An optional entity for reading the date characters from a file.

<minutes...attributes>
..
</minutes>

The minutes entity represents an automatic text displaying the minutes. Attributes: The <minutes> entity heritates all attributes from the <text> entity. Additionally it has the following attributes:

- minutes: <integer> [-32767,32767] default="0". An offset in minutes that is added to the actual date.

Mandatory entities within a <minutes> entity:

- <generic>
- <font> when the texttype is "char" or "barcode" with human readable characters displayed.
- <barcode_1d> when texttype is "barcode"
- <barcode> when texttype is "barcode" or "2dcode".
- <rasterscan> when texttype is "2dcode".

Optional entities within a <minutes> entity:

- <mask>
- <fill>
- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the entity.
- <dbfile> An optional entity for reading the date characters from a file.

<seconds...attributes>
..
</seconds>

The seconds entity represents an automatic text displaying the seconds. Attributes: The <seconds> entity heritates all attributes from the <text> entity.

Mandatory entities within a <seconds> entity:

- <generic>
- <font> when the texttype is "char" or "barcode" with human readable characters displayed.
- <barcode_1d> when texttype is "barcode"
- <barcode> when texttype is "barcode" or "2dcode".
- <rasterscan> when texttype is "2dcode".

Optional entities within a <seconds> entity:

- <mask>
- <fill>
- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the entity.
- <dbfile> An optional entity for reading the date characters from a file.

```
<counter...attributes>
..
</counter>
```

The counter entity represents an automatic counter displaying the actual value. Attributes: The <counter> entity heritates all attributes from the <text> entity. Additionally it has the following attributes:

- type: <integer> [0,1,2,3] mandatory. Defines the type of counter.
    - 0: global extern: the counter is a global number independent of the message. It is identified by its fieldnumber [0-15] and can only be changed using the graphical PC software. It is global with respect to the PC software.
    - 1: global intern: the counter is a global number independent of the message. It is identified by its fieldnumber [0-15] and its value is global with respect to the laser. Its value can be changed using the handheld-terminal, over RS232 or TCP/IP.
    - 2: local: the counter is local with respect to the message. When the message is loaded the initial value is always the same fixed value. When the message is reloaded, the counter is reseted again to its initial value.
    - 3: locally stored: the counter is local with respect to the message. When the message is about to be changed the value is stored, thus, reloading a message does not reset the counter value. The value is saved each time when the message is reloaded or another message is loaded or the system detects any kind of alarm. A sudden breakdown of the AC/DC power will usually result in a lost counter value.
- field:<integer>[0...15] mandatory for global intern and extern counters. Defines the field number for a global counter.
- start:<63-bit unsigned integer> mandatory. The reset value of the counter.
- end:<63-bit unsigned integer> mandatory. The end value of the counter. The next value loaded would be the start value.
- number:<63-bit unsigned integer> mandatory. The actual value of the counter when the message is loaded.
- step:<integer> mandatory. The increment value of the counter. After each print the counter value is incremented (>0) or decremented(<0) by the step value.
- repeat:<integer> default="0". A non-negative value defining the number of marks until the counter is incremented by the step value.
- shiftreset:<integer>[0,1] default="0". When set to "1" the counter will be resetted when there exists a <shift> entity inside the message and the actual value of the <shift> entity is changing.
- truncate: <integer>[0,1] default="0". When set to "1" leading zeros are truncated.
- digitstring: <string> default="". Defines the characters to be used for displaying the digits of the counter. Please refer to the help for the details.
- base: <positive integer> default="10". The base of the counter. A base of 2 would result in a binary counter. Please refer to the help for the details.

Mandatory entities within a <counter> entity:

- <generic>
- <font> when the texttype is "char" or "barcode" with human readable characters displayed.
- <barcode_1d> when texttype is "barcode"
- <barcode> when texttype is "barcode" or "2dcode".
- <rasterscan> when texttype is "2dcode".

Optional entities within a <counter> entity:

- <mask>

- <fill>
- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the entity.

<shift...attributes>
..
</shift>

The shift entity represents an automatic text that may change within specified time intervals. Attributes: The <shift> entity heritates all attributes from the <text> entity. Additionally it has the following attributes:

- type: <integer> [0,1,2,3,4] mandatory. Defines the type of shift. The format of the <shifttext> entity depends on the type of the shift.
  - 0: time interval given in HH:MM (hour:minute e.g 16:23, 23:00...)
  - 1: time interval given in MM (month e.g. 01,02,..12)
  - 2: time interval given in DD (day of month e.g. 01,02,....31)
  - 3: time interval given in MM:DD (month:day where month is 01...12 and day 01...31)
  - 4 time interval given in WD::HH:MM (dayofweek:hour:minute where dayofweek is 01...07, hour 00..23 and minute 00:59; day 01 is a monday)
- count:<integer> mandatory. Defines the number of <shifttext> entities in this shift entity. The number of <shifttext> entities in this shift entity MUST MATCH the count value !

Mandatory entities within a <shift> entity:

- <generic>
- <font> when the texttype is "char" or "barcode" with human readable characters displayed.
- <barcode_1d> when texttype is "barcode"
- <barcode> when texttype is "barcode" or "2dcode".
- <rasterscan> when texttype is "2dcode".
- <shifttext> For each interval there must be exactly one <shifttext> entity.

Optional entities within a <shift> entity:

- <mask>
- <fill>
- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the entity.

<shifttext...attributes>
..
</shifttext>

The shifttext entity is an entity of the shift entity. It represents exactly one time interval and the coded string for this time interval to be displayed. Attributes:

- shiftdate: <string> mandatory. The formated string according to the type of the shift entity.
  - type 0: The string must be HH:MM
  - type 1: The string must be MM
  - type 2: The string must be DD
  - type 3: The string must be MM:DD
  - type 4: The string must be WD:HH:MM

The string of the <shifttext> entity is encoded as the content of the <shifttext> tag. This content is in any case a CDATA section and UTF-8 coded. The string should not contain any LF character ! If your string needs to have a newline character, just use one CR character for each newline. Typically, for Window systems a newline is programmed as CR + LF, for Unix-like systems a newline is programmed as LF only. Thus, within the CDATA section our convention is not to used any LF character and use as a newline character the single CR character.

An example for the <shift> entity with two intervals is as follows:

```
.................
        <shift type="0" count="2" x="30500" y="73700" texttype="chars" spotreduction="300" id="17">
            <shifttext shiftdate="12:00">
                <![CDATA[a]]>
            </shifttext>
            <shifttext shiftdate="13:00">
                <![CDATA[b]]>
            </shifttext>
            <generic layer_id="0" printable="1" editable="1" linewidth="0" />
            <mask hexvalue="0x2" />
            <font type="6" facename="Arial Regular" synonym="arial.ttf" faceindex="0">
                <metric height="100" ratio="1.000000" separation="100" linespacing="1000" fontflags="0x0" 
align="0" italic="0" bold="0" />
            </font>
        </shift>
...................
```

<usermessagetext...attributes>
..
</usermessagetext>

The usermessage entity represents a variable text string associated to a fieldnumber. The content (the string) of a fieldnumber resides globally in the laser's memory or in the graphical PC software, depending on the type of the usermessage and can be actualized via RS232 or TCP/IP. Attributes: The <usermessage> entity heritates all attributes from the <text> entity. Additionally it has the following attributes:

- type: <integer> [0,1] mandatory. Defines the type of the usermessage.
    - 0: The content resides globally in the PC laser software. It can only be used in combination with the PC software.
    - 1: The content resides inside the laser's global memory. It can be actualized via RS232 or TCP/IP.
- field:<integer> [0..255] mandatory. Defines the field number of this usermessage.

Mandatory entities within a <usermessagetext> entity:

- <generic>
- <font> when the texttype is "char" or "barcode" with human readable characters displayed.
- <barcode_1d> when texttype is "barcode"
- <barcode> when texttype is "barcode" or "2dcode".
- <rasterscan> when texttype is "2dcode".

Optional entities within a <usermessagetext> entity:

- <mask>
- <fill>
- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the entity.

The string of the <usermessagetext> entity can be encoded as the content of the <usermessagetext> tag, exactly in the same way as for a <text> entity. However, this string will only be read by the laser if the corresponding field number is still not initialized. For this reason the string is usually not encoded (remember that the string is a global string ,shared over different messages and identified just by the fieldnumber).

<groupedtext...attributes>
..
</groupedtext>

The groupedtext entity acts as a container for all other text entities. Attributes: The <groupedtext > entity heritates all attributes from the <text> entity and has no further attributes.

Mandatory entities within a <groupedtext > entity:

- <generic>
- <font> when the texttype is "char" or "barcode" with human readable characters displayed.
- <barcode_1d> when texttype is "barcode"
- <barcode> when texttype is "barcode" or "2dcode".
- <rasterscan> when texttype is "2dcode".
- <objects> at least one of the elementary text entitites <text>, <year>, <month>, <day>, <week>, <hour>, <minutes>, <seconds>, <counter>, <usermessagetext>, <shift> must be embedded inside the objects entity of the groupedtext entity.

Optional entities within a <groupedtext > entity:

- <mask>
- <fill>
- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the entity.

The string displayed by a groupetext entity results from the concatenation of its objects entities in the order of their appearance. Note that the object entities of a groupedtext entity do no need to have their own , usually mandatory, <font> <generic> <barcode_1d> <barcode> <rasterscan> entities, nor do they have the optional entities <mask> <fill> <transformation>.

Example for a <groupedtext> entity:

```
........................
      <groupedtext x="40500" y="71100" texttype="chars" spotreduction="300" id="18">
         <generic layer_id="0" printable="1" editable="1" linewidth="0" />
         <mask hexvalue="0x2" />
         <font type="6" facename="Arial Regular" synonym="arial.ttf" faceindex="0">
            <metric height="100" ratio="1.000000" separation="100" linespacing="1000" fontflags="0x0"
align="0" italic="0" bold="0" />
         </font>
         <objects>
```

```
        <hour type="0" minutes="0" texttype="member" id="18" />
        <text texttype="member" id="18">
          <![CDATA[ : ]]>
        </text>
        <minutes minutes="0" texttype="member" id="18" />
        <text texttype="member" id="18">
          <![CDATA[ : ]]>
        </text>
        <seconds texttype="member" id="18" />
      </objects>
    </groupedtext>
```

....

<vectorgroup...attributes>
..
</vectorgroup>

The vectorgroup entity acts as a container for other elementary geometric entities <line>, <polyline>, <arc>, <ellipse> or <rectangle>. Attributes:

- id: <positive integer> . A unique identifier for this element.
- name <string>, default="". An optional name for this object that may be used as a reference for eventhandler actions. This attribute is not applicable for the firmware program (only for version <= 5.3.1) as by default the firmware program decomposes any vectorgroup into its elementary entities.

Mandatory entities within a <vectorgroup > entity:

- <generic> Only the layer attribute of the generic entity is a valid attribute. The layer attribute is heritated to all elementary geometric entities of the <vectorgroup> entity.

Optional entities within a <vectorgroup> entity:

- <mask>
- <fill>

<image...attributes>
..
</image>

The image entity represents a 8-bit grayscale bitmap that is printed in pixel mode line by line or column by column. Attributes:

- x: <integer> mandatory. The x position of the topleft point of the image in ideal units.
- y: <integer> mandatory. The y position of the topleft point of the image in ideal units.
- width:<integer> mandatory. The width of the image in ideal units.
- height:<integer> mandatory. The height of the image in ideal units. (x,y) and (width,height) define a rectangle. When an <image> entity is loaded into memory the image data are stretched so that they fit exactly into this rectangle. Any further scaling, stretching or rotation must be done with the <transformation> entity.
- id: <positive integer> . A unique identifier for this element.
- name <string>, default="". An optional name for this object that may be used as a reference for

eventhandler actions.

- file <string>, default="". Defines the path and filename of the image's pixeldata if they are stored seperatedly. Image data stored seperately must be stored in a propietary format explained later.
- type <integer>[0,1]: Defines the type of the image entity.
  - 0: (default) The image data are embedded within the <image> entity or in a seperate file.
  - 1: The image data are read from the internal datastring buffer which is actualizable via TCP/IP or RS232 commands.
- xpixel <unsigned integer> mandatory: The number of ´horizontal' pixels.
- ypixel <unsigned integer> mandatory: The number of ´vertical' pixels.
- encode<integer> [-1,0] mandatory: When set to '-1' then the image data are not embedded in the <image> entity and are present in a seperate file. In this case the file attribute must have been set accordingly to the full path of the image datafile. If no path is given the internal standard path to the internal harddisk is assumed. When set to '0' the image's pixeldata are embedded in the entity as the content of the <image> tag. This content is in any case a CDATA section and UTF-8 coded. Image data are base64 encoded. The image must be a grayscale image where each pixel is coded as a 8-bit color value. The pixel data must be encoded in the following order: start with the bottom-left pixel of the image and scan from left to right. Then move one pixel up and scan again from left to right and so on. Modified Base64 encoding for UTF-7 (RFC 1642) is used as encoding scheme. This means that '+' is used for char index 62 and '/' for char index 63. No line separators are used. For type 1 images where data are read from the internal Datastring buffer, the encode attribute is ignored.
- field <integer> [0,1...5] mandatory for type 1 images: Defines the field number of the internal Datastring buffer from where the pixel data are read.
- bpp <integer>[1, 8] mandatory for type 1 images: Defines the encoding of the data in the datastring buffer. The datastring buffer stores the pixel's color data. The order is the same as for the embedded base64 encoding starting from the bottom-left, scanning to the right and move one pixel up. When bpp is set to "8" each character of the datastring is the color of one pixel (8 bits per color). When set to "1" (1-bit per pixel) then one character (1-byte) contains the color of 8 pixels where the order is from MSB to LSB. A bit value of 0 means a black color, abit value of 1 a white color.

Mandatory entities within a <image > entity:

- <generic>

Optional entities within a <image > entity:

- <transformation> An optional transformation entity allows to apply any linear matrix operation to be applied to the image. The reference or invariant point for the transformation is the topleft point of the image .
- <rasterscan>

Image Data file format

When the image data are not embedded within the <image> entity, a valid filename must be set for the 'file' attribute and the file has to be stored inside the laser. Usually, a copy command stores the file sent to the laser inside the harddisk in the '.\bin\' directory and the 'file' attribute just contains the filename. If you decide to send the image data file only to the ramdisk of the laser, you would have to preceed the filename with '.\ram\'. We recommend a 'bm' file extension although any file extension that may not confuse with some standard file formats can be used. The format is binary and as follows:

Header:

```
0xXXXXXXXX   4 bytes : format, must be 0x00000000
0xXXXXXXXX   4 bytes: nx, number of horizontal pixels
0xXXXXXXXX   4 bytes: ny, number of vertical pixels
```

Pixel color data:

```
0xXX 0xXX 0xXX .........ny bytes of the first vertical pixel line starting from left-bottom.
0xXX 0xXX 0xXX .........ny bytes of the second vertical pixel line.
......
.....
0xXX 0xXX 0xXX .........ny bytes of the last = (nx -1)th vertical pixel line.
```

Thus, the image data file is a kind of grayscale image where each pixel is represented by a 1-byte color value.

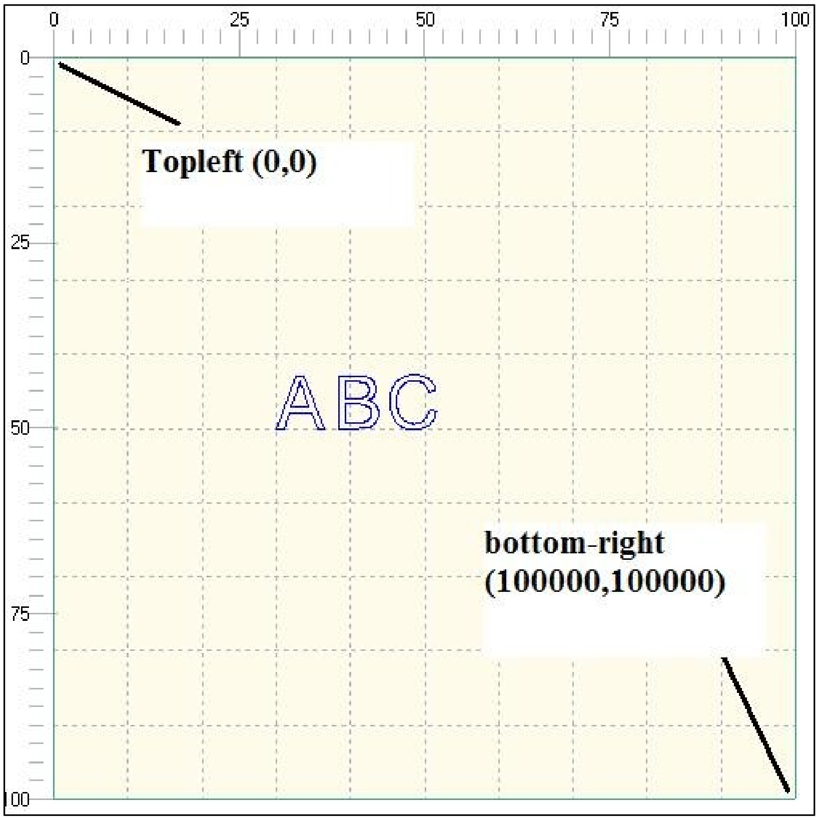
Font rendering

For third-party application that may create xml-files for laser marking it is quite important to understand how the objects, especially text, are rendered. For the elementary geometric entities like <line> <polyline> <rectangle> <arc> <ellipse> rendering is quite evident and needs no further explanations. Here we will explain the basic text rendering rules applied for truetype fonts and the single-stroke laser fonts. We refer basically to the <text> entity but the following explanations are valid also for the other text type entities like <hour> <seconds> etc...
We will start with a simple xml example of a <text> entity:

```
<text x="30000" y="50100" texttype="chars" spotreduction="300" id="0">
        <generic layer_id="0" printable="1" editable="1" linewidth="0" />
        <mask hexvalue="0x2" />
        <font type="6" facename="Arial Regular" synonym="arial.ttf" faceindex="0">
           <metric height="100" ratio="1.000000" separation="100" linespacing="1000" fontflags="0x0" 
align="0" italic="0" bold="0" />
        </font>
        <![CDATA[ABC]]>
</text>
```

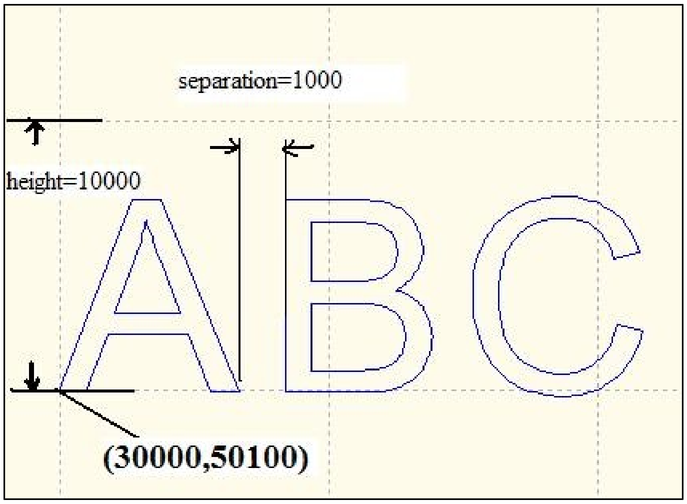
This is a simple "ABC" text using the "Arial Regular" truetype font. The insert- or reference point of the text object is (30000,50100) and it is left aligned. Rendering this text object on the screen would be done as follows:

The screen is a virtual squared view with a size of 100000 x 100000 expressing the ideal coordinate system. The top left point of this view has the coordinate (0,0) and the bottom-right point has the coordinate (100000,100000).



The reference point of the object (30000,50100) is on the left-bottom side of the text as it is left aligned. The metric height=100 translates to a height of 10000 (see the height attribute of the <text> entity). The height of a font is interpretated in the following way:

When the font was designed, e.g. a truetype font, the font designer draws the glyphs of a character in a virtual box of a specific size and stores all the glyph data with respect to this size, i.e. using the coordinate system of this virtual box. This virtual box size is sometimes called the EM_size when treating with Freetype font rendering mechanisms. The EM_size determines the designer units that has choosen the designer for drawing the characters. When we set a font height of 100, this results in a font height of 10000 ideal units. This 10000 ideal units now map to the EM_size of the font itself. thus, independently of what design units the font was designed with the font will be rendered in the same way. The reference point is the (0,0) point of the virtual box and from there on the character is drawn as stored inside the font file. Once a character is rendered the origin for the next character is obtained by moving the reference point horizontally to the right side until it reaches the rightest position of the enclosing rectange of the previous character. From there an additional separation distance is moved (separation attribute) to the right direction and the (0,0) origin for the next character is fixed.



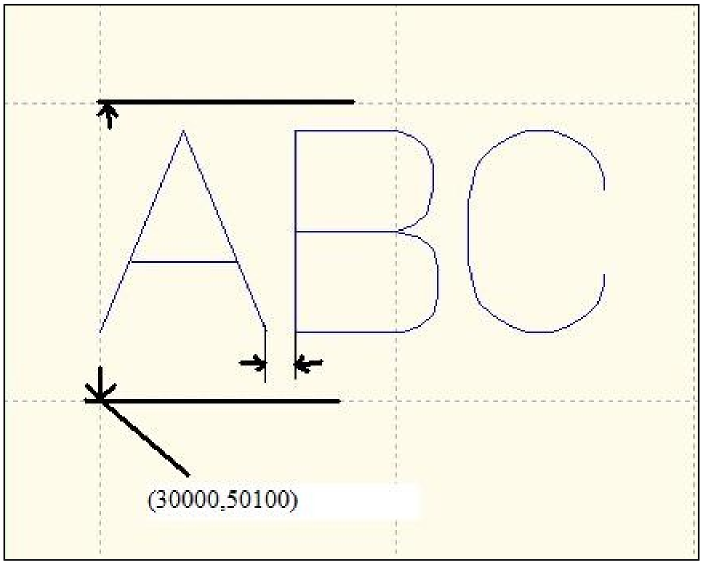
If kerning is enabled (only available for freetype fonts) the origin for the next character is obtained from the kerning table of the font itself. Thus, the origin of the next character is simply obtained from the origin of the previous character by adding the offset obtained from the kerning table. The separation attribute is then used always as an additional horizontal offset. Typically, when using kerning, the spearation attribute should be set to 0, thus the distance between the characters is automatically adjusted due the values of the kerning table which is intrinsic to the font file.

For the single-stroke fonts (laserfont, laserfonts extended, crystall font, crystall font extended) the rendering works in a similar way. The "EM_size" is stored in the header of the font file and its value is usually 255 or 1024. Note that y-coordinate of the raw font data needs to be inverted. For all mfs-fonts and unx-fonts, the characters are not "base-aligned" with the reference points, as the (0,0) origin in the raw font data does not coincide with the "baseline" of the characters. The (x,y) points of the stroke lines within the font file may range from 0 to EM_size, but are not restricted to this range. The y-coordinate needs to be inverted. According the defined font-height a correction factor has to be applied to the (x,y) points of the font data to get (x,y) points in ideal coordinates. This factor is calculated as

$$f = (\mathrm{Fontheight}*100.0)/\mathrm{EM\_size}.$$

The 'ratio' parameter of the font metric has to be applied to the x-coordinate. Thus, the ideal (x,y) coordinates are calculated from the font (x,y) coordinates by

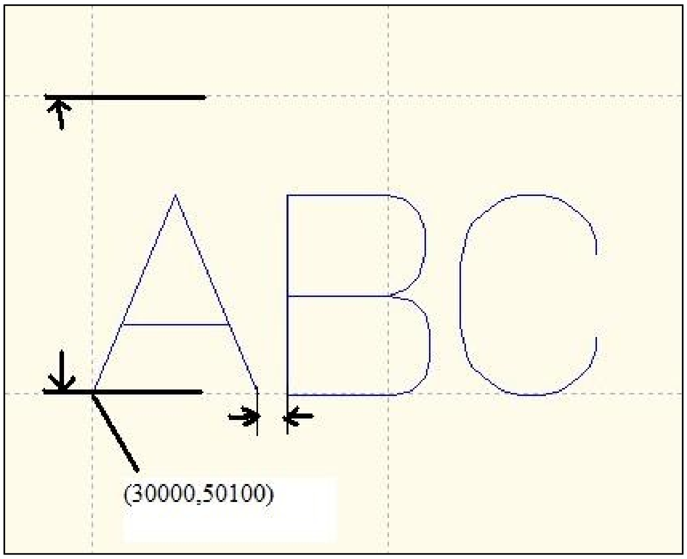
$$x\,(\mathrm{ideal}) = x\_\mathrm{reference} + x(\mathrm{font}) * \mathrm{ratio} * f \qquad y(\mathrm{ideal}) = y\_\mathrm{reference} - y(\mathrm{font}) * f$$



For the unf-fonts the "baseline" y-coordinate is stored seperately in the header of the font file and is used to correct the y-position of the character, so that the characters are base aligned to the reference point.
Thus

$$x\,(\mathrm{ideal}) = x\_\mathrm{reference} + x(\mathrm{font}) * \mathrm{ratio} * f \qquad y(\mathrm{ideal}) = y\_\mathrm{reference} - (y(\mathrm{font}) - \mathrm{baseline\_y}) * f$$

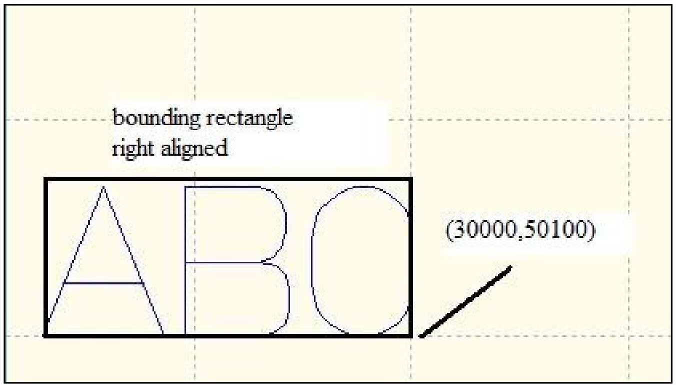
The 'separation' parameter of the metric is used to go to the reference point of the next character. If the character is a 'space', an additional 'separation' is applied if the 'Fixed Spacesize' is not enabled. If the 'Fixed Spacesize' is enabled then the additional advance for a 'space' character is calculated as follows x_advance(ideal) = (EM_size/4) * ratio * f * spacesize



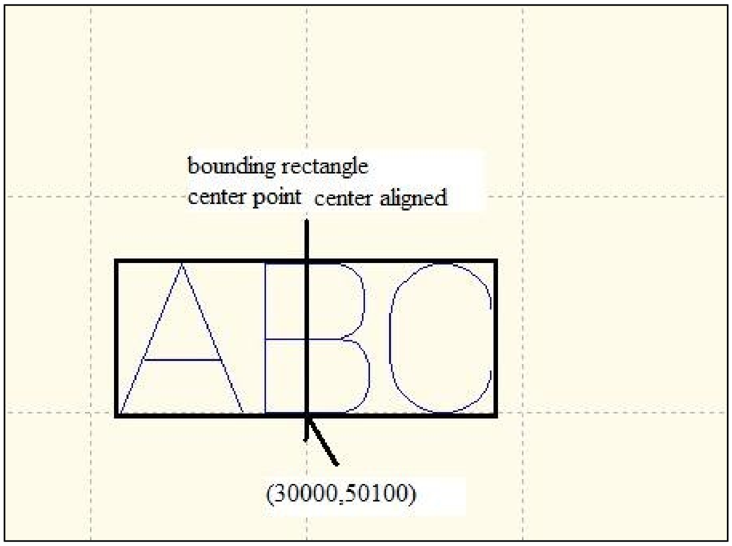
More information about the single-stroke font file format see mfs file format and unx/unf file format.

Text alignment

Per default the alignment is left-aligned with respect to the reference point. When the alignment is set to right aligned (see align attribute) , rendering is done as in left-alignment but at the end of the complete text rendering the whole text is shifted to the left so that the rightest point of the text coincides with the x position of the reference point.



If the text is center-aligned the whole text is x-positioned so that the bounding rectangle's center point coincides with the x-coordinate of the reference point.


bounding rectangle
center point
center aligned
ABC
(30000,50100)