Word and Excel document generation with Posity Neo

Content

1. Document generation with Posity Neo

This document describes how Posity Neo can be used to create individual and sophisticated Word and Excel documents. To create a document, you first have to create a template in Word (.docx) or Excel (.xlsx). The layout can be freely designed with the means of Word or Excel. Placeholders (e.g. for the customer's name) can then be inserted into this template, which are then replaced by Posity Neo. The **placeholders** must always be enclosed in curly brackets (e.g. '`{Partner.CustomerName}`'). Posity Neo then replaces these placeholders with the concrete values in processing. In Excel, multiple sheets of the workbook can also be used in the template.

Definition of terms: The terms *list, sheet, table* are understood in the following explanations as follows:

1. Table: The term table refers to the input data that is processed by Posity Neo – not a Word or an Excel spreadsheet.
2. List and Sheet in Word:
   When the term list is used in connection with Word, it refers to a 'normal' Word table (filled with Posity Neo placeholders). It's not an Excel spreadsheet built into Word. The term sheet has no meaning here.
3. List and sheet in Excel:
   The sheet refers to a sheet within an Excel workbook. This sheet is often referred to as a table – which is avoided here (the table contains the input data, see point 1). By list we mean a contiguous part of rows within a sheet (e.g. lines 5 to 12) that are to be filled with data. Each row of such a list is indicated by the placeholders `{Row}`, `{FooterBreak}`, `{HeaderBreak}` or `{Spacer}` (the effect of which will be explained later) in the first column. A single Excel sheet can therefore contain several lists.

Placeholders in lists: By means of additional placeholders (e.g. `{DecimalSeparator=,}` to set the decimal separator), you can control the layout of the document. Certain placeholders also influence the layout of lists. These placeholders can be used directly in Excel templates. In Word documents, these placeholders must be used within lists.

Placeholders in worksheet name: (Excel only) Can be used with the following limitations:

1. The placeholder must not be longer than 31 characters (limitation of Excel)
2. The value used for the placeholder is truncated at 28 characters

Tags of placeholders: Certain placeholders can be supplemented by one or more tags (information). These tags must always be marked using | (Pipe) can be separated from each other, e.g. `{CustomerOffer.CashDiscount|Format:"Discount: "##,##0.00"% "|Zero:}`.

Mail merge function: Documents can also be created that require several input tables. For example, the input tables customer, quotation and quotation items can be processed in a document. It is also possible to generate quotations and quotation items in 'individual' documents (e.g. one document per offer) for several customers at the same time (similar to the mail merge function of Word).

Field functions: Word field functions can also be used in the generated Word documents (see also placeholder {UpdateFields}). Placeholders can be used within the field functions, which are then to be applied after the document has been generated, e.g. **{**IF "`{Question.Selected}`"="True" "☒" "☐"**}** (bold curly braces are part of the Word field function).

Protection against changes: If you want to protect the generated Word document from changes, you can protect the template from changes with a password. The generated document is read-only as well as the corresponding password in the generated document.

2. General rules of syntax notation

The following syntax rules must always be followed:

- Spaces are NOT allowed, unless they are explicitly desired in format definitions.
- Upper and lower case must be observed
- No control characters (Paragraph ¶, NewLine ↲, TabChar →) can be used. To do this, the placeholders {NewParagraph}, {NewLine} and {TabChar} must be used (see non-terminal symbol StringWithBreak and «3. Syntax and semantics of placeholders»).

Explanation of syntax notation:

- `:==` defines a placeholder
- `[ ]` the inner part is optional
- `{ }` the inner part can be used as often as you like (not even once)
- `()` Outline for delimitation or for better comprehension of the definition – but may not be included in the template (in orange color)
- `|` selection (or); Text before or after the pipe must be used; several options are combined (e.g. {CO. CashDiscount|Format:"Discount: "##,##0.00"% "|Zero:}, then the options are applied in order of mention.
- `";"` the inner part (in this case the semicolon) must be copied into the template as a text constant (in blue color)
- `<>` the contents of this symbol (non-terminal icon) are defined for the syntax (next chapter) elsewhere, in the immediately following list below:

The following non-terminal symbols (these symbols no longer appear as "words" in the generated, final placeholders) are used in the following syntax notation to define the placeholders:

- `AbsolutPathFileAndExtension :== [<AnyString>]["{"< attributes >"}"][<AnyString>]`
  Path specification according to Windows standard
- `AnyString :==` Any string including special characters but without { and }.
- `Attributes :== (<TableName>|<TableAliasName>)"." <AttributeName>`
- `AttributeName :== <String>`
- `AttributeWithNr :== ("{"< Attribute >"}")|<Number>` Attribute value must contain a number
- `Bold :== <Boolean>`
- `Boolean :== "true"|"false"`
- `Color :== "0".."255"` Integer values between 0 and 255
- `ColorB :== Color`
- `ColorG :== Color`
- `ColorR :== Color`
- `ComparisonOperator :== "<="|">="|"<"|">"|"<>"|"="`
- `Condition :== ("{"<Attribute>"}"|"""<String>"""|<Number>)<ComparisonOperator> ("{"<Attribute>"}"|"""<String>"""|<Number>)`
  - The data types of the comparison must match.
- `FontName :== <String>` Valid name of a font.
- `FontSize :== <Number>`
- `FormatPattern :== <AnyString>`
  - Here the formatting rules are allowed according to the specification of the C# output format format command.
  - Any character can be used in the format, including tabs, line breaks, and page breaks (the effect depends on whether it's a Word or Excel template). Caution: If characters used can be mistakenly interpreted as part of the formatting information, an escape character ("\") must be prefixed.

  - Decimal separators must always be written as periods, thousands separators as commas (see also placeholders DecimalSeperator and ThousandSeparator).
  - For numbers, either a single format can be specified, or two formats can be specified (for the positive number and the negative number), or three formats can be specified (for the positive number, for the negative number, and for the 0 case). The different cases are separated by a semicolon (;)). If the formatting contains text constants before or after the number (as in the example below) and the number can take negative values, then the formatting must always be specified for the negative value as well, otherwise the minus sign will be placed before the first text constant. Examples:
    - A format: '"Temperature: "#,##0.00" C°"'
    - Two formats: '"Temperature: "#,##0.00" C°";" Temperature: "-#,##0.00" C°"'. Note that the ENTIRE format must be repeated before and after the semicolon.
    - Three formats: '"Temperature: "#,##0.00" C°";" Temperature: "-#,##0.00" C°";" *none*"'. Note that the ENTIRE format must be repeated before and after the semicolons.
  - In the case of a duration, the times can also be output in industrial format. To do this, a "[D]" (days), "[H]" (hours), "[m]" (minutes) or "[s]" (seconds) must be placed in front of the actual format. For example, a valid format is "[H]##00.00" (e.g. 08.75 hours). Cannot be used in Excel templates.
- Italic :== <Boolean>
- Level :== <Digit> values between 0 and 8 are allowed.
- Number :== <Digit>["."<Digit>]
- Digit :== ("0"|..|"9"){"0"|..|"9"}
- Pixels :== <Digit>
- String :== ("a"|..|"9"){"a"|..|"9"}
- StringWithBreak :== {<AnyString> | <NewParagraph> | <NewLine> | <TabChar> | <FormatParagraph>}
- SymbolShortcut :== "o"|"O"|"oo"|"OO"|">"|"<>"|"[]"
- SymbolUnicode :== <Digit>[<Digit>][<Digit>][<Digit>][<Digit>] Values for the Unicode characters according to the symbol table of Word are allowed.
- TableAliasName :== <TableName> TableAliasName must be defined using TableAlias assignment.
- TableName:== <String>
- Underscore :== <Boolean>

3. Syntax and semantics of placeholders

```
AttributeValue :==
     "{"<attributes>
     ["|ColorCondition:"<Condition>";"<ColorR>","<ColorG>","<ColorB>]
     ["|FontCondition:"<Condition>";"<FontSize>","<Bold>","<Italic>","<Underscore>]
     ["|DependOn:<attributes>]
     ["|Format:"["""<StringWithBreak>"""]<FormatPattern>["""<StringWithBreak>"""]]
     ["|MaxWidth:"<pixels>]
     ["|MaxHeight:"<Pixels>]
     ["|FitRow:"("Yes"|"Nope")]
     ["|OldValue"]
     ["|NonZero:"<StringWithBreak>]
     ["|Zero:"<StringWithBreak>]
     ["|True:"<StringWithBreak>]
     ["|False:"<StringWithBreak>]
     "}"
```

- The placeholder is replaced by the value in the data to be processed, e.g. '{CustomerInvoice.InvoiceDate}' is replaced by the date of the invoice (e.g. 10/10/2016).
- Placeholder position: At the point in the template where you want the value to be.
- The inserted value of the attribute must match the formal definition of the non-terminal symbol <StringWithBreak>.

- **ColorCondition**: If the condition is met, the attribute is displayed in the corresponding color (RGB), e.g. '{CustomerInvoice.InvoiceDate|ColorCondition:{P.NamePartner}="Posity AG"; 255,50,0}'. In comparison, only attributes of the current data set can be used. Both attributes must be of type String (all types except Numeric are used as strings) or of type Numeric. If the comparison is not possible (wrong types, wrong attribute), the condition is ignored. Capitalization is ignored in the evaluation of the comparison operation! Attributes must be enclosed in curly brackets.
- **FontCondition**: If the condition is met, the attribute is formatted with the appropriate information, e.g. '{CustomerInvoice.InvoiceDate|ColorCondition:{P.NamePartner}="Posity AG"; 10,true,false,false}'. In comparison, only attributes of the current data set can be used. Both attributes must be of type String (all types except Numeric are used as strings) or of type Numeric. If the comparison is not possible (wrong types, wrong attribute), the condition is ignored. Capitalization is ignored in the evaluation of the comparison operation! Attributes must be enclosed in curly brackets.
- **DependOn**: If the value of the attribute referenced here is 0 or empty, the attribute is suppressed – an empty string is output, e.g. 'DependOn:CustomerInvoice.CashDiscount'.
- **Format**: In the <FormatPattern> the output format can be set according to the specification of the C# format command, supplemented by a preceding and subsequent string (e.g. '"Date: "dd.MM.yyyy', or '#,##0.00' or '#,##0" \%"'). See also description of <FormatPattern>.
- **MaxWidth** and **MaxHeight** are only possible for images! If the image exceeds one of the specified sizes, it is proportionally reduced in size until it reaches the specified maximum size (aspect ratio is maintained). Only a size limit can be used. The unit is in DPI (100 is therefore equivalent to 2.54 cm). For example, 'MaxWidth:400' ensures that the image is a maximum of 400 DPI, or 5.16cm wide, 'MaxWidth:400|MaxHeight:200' ensures that the image is a maximum of 400 DPI wide and a maximum of 200 DPI high. In Word, the mechanism for positioning the image is controlled by the template, in Excel the image is placed so that the corners at the top left of the image and the corresponding cell are in the same position.
- **FitRow** is only possible for images (and only for Excel) and sets the height of the row/cell in which the image is located to the height of the image. Example: '|FitRow:Yes'. Depending on the system setting of the scale, there may be problems with the correct calculation of the height, so please check if the 'custom scaling' of Windows is set to 100%.
- If OldValue **is** specified, the value of the previous input record is output. If it is the very first record, the placeholder is left blank.
- **NonZero**: For attributes of type Number or String, you can specify what should be output if the value is not 0 or the string is not empty (e.g. line break). For example, with 'NonZero:not zero', instead of a number, the text 'non-zero' is displayed, unless the value of the placeholder is 0. If you want to suppress the line completely, you can use the RemoveRow placeholder (placeholder, not tag).
- **Zero**: For attributes of type Number or String, you can specify what to output if the value is 0 or the string is empty. If you don't want the number 0 to be displayed, you can also suppress it with this attribute: {CO. CashDiscount|Zero:}. If you want to suppress the row entirely, you can use the RemoveRow placeholder (placeholder, not tag).
- **True**: For attributes of type Boolean/String, it is possible to specify what should be output if the value is 'True', e.g. 'True:selected'. If you want to suppress the row entirely, you can use the RemoveRow placeholder (placeholder, not tag).
- **False**: For attributes of type Boolean/String, it is possible to specify what should be output if the value is 'False', e.g. 'False:logged out'. If you want to suppress the row entirely, you can use the RemoveRow placeholder (placeholder, not tag) werden.ag
- Example 1: {CI.InvoiceDate|Format:dd.MM.yyyy}
- Example 2: {CO. CashDiscount|Format:"Discount: "##,##0.00"% "|Zero:}
- Example 3: {P.FirstNamePartner|ColorCondition:{P.NamePartner}="Posity AG"; 255,50,0}
- Example 4: {Invoice.Amount|ColorCondition:{Invoice.Amount}<0.0; 255,0,0}

```
Background Color Cell :==
     "{BGC:"<RangeValue>{"|"<RangeValue>}";"<AttributeWithNr>";"<ColorR>","<ColorG>","<
     ColorB>{"|"<ColorR>","<ColorG>","<ColorB>"}"
```

- Specifies the color of a cell background. If the value is within specified limits (Value >= RangeValue(x) and Value < RangeValue(x+1)), the cell background is colored according to the specified color(x).
- Position of the placeholder: In the cell to be colored
- Example 1: {BGC:0|2; 1.2; 0,200,120}
- Example 2: {BGC:0|2|99;{ COI. Quantity}; 0,200,120|0,10,10}

```
BorderCell :==
     "{BOC:"<AttributeWithNr>{"|"<AttributeWithNr>}";"("Line"|"Dot"|"Dash"|"0"|"1"|"2")
     ";"
     ("Thin"|"Medium"|"Thick"|"0"|"1"|"2")"}"
```

- Formats the surrounding border of the cells in the current row at the appropriate column positions (AttributeWithNr) according to the parameters. If no AttributeWithNr is specified (empty list), or if the AttributeWithNr is 0, no border is set.
  ATTENTION: the table must not have any connected cells!
- Position of the placeholder: In the row to be colored (can be in the first column, but format the border of the cell in the third column).
- The values 0, 1 or 2 are 'translated' accordingly (line = 0, dot = 1, etc.)
- Example 1: {BOC:1; Line; Thin} or {BOC:1; 0; 0}
- Example 2: {BOC:1|3|5; Dot; Medium}
- Example 3: {BOC:{CO. Number}; Line; Thin}

```
DecimalSeparator:== "{ DecimalSeparator:"<AnyString>"}"
```

- Specifies which decimal separator should be used in the FormatPattern (the decimal separator is always specified as a period in the format). This setting is used regardless of the operating system and any country settings.
- Placeholder position: Anywhere in the document (optional)
- Example: {DecimalSeparator:,} the comma is used as a decimal separator.
- Cannot be used in Excel templates.

```
DocumentBreak :== "{DocumentBreak:"(<TableName>|<TableAlias>)
      {"|dependence:"<attributes>",""<attributes>"}
      "|PageBreak:"("Yes"|"No")}
```

- This placeholder can be used to create a merge-like layout. In the first place, the master input table (e.g. offer) is specified, for which a 'document' is to be created per record on the basis of the template. If an alias name has been assigned, this alias name must be used.
  - In Word, after processing each record of the master input table, a page break is introduced (the page break itself can be suppressed) and then a Word template is processed again with the next record. For example, if there are 100 quotations in the master input table, the Word template is applied 100 times (the result is a single Word document in which the template has been processed 100 times).
  - In Excel, a new sheet is inserted if the page break is not suppressed, otherwise the next template to be processed is appended to the last used row of the current sheet.
- If dependencies between input tables are to be taken into account in a template (e.g. only the quotation items of the currently processed quotation are to be taken into account), this can be specified in the Document Break placeholder with the dependency attribute (see example below). Any number of dependencies (as primary key foreign key tuples) on the master input table can be specified.
- If PageBreak **is** set to Yes, each new tuple of the master input table is output in Word on a new page, or inserted in Excel on a new sheet, otherwise the data is in the same document or on the same sheet one after the other without a page break.
- Example: {DocumentBreak:CustomerOffer|Dependence:CustomerOffer.PK_CustomerOffer,CustomerOfferItem.FK_CustomerOffer|PageBreak:No}
- Placeholder position: Anywhere in the document (optional)

```
ExportFile :== "{Exportfile:"<AbsolutPathFileAndExtension>
     ["|Open:"("No"|"Yes")]
     "}"
```

- Specifies that the generated document should be saved to a file. Allowed formats of the target file are: docx, doc, txt, rtf, pdf, xps and xml. Depending on the format, display losses must be expected.
- This placeholder has NO effect on the filename of the created file, if the name of the export file is passed when the generation is called using parameters! If a file name is not specified either by parameter or export file placeholder, the generated file receives a default name, generated from the name of the template plus a date-time stamp (e.g. "ProtectionLevel_20151223170849737.docx"). If you want to use the default name, you can specify "default" instead of the file name.
- The AbsolutPathFileAndExtension tag specifies the absolute path, plus filename, plus fileextension of the import file. The name must be specified according to the Windows specification, e.g. "C:\Temp\RechnungDaten.txt".
- If "**Open**:Yes" is specified, the created file will open automatically.
- The export file placeholder is interpreted as the last step in the processing. Therefore, it is possible to use placeholders in the filename (see 2nd example)!
- Example 1: {Exportfile:C:\Temp\Offerte.docx|Open:No}
- Example 2: {Exportfile:C:\Temp\Offerte-{CI.OfferNumber}.pdf|Open:Yes}
- Example 3: {Exportfile:Default|Open:Yes}
- Placeholder position: Anywhere in the document (optional)

```
FooterBreak :==
     "{FooterBreak:"(<TableName>|<Attribute>){","(<TableName>|<Attribute>)}"}"
```

Closing line to a range of identical values in a list. After a value change of the specified attribute, the attribute list, or the end of the specified input table (the attribute name is optional), this row is copied to the end of the list and then the attribute values of the copied row are replaced. Must be in a Word, Excel list in the first column.

- Example: {FooterBreak:COI. NameProductGroup}
- Placeholder position: In the first column of a list. The row contains the data that should be displayed during the wrap.
- Please note: if charts (Bar Chart, Pie Chart) are used, the adjustment of the data ranges of the charts does not work, because FooterBreaks interrupt them. A possible workaround is to insert the data used in the chart on a separate worksheet and relate the data section of the charts to it.

```
FormatParagraphIndentation :== "{FPI:Left:"<Number>",Right:"<Number>","
     ("FirstLine:"<Number>"|"Hanging:"<Number>)
     [",Bullet:"
     (<SymbolUnicode>[","<Level>"],"<Fontname>|
     <SymbolShortcut>[","<Level>])]"}"
```

- Formats the indents of a Word paragraph. Cannot be used in Excel templates. May only be used once per paragraph.
- Typically used in conjunction with the Zero or NonZero tag. However, it may also be included in the imported data (= in the text of an attribute).
- The numbers indicate the indentations in centimeters (cm). The information is given in the following order: Left (Left, left distance to the document margin), Right (Right, right distance to the document margin), first line (FirstLine), or indentation after the first line (Hanging). FirstLine and Hanging cannot be specified at the same time.
- Bullet (Optional): If you want the paragraph to have a bullet point (or a bulleted list), this can be activated by using the bullet keyword. In Word (and in the generated document), bulleted lists are based on bulleted sheets that contain the document bullets at each bullet level, indentation mass, and other formatting data. The bullet parameter therefore assumes that a bulleted sheet with the desired bullet point is included in the Word document in which the ParagraphIndentation tag format with bullet option is used.
  A bulleted sheet with the desired bullet is included in Word if it is visible in the 'Bullets > Document Bullets' drop-down menu. A bulleted sheet becomes visible in the document when it is used in a bulleted list (the

document bulletis still present when the bulleted point is removed from the text) or via the same drop-down menu 'Bullet > Define New Bullet...' , new bulleted sheets can be created, which can then be used as document bullets in the edited Word document. The bullet parameter uses the symbol and font to reference the document bullet to be used. The document bullet is used, where the symbol and font match the specification in the bullet parameter. If identical document bullets occur, the first in the list is used. This means that the document bullet alignment (left, right, or center) of the first document bullet found is always used.

- Symbol Unicode: Any symbol from the symbol library (of Word) can be used to reference a document bullet (i.e. any symbol can be used as an enumeration symbol) by specifying its Unicode value as a decimal value in the tag. The symbol library shows the hex value of a symbol. Example: ༒ 0F12 (Hex) → Calculator App → 3858 (Decimal), Font Name: Microsoft Himalaya: {FPI:Left:0,Right:0,Hanging:0,Bullet:3858,0,Microsoft Himalaya}
- Symbol shortcut: The following symbol shortcuts can be applied (fixed), provided that the bulleted sheet is available in the Word document!
  - ○ Created with {FPI:Left:0,Right:0,Hanging:0,Bullet:o}
  - O Created with {FPI:Left:0,Right:0,Hanging:0,Bullet:O}
  - • Created with {FPI:Left:0,Right:0,Hanging:0,Bullet:oo}
  - ● Created with {FPI:Left:0,Right:0,Hanging:0,Bullet:OO}
  - ➢ Created with {FPI:Left:0,Right:0,Hanging:0,Bullet:>}
  - ❖ Created with {FPI:Left:0,Right:0,Hanging:0,Bullet:<>}
  - ▪ Created with {FPI:Left:0,Right:0,Hanging:0,Bullet:[]}
- Level (optional): Defines the indentation level (default paragraph settings) of the bulleted list. Numerical value 0-8, 0 no indentation (is default), example bullet with indentation level 1: {FPI:Left:0,Right:0,Hanging:0,Bullet:>,1}
- Font (must be specified if a Unicode symbol is used): Name of the font that contains the symbol used. Example: {FPI:Left:0,Right:0,Hanging:0,Bullet:167,0,Walbaum Display Heavy}
- Bullet + Left,Right,Hanging,Firstline: If the Left,Right,Hanging,Firstline parameters have values > 0 and are used with the Bullet parameter, they will overwrite the information in the bulleted list template or the document bullet point, where it is also included! Example enumeration is aligned 1cm from the left margin: {FPI:Left:1.0,Right:2.5,FirstLine:0.0,Bullet:oo}
- Example (no bullet): {FPI:Left:1.0,Right:2.5,FirstLine:0.0}
- Position of the placeholder: Anywhere within the relevant paragraph.

```
HeaderBreak :==
     "{HeaderBreak:"(<TableName>|<Attribute>){","(<TableName>|<Attribute>)}"}"
```

- Heading over a range of identical values in a list. After a value change of the specified attribute, the attribute list, or the end of the specified input table (the attribute name is optional), the row is copied to the end of the list and then the attribute values of the copied row are replaced. Must be in a Word, Excel list in the first column.
- Example: {HeaderBreak:COI. NameProductGroup}
- Placeholder position: In the first column of a list. The row contains the data that should be displayed during the wrap.
- Please note: when using charts (Bar Chart, Pie Chart), adjusting the data ranges of the charts does not work because HeaderBreaks interrupt them. A possible workaround is to insert the data used in the chart on a separate worksheet and relate the data section of the charts to it.

```
IncludeHeaderFooter :== "{IncludeHeaderFooter}
```

- Headers and footers of the document are usually not searched for placeholders for performance reasons. If these are also to be searched, this placeholder must be inserted.
- Example: {IncludeHeaderFooter}
- Placeholder position: Anywhere in the main text of the document, not within the document's headers or footers

`Importfile :== "{Importfile:"<AbsolutPathFileAndExtension>"}"`

- This tag has NO effect if the name of the import file is passed when the generation is called using parameters!
- In AbsolutPathFileAndExtension, the absolute path, plus filename, plus fileextension of the import file is specified. The name must be specified according to the Windows specification, even for Apple operating systems! E.g. "C:\Temp\RechnungDaten.txt".
- Example: {Importfile:C:\Temp\RechnungDaten.txt}
- Position of the placeholder: Anywhere in the document (is optional for Windows systems, opens Fileopen dialog without specifying it)

`NewParagraph :== "{NewParagraph}"`

- Divides the Word paragraph into two format-identical paragraphs at the defined point. Cannot be used in Excel templates.
- Typically used in conjunction with the Zero or NonZero tag. However, it may also be included in the imported data. This tag does not work within lists, see {NewLine}.
- Example: {NewParagraph}
- Example: {CI.ListPrice|Format:"List price: "##,##0.00"{NewParagraph}"|DependOn:CI.Discount}
- Placeholder position: At the point where the paragraph should be split.

`NewLine :== "{NewLine}"`

- Inserts a line break in a paragraph at the defined location. Cannot be used in Excel templates.
- Typically used in conjunction with the Zero or NonZero tag. However, it may also be included in the imported data.
- Example: {NewLine}
- Example: {CI.ListPrice|Format:"List price: "##,##0.00"{NewLine}"|DependOn:CI.Discount}
- Placeholder position: At the point where you want to insert the line break.

`OptionalDocumentPart :==`
`"{ODP:"(<Attribute>|<String>|<Number>)<ComparisonOperator>(<Attribute>|<String>|<Number>)"}"`

- Marks a table as optional. If the condition is not met, the entire table is removed from the document. This allows specific parts of the document to be shown or hidden.
- Example: {ODP:P.NamePartner="Posity AG"}
- Placeholder position: In any cell of the table.

`RemoveRow :== "{RemoveRow}"`

- In certain situations, a row of a list should not be integrated into the document, e.g. because there is no value in an attribute (for example, if the discount of row is 0). This tag is typically used in combination with the Zero, NonZero, True, or False tag of the AttributeValue placeholder.
- Example: {CO. CashDiscount|Zero:{RemoveRow}}

`Row :== "{Row}"`

- If the row keyword is set in a list, a single row is generated for each processed record in this list. Creating a new row is performed no more than once for each record (if multiple attributes are listed).
- Example: {row}
- Placeholder position: In the first column of a list. The row contains the data that you want to display for each record.

`Spacer :== "{Spacer}"`

- Within a single list, several rows can be created, which can be edited by Posity Neo (e.g. of type Row, FooterBreak, HeaderBreak). In order for the frames of these lines to be formatted independently of each other, it must be possible to insert a blank line between them. However, this blank line should no longer appear in the final document. This is achieved by marking this blank line in the first column with spacer, these lines are deleted after the document is created.
- Example: {Spacer}
- Position of the placeholder: In the first column of the row to be deleted.

```
Sum :==
     "{sum:{"<attribute>"}"
     ["|Format:"["""<StringWithBreak>"""]<FormatPattern>["""<StringWithBreak>"""]]
     ["|SumCondition:"<Condition>]
     ["|ShowConditionAll:"<Condition>;"<Boolean>"]
     ["|ShowConditionAny:"<Condition>;"<Boolean>"]
     "}"
```

- The Sum command must be used in a list in combination with the FooterBreak or HeaderBreak. In the Sum placeholder, the values are summed up to the FooterBreak or HeaderBreak.
- **Format**: In the <FormatPattern> the output format can be set according to the specification of the C# format command, supplemented by a preceding and following string (e.g. '"Total: "#,##0.00' or '#,##0" \%"'). See also description of <FormatPattern>.
  Cannot be used in Excel templates.
- **SumCondition**: If the condition is met, the value of the attribute is taken into account in the sum. In comparison, only attributes of the current data set can be used. Both attributes must be of type String (all types except Numeric are used as strings) or of type Numeric. If the comparison is not possible (wrong types, wrong attribute), the condition is ignored. Capitalization is ignored in the evaluation of the comparison operation! Attributes must be enclosed in curly brackets.
- **ShowConditionAll**: The sum is displayed only if the defined condition for each cleared record is met for the total formed. Conditions of records that are not used for summation due to the SumCondition flag are not honored. ShowConditionAll can be combined with ShowConditionAny. The Bool value after the ShowCondition can be used to control the display of the sum if the ShowCondition does not apply. True removes the line where the sum tag is located. With "false" the sum is not displayed. The line is removed if ShowConditionAll or ShowConditionAny are true .
- **ShowConditionAny**: The sum is displayed only if the defined condition for any cleared record is met for the total formed. Conditions of records that are not used for summation due to the SumCondition flag are not honored. ShowConditionAny can be combined with ShowConditionAll. The Bool value after the ShowCondition can be used to control the display of the sum if the ShowCondition does not apply. True removes the line where the sum tag is located. If false, the total is not displayed. The line is removed if ShowConditionAll or ShowConditionAny are true .
- Example: {sum:{COI. Cost}|Format:#,##0.00}
- Example: {sum:{invoice.amount}|SumCondition:{Invoice.Amount}>0.0}
- Example: {sum:{invoiceline.amount}|ShowConditionAll:{InvoicePosition.Option}="True"}
- Example: {sum:{invoiceline.amount}|ShowConditionAny:{InvoicePosition.Option}="True";" true"}
- Position of the placeholder: At the point where the sum should be displayed (in the list).

```
TabChar :== "{TabChar}"
```

- Inserts a tab character in the defined position of a paragraph. Cannot be used in Excel templates.
- Typically used in conjunction with the Zero or NonZero tag. However, it may also be included in the imported data.
- Example: {TabChar}
- Example: {CI.ListPrice|Format:"Listprice{TabChar}"##,##0.00|DependOn:CI.Discount}
- Placeholder position: Where you want to insert the tab character.

```
TableAlias :== "{TableAlias:"<TableAliasName>=<TableName>"}"
```

- To improve readability and display, aliases can be assigned to each input table name. Attention, if an alias name is assigned, it must be used consistently. For example, if the alias CI is set for the input table CustomerInvoice (see example below), then the placeholder can be written with '{CI.InvoiceDate}' instead of '{CustomerInvoice.InvoiceDate}'.
- Example: {TableAlias:CI=CustomerInvoice}
- Placeholder position: Anywhere in the document (optional)

```
ThousandSeparator :== "{ThousandSeparator:"<AnyString>"}"
```

- Specifies which thousands separator should be used in the FormatPattern (in the format, the thousands separator is always given as a comma). This setting is used regardless of the operating system and any country settings.
- Example: {ThousandSeparator:'}
- Placeholder position: Anywhere in the document (optional)
- Cannot be used in Excel templates.

```
UpdateFields :== "{UpdateFields}"
```

- If the field functions in the document or in charts (image and data area) are to be updated automatically, this placeholder must be set – otherwise they must be updated manually after opening the document. The automatic update only works if the document is actually opened in Word or Excel through the processing process.
- Note: Only field functions in the main text (not in the header or footer) will be updated.
- Example: {UpdateFields}
- Placeholder position: Anywhere in the document (optional)

```
ZebraLines :==
"{Zebra:"<ColorR>","<ColorG>","<ColorB>";"<ColorR>","<ColorG>","<ColorB>"}"
```

- In the lists, the rows can be stored with alternating colors. The first color is used for the first line, then the second color is used to color each line alternately. If other lines are embedded between the lines (e.g. header or footer), or if you switch to a new list, you start again with the first color.
- Example: {Zebra:190,190,190; 220,220,220}
- Position of the placeholder: In the row which should be colored zebra-like (in any column of the row).

4. Example Word Template and Generated Document

On the following pages, an example of a template and a document created with it is shown. In the template, the borders of the tables have been colored green to make the template easier to interpret. This coloring was not used in the generation.

The input contains two tables: CO (CustomerOffer) and COI (CustomerOfferItem).

{CO. FIRSTNAMEPARTNERCUSTOMER}{CO. FirstNamePartnerCustomer|Zero:|NonZero: }{CO. NAMEPARTNERCUSTOMER}
{CO. MAILADDRESSLINE1PARTNERCUSTOMER} {CO. MAILADDRESSLINE1NUMBERPARTNERCUSTOMER}
{CO. POSTALCODECITYPARTNER} {CO. NAMEPLACEPARTNER}

OFFER No. {CO. OfferNumber}
{CO. NAMEOFFER} Winterthur, {CO. OfferDate|Format:dd.MM.yyyy}

{CO. ImageConstruction|MaxWidth:600|MaxHeight:300}

Plant location:	{CO. NamePlantLocation}
Installation:	{CO. NameSetupType}
Slope:	{CO. Inclination|Format:#0}
Orientation:	{CO. Orientation|Format:#0}
Module surface	{CO. ModuleArea|Format:#,##0} m2
Installed capacity:	{CO. InstalledOutput|Format:#,##0.00} kWp
Specific Yield:	{CO. SpecificOutput|Format:#,##0} kWh/kWp
Estimated annual yield:	{CO. EstimatedAnualOutput|Format:#,##0} kWh
Commissioning until:	{CO. CommissioningDate|Format:dd.MM.yyyy}
Project costs incl. VAT:	{CO. ProjectCostsWithTax_Currency} {CO. ProjectCostsWithTax|Format:#,##0.00}; *{CO. CostsPerkWp_Currency} {CO. CostsPerkWp|Format:#,##0}/kWp*

Quantity	Description	Unit price	CHF/kWp	Price
{Spacer}				
{Row}{COI. Quantity|Format:#,##0|Zero:}{COI. ShortNameMassUnit}	{COI. NameProduct}{COI. DescriptionProduct|Zero:|Nonzero:}{COI. DescriptionProduct}	{COI. RetailPricePerProduct|Format:#,##0.00|Zero:}	{COI. CostPerOutput|Format:#,##0|Zero:}	{COI. Cost|Format:#,##0.00}{COI. Option|True:(Option)|false:}{COI. ByClient|True:(on-site)|false:}
{Spacer}				
{FooterBreak:COI. NameProductGroup}**{COI. NameProductGroup|OldValue}**				**{Sum:COI. Cost|Format:#,##0.00}**
{Spacer}				
{Spacer}				

{CO. ListPrice|Format:"List price "#,##0.00
|DependOn:CO. Discount}{CO. Discount|Format:"Discount: "##,##0.00"% "|Zero:}{CO.
DiscountAmount|Format:#,##0.00
|DependOn:CO. Discount}Net {CO. ProjectCosts|Format:#,##0.00}
8% VAT {CO. VAT|Format:#,##0.00}
Price for work {CO. ProjectCostsWithTax_Currency} {CO. ProjectCostsWithTax|Format:#,##0.00}

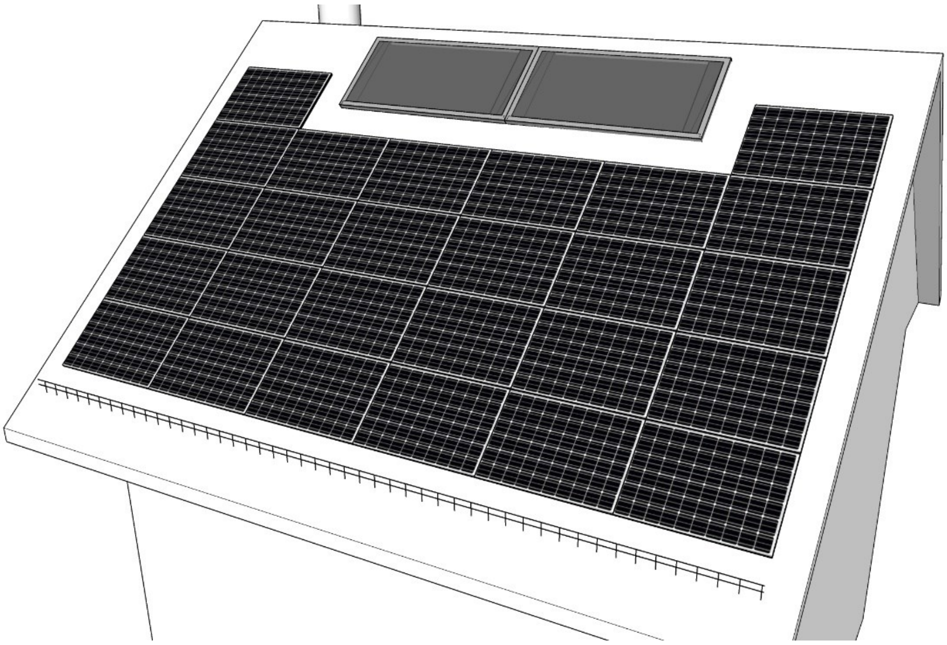
Offer validity:	{CO. ValidityOfOffer} days
Payment Terms:	{CO. TermsOfPayment}{CO. CashDiscount|Format:" Discount: "##,##0.00"% "|Zero:} {CO. PaymentPeriodCashDiscount|Format:"("##,##0" days)"|Zero:}
Term of payment:	{CO. PaymentPeriod|Format:#,##0} days
Guarantees:	{CO. Warranty}
Confirmation of order:	{CO. TextFeederOffer}
Place and date	Signature, the client

{TableAlias:CO=CustomerOffer} {TableAlias:COI=CustomerOfferItem} {ThousandSeparator:'} {DecimalSeparator:.}

COMPANY NAME
STREET
8400 WINTERTHUR

OFFER
ROOFTOP SYSTEM FAMILY XYZ

No. A150000
Winterthur, 27.11.2015



Plant location:	Sloping roof
Installation:	Grown
Slope:	40°
Orientation:	South
Module surface	42 m2
Installed capacity:	8.58 kWp
Specific Yield:	1,040 kWh/kWp
Estimated annual yield:	8,923 kWh
Commissioning until:	30.03.2016
Project costs incl. VAT:	CHF 28'284.30; *CHF 3,297/kWp*

Quantity	Description	Unit price	CHF/kWp	Price
26 pcs.	PV module BenQ SunForte 330 Wp High-performance quality module Module Effectiveness Gard 20.3% (LxWxH): 1559 x 1046 x 46 mm Weight: 18.6 kg Pmax 330 W (-0/+3 power tolerance) Umpp 54.7 V, Impp 6.04 A IEC 61215, IEC 61730	404.35	1'225	10'513.15
26 pcs.	Mounting system with insertion profiles, black Cross-laminated insertion system from Creotecc black anodised with roof hook for pan tiles	84.50	256	2'197.00
1 pc.	Inverter Fronius Symo 8.2-3-M AC output power 8,200 VA 3 phase max. 13.5A Efficiency 97.8% Protection class IP 55 2 MPP trackers 2+2 DC connectors / weight 21.9 kg	2'034.50	237	2'034.50
1 pc.	Fronius Smart Meter Measurement and visualization of self-consumption bidirectional counter 63A/3ph Ethernet port on site	392.08		0.00 (Option)
1 pc.	DC Ü-Protection PV DC 2INx2 2MPPT 2SPD WM4 1000 V GAK for inverters with 2 MPPT Inputs 2+2, Outputs 1+1, Connection via WM4 plug, Type II surge protection, HxWxD: 360 x 360 x 171 mm	333.59	39	333.59
120 m	Solar cable 6 mm² halogen-free, abrasion-resistant, various colours +/-	0.94	13	112.32
10 m	Aluminium tube M40 for riser zone, incl. pipe clamps and elbows	7.80	9	78.00
15 m	Equipotential bonding 16 mm²	1.22	2	18.23
8 pcs.	Solar plug MC4 1 pair, plug and socket	3.12	3	24.96
Material				**15'311.75**

35 h	Assembly of substructure and modules	85.01	347	2'975.18
25 h	DC Electrical Assembly	92.00	268	2'300.00
1 pc.	AC Electrical Assembly from inverters to house connection boxes Fuses, Surge Protection, Disconnect Switches Incl. installation display, safety certificate	1'500.00	175	1'500.00

Quantity	Description	Unit price	CHF/kWp	Price
Work / Assembly				**6'775.18**
3 p.m.	Planning & Documentation - Registration form for solar systems - Registration for one-off remuneration or other subsidies - EEA registration with local EW - Audit support and WR registration (incl. fees)	104.00	182	1'560.00
4 h	Transports Material delivery and disposal	120.00	56	480.00
Service / Planning				**2'040.00**
1 pc.	Safety scaffolding SUVA compliant	1'900.00	221	1'900.00
Partners				**1'900.00**
4 pcs.	ABS Loop for installation and subsequent maintenance access	40.56	19	162.24
Safety				**162.24**
Net				26'189.16
8% VAT				2'095.13
Price for work				**CHF 28'284.30**

Offer validity: 60 days

Payment Terms: 70% upon receipt of all permits
30% after commissioning

Term of payment: 14 days

Guarantees: Work Solarville 2 years
Products according to the manufacturer's data sheets
(Solarville warranty 2 years)

Confirmation of order: I agree with the general terms and conditions, which can be viewed under www.solarville.ch/agb. I confirm that I have fully understood the offer with enclosures and instruct Solarville AG to carry out its above-mentioned services.

Place and date Signature, the client

5. Example Excel template and generated document

On the following pages, an example of a template and a document created with it is shown.

The input contains two tables: CO (CustomerOffer) and COI (CustomerOfferItem).

{Importfile:C:\temp\OfferteDaten.txt} {TableAlias:CO=CustomerOffer} {TableAlias:COI=CustomerOfferItem}
{Exportfile:C:\temp\output\solarville_offerte.xlsx|Open:Yes}

Customer:
{CO. FirstNamePartnerCustomer} {CO. NamePartnerCustomer}
{CO. MailAddressLine1PartnerCustomer} {CO. MailAddressLine1NumberPartnerCustomer}
{CO. PostalCodeCityPartner} {CO. NamePlacePartner}

Quotation data:
No. {CO. OfferNumber}
Offer: {CO. NameOffer}
Date: {CO. OfferDate|Format:dd.MM.yyyy}

Location:	{CO. NamePlantLocation}
Installation:	{CO. NameSetupType}
Slope:	{CO. Inclination|Format:#0}
Orientation:	{CO. Orientation|Format:#0}
Module area:	{CO. ModuleArea|Format:#,##0} m2
Installed capacity:	{CO. InstalledOutput|Format:#,##0.00} kWp
Specific Yield:	{CO. SpecificOutput|Format:#,##0} kWh/kWp
Estimated annual yield:	{CO. EstimatedAnualOutput|Format:#,##0} kWh
Commissioning until:	{CO. CommissioningDate|Format:dd.MM.yyyy}
Project costs incl. VAT:	{CO. ProjectCostsWithTax_Currency} {CO. ProjectCostsWithTax|Format:#,##0.00}; {CO. CostsPerkWp_Currency} {CO. CostsPerkWp|Format:#,##0}/kWp

{CO. ImageConstruction|MaxWidth:600|MaxHeight:300|FitRow:Yes}

Quotation Positions:				
Quantity	**Description**	**Unit price**	**CHF/kWp**	**Price**

<table>
<tr><td>{Row} {COI. Quantity|Format:#,##0|Zero:} {COI. ShortNameMassUnit}{Zebra:190, 190,190; 220,220,220}</td><td>{COI. NameProduct}{COI. DescriptionProduct}{COI. NameProductGroup}</td><td>{COI. RetailPricePerProduct|Zero:}</td><td>{COI. CostPerOutput|Zero:}</td><td>{COI. Option|True: {COI. Cost|Format:#,##0.00} (option)|False: {COI. Cost}} {COI. ByClient|True:(on-site)|false:}</td></tr>
<tr><td>{Spacer}</td><td></td><td></td><td></td><td></td></tr>
<tr><td>**{FooterBreak:COI. NameProductGroup}{COI. NameProductGroup|OldValue}{ BOC:5; 0; 0}{BGC:0|2; 1; 201,255,233}**</td><td>**{BGC:0|2; 1; 201,255,233}**</td><td>**{BGC:0|2; 1; 201,255,233}**</td><td>**{BGC:0|2; 1; 201,255,233}**</td><td>**{Sum:COI. Cost}{BGC:0|2; 1; 201,255,233}**</td></tr>
<tr><td>Net:</td><td></td><td></td><td></td><td>{CO. ProjectCosts}</td></tr>
<tr><td>8% VAT:</td><td></td><td></td><td></td><td>{CO. VAT}</td></tr>
<tr><td>**Price:**</td><td></td><td></td><td></td><td>**{CO. ProjectCostsWithTax_Currency} {CO. ProjectCostsWithTax|Format:#,##0.00}{BOC:5; 0; 0}**</td></tr>
</table>

Offer validity: {CO. ValidityOfOffer} days

Payment Terms:	{CO. TermsOfPayment} {CO. CashDiscount|Format:"Discount: "##,##0.00"% "|Zero:} {CO. PaymentPeriodCashDiscount|Format: "("##,##0" days)"|Zero:}
Term of payment:	{CO. PaymentPeriod|Format:#,##0} days
Guarantees:	{CO. Warranty}
Confirmation of order:	{CO. TextFeederOffer}

Place and date	**Signature, the client**

{UpdateFields}
{IncludeHeaderFooter}

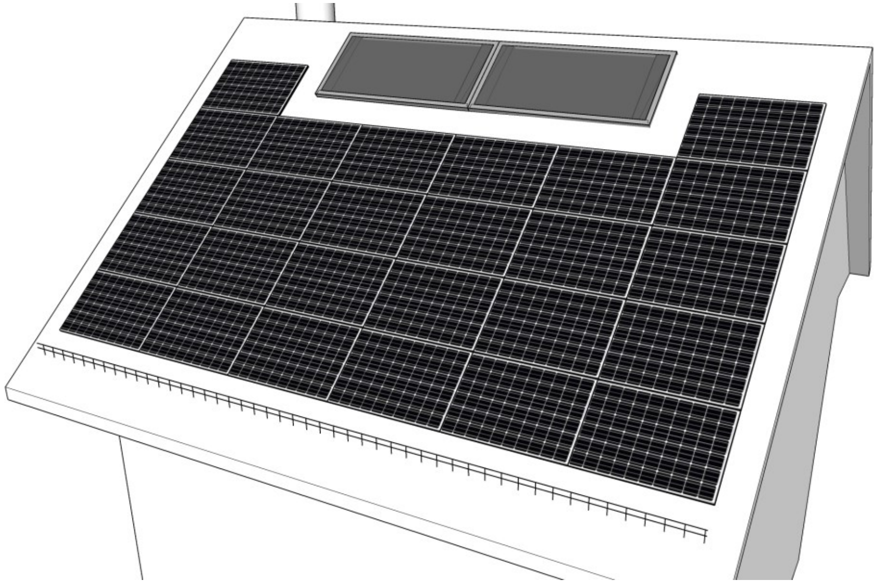
Customer:

SURNAME
STREET
8400 WINTERTHUR

Quotation data:

No. A150000
Offer: Rooftop system for the Nagel family
Date: 27.11.2015

Location:	Sloping roof
Installation:	Grown
Slope:	40°
Orientation:	South
Module area:	42 m2
Installed capacity:	8.58 kWp
Specific Yield:	1,040 kWh/kWp
Estimated annual yield:	8,923 kWh
Commissioning until:	30.03.2016
Project costs incl. VAT:	CHF 28'284.30; CHF 3,297/kWp



Quotation Positions:				
Quantity	Description	Unit price	CHF/kWp	Price
26 pcs.	PV Module BenQ SunForte 330 Wp High Performance Quality ModuleModule Effectiveness 20.3%(LxWxH): 1559 x 1046 x 46 mmWeight: 18.6 kgPmax 330 W (-0/+3 Power Tolerance)Umpp 54.7 V, Impp 6.04 AIEC 61215, IEC 61730 Material	404.35	1'225.31	10'513.15

26 pcs.	Mounting system with inlay profiles, blackInsertion system in cross composite from Creotecc black anodized with roof hook for pan tilesMaterial	84.50	256.06	2'197.00
1 pc.	Inverter Fronius Symo 8.2-3 MAC output power 8,200 VA3 phase max. 13.5AWdegree of variation 97.8%Protection class IP 55 2 MPP tracker 2+2 DC connections / weight 21.9 kgMaterial	2'034.50	237.12	2'034.50
1 pc.	Fronius Smart MeterMeasurement and visualisation of self-consumptionBidirectional meter 63A/3phEthernet connection on-siteMaterial	392.08		0.00 (option)

1 pc.	DC Ü-Protection PV DC 2INx2 2MPPT 2SPD WM4 1000 VGAK for inverters with 2 MPPTinputs 2+2, outputs 1+1, connection via plug WM4, surge protection type II, HxWxD: 360 x 360 x 171 mmMaterial	333.59	38.88	333.59
120 m	Solar cable 6 mm^{2}halogen-free, abrasion-resistant, various colours +/-material	0.94	13.09	112.32
10 m	Aluminium tube M40for climbing zone, incl. pipe clamps and elbowMaterial	7.80	9.09	78.00
15 m	Equipotential bonding 16 mm^{2}Material	1.22	2.12	18.23
8 pcs.	Solar Plug MC41 Pair, Plug and SocketMaterial	3.12	2.91	24.96
Material				**15311.75**
35 h	Assembly of substructure and modulesWork / Assembly	85.01	346.76	2'975.18
25 h	DC Electrical AssemblyWork / Assembly	92.00	268.07	2'300.00

1 pc.	AC electrical installationfrom inverter to house junction boxfuses, surge protection, circuit breakersIncl. Installation Indicator, Safety CaseWork / Assembly	1'500.00	174.83	1'500.00
Work / Assembly				**6775.18**
3 p.m.	Planning & Documentation- Registration form for solar installations- Registration of one-off remuneration or other subsidies- EEA registration with local EW - Audit support and WR registration (incl. fees)Service / Planning	104.00	181.82	1'560.00
4 h	Transports, Material Delivery and DisposalService / Planning	120.00	55.94	480.00
Service / Planning				**2040.00**
1 pc.	Safety scaffoldingSUVA compliantPartner companies	1'900.00	221.45	1'900.00
Partners				**1900.00**

4 pcs.	ABS loopfor assembly and subsequent maintenance accessSafety	40.56	18.91	162.24
Safety				**162.24**
Net:				26'189.16
8% VAT:				2'095.13
Price:				**CHF 28'284.30**

Offer validity:	60 days
Payment Terms:	70% upon receipt of all permits30% after commissioning
Term of payment:	14 days
Guarantees:	Work Solarville 2 yearsProducts according to the manufacturer's data sheets (Solarville warranty 2 years)

Confirmation of order:

I agree with the general terms and conditions, which can be viewed under www.solarville.ch/agb. I confirm that I have fully understood the offer with enclosures and instruct Solarville AG to carry out its above-mentioned services.

Place and date

Signature, the client