

Templates

Templates are text or files that Tempus Serva will output content to, when displaying a userinterface or exporting a record.

- [Export templates](#)
- [Form templates](#)
- [Interface templates](#)
- [Word documents from templates](#)

Export templates

Export templates are **binary files** uploaded to the template, containing special tags where the content from Tempus Serva fields should be inserted. Transformation of tables, files etc. is done automatically.

An example for a Word report is found here: [Word documents from templates](#)

DOCX / Word limitations

Content can inserted most places in the document, except for the following

- Page header and footers
- Document metadata

Due to Microsoft Word everchanging document format, it is not possible to preserve the all styling of text inserted into drawing building blocks (textareas,circles etc.).

- Normal formatting will be preserved
- Special properties like angle will be lost

PDF templates

PDF templates a basically Word templates transformed to PDF afterwards. To build PDF's just upload a normal docx template, and change the type to PDF after upload.

To enable docx to pdf conversion the configuration 'servicePdfConverter' has to be set up. To do this, first a token has to be generated and inserted into the database on the convert server. This token has to be added to the value of 'servicePdfConverter', after the url. The value of 'servicePdfConverter' should then look like:

<https://convert.tempusserva.dk/Word2pdf/Word2pdf?token=XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX>

Form templates

Structure and syntax

The structure of the tags in the HTML template are

- {header}
- {tablename}
- {pageselector}
- {default_start}
 - {default_name}
 - {default_value}
 - {default_help}
- {default_end}
- {defaultField}
- {footer}

You may also add additional tags for special fields:

- "{ " + [field systemname] + "_name}"
- "{ " + [field systemname] + "_value}"
- "{ " + [field systemname] + "_help}"

Template handling

Templates are injected with values from the current record like this

1. For each field
 1. Try to inject [system fieldname] + [prefix]
 2. If failed
 - Copy content from {default_start} to {default_end}
 - Inject values into content copy
 - Inject content copy into {defaultField}
2. Inject the form page selector into {pageselector}
 1. If failed: Ignore

Missing values for default_XXX tags will automatically get appended to the template.

Relation til styling

type	class	id	content	template
tbody	PAGE_[field PageID]		Container for a single field	{default_start} / {default_end}
div	FieldLabel	NB_[system fieldname]	Label for field	{default_name}
div	FieldValue	VB_[system fieldname]	Value of field / Input for field	{default_value}
div	FieldNotes	HB_[system fieldname]	Optional help text for the field	{default_help}

Example

```
{header}

<h3>{tablename}</h3>

<table class='tableForm'>
  <tr><td style="padding: 0px; border: 0px;">{pageselector}</td></tr>
  {default_start}
  <tr>
    <td class="tableFieldTableLabel"><p>{default_name}</p></td>
  </tr>
  <tr>
    <td class="tableFieldTableValue">
      <p>{default_value}</p>
      <p>{default_help}</p>
    </td>
  </tr>
  {default_end}
  {defaultField}
</table>

{footer}
```

Interface templates

Interface templates is cleartext **text content in HTML** format, that is edited directly the backend.

Two types of tags exist

- General repeatable placeholders
- Specific placeholders referencing a single field

Specific placeholders have priority, over generic placeholders.

Its generally recommended to use the generic placeholders, because they automatically adapt when fields are added or deactivated in the solution. Also note that specific placeholders ignore the field sort order.

Word documents from templates

Basic template

Checklist

1. [Solution] > "Advanced" > Template: "Add"
 - Copy template tags from popup: "Template field tokens" > "Office"
2. Prepare file on local PC
 - Copy placeholder tags from backend
 - Save in safe location so you dont need to download when updating later
3. [Solution] > "Advanced" > Template: "Add"
 - Set content type to "MS Word" or "PDF"
 - Upload file
4. If solution has more than 1 export template
 - Check that solution has [Solution] > "Advanced" > "Template selector" enabled

Tutorial wordtemplate.PNG

Trouble shooting

Content from TempusServa is not displayed inside tags

- Disable spelling/grammar
- Take care not to edit the text inside tags
- Select the whole tag before applying styling
- Paste tags from notepad or other program without text styling
- Try the "Fix word tags"-codeunit ([insert link](#))

Subtemplates

Subtemplates can be used for special fields that support the notion.

- Complex question

The whole sub template is injected into the tag, as if it had been normal data.

Complex questions

Supported tag names

```

${FIELDLABEL}
${FIELDHELP}
${CHOICEIMAGE}
${CHOICELABEL}
${CHOICETEXT}
${ANSWERLABEL}
${ANSWERTEXT}
${PICTURE}
${COMMENT}
```

Complex questions

Supported tag names in first template

```

${FIELDLABEL}
${FIELDHELP}
```

Supported tag names in second template

```

${LOCATION}
${CHOICEIMAGE}
${CHOICELABEL}
${CHOICETEXT}
${ANSWERLABEL}
${ANSWERTEXT}
```

Special considerations

Named styling is not supported, but can work with a workaround.

1. Copy styled content from subtemplate

2. Paste into main template and save file
3. Remove content and save again

Sections

Sections that are shown based on boolean test

Trouble shooting

Table page flow

Word does not respect table setting for page flow, so table rows break in undesired ways

1. Create a new empty table with one cell (without borders)
2. Right-click cell and disable "Allow row to break across pages" / "Tillad opdeling af række"
3. Paste the old table into the new table

RowControl.PNG