BEVER

Bever Systems
bevercrm.com

BEVER ADVANCED D365 DOCUMENT GENERATION

Quick Guide

The essentials of building Word templates and making them available in Dynamics 365, on a few pages.

Dynamics 365 Online | Dataverse | Microsoft Word



AUDIENCE

For template authors and administrators

EDITION

November 2026

INSIDE THIS DOCUMENT

Contents

1 How it works 3
- The records you work with 3
- Where templates show up 3

2 Your first template in six steps 4
- The template record 4

3 Placeholders and field paths 5
- One related record 5
- Optional attributes with ?. 5
- Writing the syntax as text 5

4 Variables, defaults and conditions 6
- Runtime variables 6
- Defaults and joining text 6
- Inline conditions with If(...) 6

5 Formatting and language 7
- Language 7
- Dates and time zones 8

6 Repeating and conditional regions 8
- Repeating content with {{#each}} 8
- Conditional content with {{#if}} 9

7 Named queries 9

8 Availability rules 10

9 Generating documents 11
- One record or many 11
- Scope: whose access a document reflects 12
- The subscription 12

10 Troubleshooting 12

Cheat sheet 14

The essentials of building a Word template, deciding which records may use it, and generating documents from Dynamics 365. Every topic is covered in full in the User Guide; this guide is the short road through it.

> **Who this is for.** Template authors working in Word, and administrators configuring templates in the Dynamics 365 app. Nothing in this guide needs code: adding a template, adding a column to a document or changing a date format is configuration, never a release.

1 How it works

A user opens a record, or selects several in a grid, clicks **Generate Document** in the ribbon and picks a template. The rest happens on the server:

1. **Pick** — Open a record, or select up to 250 in a grid, and click *Generate Document*.
2. **Check** — The server lists only the templates this record is eligible for.
3. **Render** — The chosen `.docx` is filled with live Dataverse data.
4. **Download** — The browser saves the finished Word document.

Everything a document can express is written **inside the Word file** with `{{…}}` placeholders: fields, nested lookups, tables of related records, conditions and formatting.

The records you work with

Record	Holds	Edited by
Document Generation Template	The template's settings. The `.docx` is attached to it as a note.	Administrator, in the app
The attached `.docx`	The layout and the placeholders	Template author, in Word
Document Generation Availability Rule	Which records of the table may use the template. Optional.	Administrator
Document Generation Query	A named FetchXML query a template can list rows from. Optional.	Administrator
Document Generation Template Configuration	The environment's subscription license. One per environment.	Administrator, on the configuration page

Where templates show up

Surface	Shows
The **Generate Document** ribbon menu	Eligible templates marked *Quick Template* = Yes
The **View All Available Templates** dialog	**Every** template eligible for the record, with preview and download

Installing the managed solution adds the **Generate Document** button to the form and home grid of every table. There is nothing to configure for it.

2 Your first template in six steps

1. **Create the template record.** In the app, create a *Document Generation Template*. Set *Name*, *Type* = Word, *Associated Entity Logical Name* = the table it serves (chosen from the list), *Language* and *Scope*.
2. **Write the `.docx` in Word.** Type placeholders such as `{{name}}` where the values belong. Sections 3 to 6 of this guide are the syntax; the cheat sheet on the last page is the one-page version.
3. **Attach the file** to the template record as a note. The newest attached file is always the template.
4. **Decide whether it belongs in the ribbon menu.** *Quick Template* = Yes lists it there; No keeps it in *View All Available Templates* only.
5. **Narrow it, if needed.** With no availability rules the template is offered on every record of its table. Add rules (§8) only to narrow that.
6. **Create the named queries it uses**, one active *Document Generation Query* per `{{#each query:Name}}` (§7). Then test on one record, and on a selection of several records in a grid.

The template record

Column	Required	What to put in it
Name	✔	Shown on the ribbon button. The generated file is named after it.
Type	✔	**Word**, the only type supported today.
Associated Entity Logical Name	✔	The table the template serves, such as `account` or `incident`. A template is never offered on any other table.
Language		The language the template is written in. Choice labels, dates and numbers follow it (§5).
Use Current User Selected Language		**Yes** = render in the language of whoever generates the document instead.
Scope		Whose access the data is read with: **System** (the default) or **User** (§9).
Description		The ribbon button's tooltip, and the text in the dialog.
Quick Template		**Yes** = listed directly in the ribbon menu.
Status	✔	Must be **Active**. Deactivate a template to take it out of service while you revise it.

✓ **Correcting a template is an upload, not a delete.** Attach the fixed `.docx` as a new note and it takes effect immediately; the previous file stays on the record as history. A template with no attached file cannot render.

3 Placeholders and field paths

A placeholder is `{{ … }}`, written anywhere text can go: body paragraphs, table cells, text boxes, headers, footers, footnotes and endnotes. The value takes the placeholder's own formatting, so a bold placeholder prints a bold value.

```
{{name}}                          a field of the record
{{ownerid}}                       a lookup prints the related record's name
{{ownerid.fullname}}              one hop through a lookup
{{ownerid.businessunitid.name}}   two hops; there is no practical limit
{{$root.name}}                    the record the document is for, from anywhere
```

- **An empty lookup anywhere along a path prints nothing.** The whole path short-circuits.
- **A mistake in a path is an error on every record**, whether or not that record has a value there: a misspelt attribute, or a text column followed as if it were a lookup (`{{name.x}}`). A template that works on a test record therefore cannot fail later on another.
- Word may split what you type into several runs, and you may press **Enter** in the middle of a long placeholder to keep it readable. Both are fine.

One related record

Write	Gives
`{{[contact^parentcustomerid].fullname}}`	The oldest contact whose `parentcustomerid` points at this record (1:N)
`{{[new_tag^^new_case_tag].name}}`	The oldest record related through the N:N relationship `new_case_tag`
`{{First(relationship:new_case_notes).subject}}`	The oldest record of a relationship
`{{First(LastInvoice).new_total}}`	The first row of the named query `LastInvoice`, in the query's own order

Oldest means by `createdon`, with the primary key as a tie-break: a guarantee, so the same data always renders the same document. Use these where business rules allow only one related record, or where a named query sorts to the one you mean (sort by date descending and `First(…)` is the *latest*). An empty source prints nothing.

Optional attributes with `?.`

An Owner, Customer or Regarding lookup points at records of different tables, and those rarely share every column. `?.` makes the next segment optional: empty, instead of an error, when that table does not define it.

```
{{ownerid?.internalemailaddress ?? ownerid?.emailaddress}}     a user's or a team's e-mail
{{customerid?.fullname ?? customerid?.name}}                   a contact's or an account's name
```

Keep the strict `.` wherever a missing column would be a mistake.

Writing the syntax as text

Put a backslash before the braces: `\{{name}}` prints `{{name}}`. One escape at the start is enough for a whole marker.

4 Variables, defaults and conditions

Runtime variables

Variable	Gives
`{{$user.fullname}}`	The user generating the document. Further hops are allowed.
`{{$now}}`	One timestamp for the whole run, in the user's time zone
`{{$today}}`	The user's calendar date
`{{$recordid}}`, `{{$entityname}}`	The current record's id and table: the usual query parameters
`{{$index}}`, `{{$count}}`	Position and row count inside a repeating region; `1` outside one
`{{$parent.name}}`, `{{$root.name}}`	The record enclosing the current region; the record the document is for

Defaults and joining text

```
{{firstname ?? "Unknown"}}
{{ownerid.fullname ?? parentaccountid.ownerid.fullname ?? "Unassigned"}}
{{firstname + " " + lastname}}
{{"Case " + ticketnumber + " (" + statuscode + ")"}}
```

`??` falls through on **null only**: a field with no value, or anything reached through an empty lookup. An empty string is a value, and wins. To treat blank text as missing too, use `If`: `{{If(firstname = null, "Unknown", firstname)}}`.

> **Default the fields, not the expression.** A concatenation is never null, so `{{firstname + " " + lastname ?? "Unknown"}}` prints a lone space. Write `{{(firstname ?? "Unknown") + " " + lastname}}`.

Inline conditions with `If(…)`

`If` takes any number of *condition, value* pairs, then a mandatory default:

```
{{If(statuscode = 1, "Open", "Closed")}}
{{If(new_days_open > 30, "Critical", new_days_open > 15, "High", new_days_open > 5, "Medium", "Low")}}
{{If(ownerid <> null AND new_count > 10, "Red", "Green")}}
{{If(NOT new_isclosed, "Open", "Closed")}}
```

The operators are `=` `<>` `!=` `<` `<=` `>` `>=`, `AND` `OR` `NOT` and parentheses. A choice compares by value or by label (`statuscode = 1`, `statuscode = "Open"`), a lookup by GUID or by name, and text without regard to case. Text goes in double quotes, and so does a date: `"2026-01-31T00:00:00Z"`.

Empty values guard themselves. An empty value equals nothing and differs from everything, so no manual null check is needed:

Condition	Field empty	Field = "Closed"	Field = "Open"
`new_status = "Closed"`	not satisfied	satisfied	not satisfied
`new_status <> "Closed"`	**satisfied**	not satisfied	satisfied

The orderings `<` `<=` `>` `>=` are never satisfied by an empty value. Test emptiness itself with `= null` and `<> null`.

5 Formatting and language

A format follows the field after a colon: `{{createdon:dd/MM/yyyy}}`. Put it in quotes when it contains a space, a comma or a bracket: `{{createdon:"MMMM dd, yyyy"}}`.

Dataverse type	Default	Named formats	Also accepts
Text	as stored	`upper`, `lower`, `trim`	—
Two options	the field's own labels	`label`, `text`, `checkbox`, `crosscheckbox`, `checkmark`	any `A/B` pair
Date and time	`dd.MM.yyyy HH:mm`; date-only `dd.MM.yyyy`	`date`, `time`, `datetime`	any .NET date format
Whole number	`N0`	`percent`	any .NET numeric format
Decimal, money	`N` + the column's precision	`percent`; money also `currency`, `code`	any .NET numeric format
Choice, status	the localized label	`value`, `label`	—
Multi-choice	labels joined by a comma	`value`	any separator text
Lookup	the record's name	`name`, `id`, `entitytype`	—

```
{{new_date:dd/MM/yyyy}}          → 15/03/2024
{{new_date:"MMMM dd, yyyy"}}     → March 15, 2024
{{new_date:HH:mm}}               → 09:30
{{revenue:N0}}                   → 1,235
{{revenue:currency}}             → 1,234.50 $
{{revenue:code}}                 → 1,234.50 USD
{{new_ratio:percent}}            → 12.50 %
{{statuscode}}                   → Active
{{statuscode:value}}             → 1
{{new_channels:" / "}}           → Email / Phone
{{ownerid:entitytype}}           → systemuser
{{approved:checkmark}}           → ✓  /  ✗
{{approved:Agreed/Declined}}     → Agreed
```

Language

A document renders in **one language**, decided by the template record:

Use Current User Selected Language	*Language*	The document renders in
Yes	*(ignored)*	the language of whoever generates it
No	set, e.g. Russian (1049)	that language, whoever generates it
No	empty	the language of whoever generates it

The language decides choice labels, Yes/No labels, and the month names and separators of dates and numbers. The template's own prose is not translated: write one template per language. To print **one** placeholder in a different language, name it with `lcid=` after the format:

```
{{new_type:label:lcid=1049}}            the choice's Russian label, whoever is reading
{{createdon:"d MMMM yyyy":lcid=1033}}   9 March 2026
{{revenue:N2:lcid=1049}}                1 234,50
```

Dates and time zones

A date column prints what the same user sees on its form. *User Local* columns, and `{{$now}}`, are shown in the time zone of whoever generates the document; *Time Zone Independent* and *Date Only* columns are printed as stored. Conditions compare dates the way they print, so `{{If(createdon >= $today, …)}}` means "since midnight for this user".

6 Repeating and conditional regions

Repeating content with `{{#each}}`

```
{{#each relationship:contact_customer_accounts}} … {{/each}}    related records, 1:N or N:N (preferred)
{{#each query:OpenCases}} … {{/each}}                           the rows of a named query (§7)
{{#each OpenCases}} … {{/each}}                                 the same, short form
```

Prefer `relationship:` : it needs only the relationship's schema name and has no query to maintain. Use a named query when you need filtering, sorting, aggregation or columns from further out.

Where the markers sit decides what repeats:

The two markers are in	What repeats
two cells of **one table row**	that row
cells of **two different rows**	those rows and every row between them
two paragraphs	those paragraphs and everything between them
one paragraph	that paragraph

Example: a table of an account's contacts. Build a table with a header row and **one** body row. Put the opening marker at the start of the first cell and the closing marker at the end of the last:

Name	Email
`{{#each relationship:contact_customer_accounts}}{{fullname}}`	`{{emailaddress1}}{{/each}}`

The body row is copied once per contact and each copy is filled from its own contact. With no contacts the row disappears and the header remains.

Inside a region `{{field}}` reads the **row**, `{{$parent.field}}` the record containing the region and `{{$root.field}}` the record the document is for; `{{$index}}` and `{{$count}}` number the rows. Regions nest to any depth:

```
{{#each relationship:new_account_orders}}
Order {{name}}
  {{#each relationship:new_order_orderlines}}
  - {{new_product}} for order {{$parent.name}} ({{$root.name}})
  {{/each}}
{{/each}}
```

A marker alone on its line leaves no blank line behind. A page or section break on a marker's own line stays exactly once; to repeat a page break per row, put it on a line of its own **between** the markers.

Conditional content with `{{#if}}`

```
{{#if new_amount > 1000000}}
Requires board approval by the Credit Committee.
{{#else}}
Standard approval applies.
{{/if}}
```

- `{{#else}}` is optional, and must start its own paragraph or table row.
- For a condition inside one line, use `If(…)`: `{{#if x}}yes{{#else}}no{{/if}}` in one paragraph is an error that tells you so.
- `{{#if}}` and `{{#each}}` regions nest inside each other freely.

7 Named queries

When a relationship is not enough, an administrator creates a **Document Generation Query** record, and templates refer to it by name. FetchXML is never written inside the Word file.

Column	Meaning
Name	What templates write: `{{#each query:OpenCases}}`, `{{First(OpenCases).…}}`
Fetch XML	The query, with `{{…}}` placeholders as parameters
Template	Optional. Empty = shared by every template. Set = that template only, and it wins over a shared query of the same name.
Status	Only active queries are used

```
<fetch>
  <entity name="incident">
    <attribute name="ticketnumber" />
    <attribute name="createdon" />
    <filter type="and">
      <condition attribute="customerid" operator="eq" value="{{$recordid}}" />
      <condition attribute="statecode" operator="eq" value="0" />
    </filter>
    <order attribute="createdon" descending="true" />
  </entity>
</fetch>
```

Any placeholder can be a parameter: `{{$recordid}}`, `{{$user:id}}`, a field, a lookup. Values are written in invariant form and XML-escaped, so a value can never change the structure of the query.

Three rules for queries:

1. **A row holds only what the query selects.** Reading a column the query does not select is an error naming the query; add `>` to it.
2. **Read linked tables through a `link-entity` with an `alias`**, not through a lookup: `{{User.fullname}}`, never `{{ownerid.fullname}}`. A `link-entity` that selects columns without an alias is refused.
3. **Sort when "first" matters.** `{{First(Q).…}}` is the first row in the query's own `<order>`; a query without one has no defined first row.

The query says	Write
`<order>`	`{{CaseStatus}}`
selecting `fullname`	`{{User.fullname}}`
a `link-entity` inside another, `alias="BusinessUnit"`	`{{BusinessUnit.name}}`

Aliases are case sensitive. A value keeps its column's formatting: an aliased choice prints its label, and money its record's currency.

How many rows. A query with no limit returns every matching row. `top="2"` and `count="2"` return the first two; `count="2" page="2"` returns rows 3 and 4. A `distinct` or aggregate query returns at most 5000 rows: more is an error, never a silently shorter table.

8 Availability rules

Which records may use a template is decided in two steps:

1. **Which table.** The template's *Associated Entity Logical Name*. A template is never offered on another table.
2. **Which records of it.** *Document Generation Availability Rule* records, each adding one comparison. **A template with no rules is available for every record of its table.**

Column	Meaning
Template	The template the rule belongs to
Attribute Path	The attribute to test. It may reach through lookups: `ownerid.businessunitid.name`

Column	Meaning
Operator	Equal, Not Equal, GreaterThan, LessThan, GreaterEqual, LessEqual, Contains Data, Does Not Contain Data
Value	Text, read as the attribute's type: `1` or `Active` for a choice, a GUID or a name for a lookup, `2026-01-31` for a date, `true` or `false` for two options
Group By	**Yes** puts the rule in the AND group, **No** in the OR group
Status	Only active rules count

How rules combine. The template is offered when **either** group holds:

```
(and-rule 1  AND  and-rule 2  AND …)    OR    (or-rule 1  OR  or-rule 2  OR …)
```

Rules a, b and c	Meaning
all Group By = Yes	a AND b AND c
all Group By = No	a OR b OR c
a and b Yes, c No	(a AND b) OR c

Intent	Attribute Path	Operator	Value	Group By
Only active cases	`statecode`	Equal	`0`	No
Only assigned cases	`ownerid`	Contains Data	—	No
Large accounts only	`revenue`	GreaterThan	`1000000`	No
One business unit	`ownerid.businessunitid.name`	Equal	`Head Office`	No
Active and high priority	`statecode` / `prioritycode`	Equal / Equal	`0` / `High`	Yes / Yes

An empty attribute behaves as in templates: *Equal* is not satisfied, *Not Equal* is. A rule whose path contains a mistake shows a warning icon on its template, with a message naming the rule; other templates are unaffected.

Rules are checked both when the list of templates is built and again when the document is generated. They decide which template fits which record: they are not a security boundary. A shape like `(a OR b) AND (c OR d)` needs a calculated column, or an `{{#if}}` inside the document.

9 Generating documents

One record or many

From a grid, select up to **250** records to get one document: each record is rendered from the same template and the results are joined by a page break (the default) or a line break. Which templates are offered is decided by the **first** selected record.

- **Page break.** Each record starts on a new page and keeps its own page setup, headers and footers; a header's `{{name}}` shows each record's own name. Page numbers run on through the file.
- **Line break.** Records follow each other on the same pages, so headers and footers are the first record's. Keep placeholders out of headers and footers in such templates.

The file is named after the template and the time, for example `Client Summary 2026-08-03 14-30-00.docx`.

Scope: whose access a document reflects

Scope	The data is read as	Choose it when
System *(default)*	the system	The document must be complete whoever prints it: an invoice, a contract.
User	the person generating it	The document must not show anything that person could not look up themselves.

Under either scope the person must be able to open the record the document is for. Templates, rules and queries are configuration, always read as system, so users need no privileges on those tables.

The subscription

Document Generation needs an active Bever Marketplace subscription, activated once in each environment. A System Administrator, or a user with the *Document Generation Administrator* role, opens the solution's configuration page, enters the license key and selects **Activate**. **Deactivate** releases the environment, for example before moving the license to another one.

Subscription	Documents	*View All Available Templates*
Active, Trial	generated	as usual
Past Due	generated	as usual; the configuration page shows how to fix the payment
Canceled, Paused, or no valid license	**not** generated	a message saying why

The subscription is checked every time templates are listed or a document is generated.

10 Troubleshooting

An empty value printing as nothing is correct output, not an error. Only structural problems stop a document: a placeholder that does not parse, an attribute that does not exist, a region that is never closed. The message names the placeholder, the template and the record at fault, and usually the fix.

Symptom	Cause and fix
The menu says "*No templates available*"	No eligible template is a Quick Template. Set *Quick Template* = Yes, or use *View All Available Templates*.
A template appears nowhere	It is inactive, its *Type* is not Word, its table is not this one, or its rules do not hold for this record. Check in that order.
A template shows a warning icon	One of its rules has a mistake in its *Attribute Path*. The message names the rule.
Every template appears for every record	There are no active rules: no rules means the whole table.
"***There is no valid license in this environment***"	An administrator activates one on the configuration page.
"*... has no attribute ...*"	A typo in the placeholder, or the field belongs to a related table: add the lookup hop.

Symptom	Cause and fix
"... is not a lookup and cannot be traversed"	A text column written as if it pointed somewhere, `{{name.x}}`.
"'query:Q' does not select the column ..."	Add `>` to the query.
"... reads through the lookup ... on a row of 'query:Q'"	Add a `link-entity` with an alias to the query and write `{{Alias.fullname}}`.
`{{x ?? "default"}}` prints nothing for a blank field	`??` falls through on null only. Use `{{If(x = null, "default", x)}}`.
"the '{{#else}}' marker must start its own paragraph..."	Use the inline `If(…)`, or put the branches in separate paragraphs.
Choice labels in the wrong language	Check *Language* and *Use Current User Selected Language*; the language must be enabled in the environment.
A date looks a day or some hours off	The document follows the column's **Behavior**, exactly as the form does for the same user.
"You do not have permission to read '...' records"	The template's Scope is User. Grant read access, or change the Scope.
"Select at most 250 records"	Select fewer records.
The template body is missing	No note on the template record carries a file. Attach the `.docx` again.

For anything not covered here, see the User Guide, section 10.

Cheat sheet

```
FIELDS
  {{name}}                                    field
  {{ownerid.businessunitid.name}}             nested lookup, any depth
  {{[contact^parentcustomerid].fullname}}     oldest related record (1:N)
  {{[new_tag^^new_case_tag].name}}            oldest related record (N:N)
  {{First(MyQuery).new_total}}                first row of a named query
  {{First(relationship:new_case_notes).subject}}   first related record
  {{ownerid?.internalemailaddress ?? ownerid?.emailaddress}}   ?. = empty, not an error
  {{$parent.name}}  {{$root.name}}            enclosing record, the document's record

VARIABLES
  {{$user.fullname}} {{$now}} {{$today}} {{$recordid}} {{$entityname}} {{$index}} {{$count}}

OPERATORS
  ??  default (null only)     +  join text     =  <>  !=  <  <=  >  >=     AND  OR  NOT  ( )

CONDITIONALS
  {{If(cond, value, default)}}      {{If(c1, v1, c2, v2, …, default)}}

FORMATS
  {{d:dd/MM/yyyy}} {{d:HH:mm}} {{d:date}} {{d:time}} {{d:"MMMM dd, yyyy"}}
  {{b:label}} {{b:text}} {{b:checkbox}} {{b:checkmark}} {{b:Agreed/Declined}}
  {{n:N2}} {{n:percent}} {{m:currency}} {{m:code}}
  {{c:value}} {{c:label}} {{ms:" / "}} {{l:id}} {{l:entitytype}} {{s:upper}}

LANGUAGE
  {{c:label:lcid=1049}}   {{d:"d MMMM yyyy":lcid=1033}}   {{n:N2:lcid=1049}}

REGIONS
  {{#each relationship:new_account_cases}} … {{/each}}     preferred
  {{#each query:MyQuery}} … {{/each}}
  {{#if cond}} … {{#else}} … {{/if}}

QUERIES
  a row holds only what the FetchXML selects; linked tables through a link-entity with an alias
  <fetch top="2">  first 2 rows      <fetch count="2" page="2">  rows 3 and 4      <fetch>  every row

ESCAPING
  \{{name}}   prints {{name}}
```