VARS, PARAMETERS, FORMULAS & GARAGE

The application allows the use of variables to store, operate, and assign values to or from fields. This feature provides additional power and flexibility to templates, enabling them to load and reuse values as part of calculations or to assign them in other areas or levels.

To assign a value to a variable, go to the field configuration and enter a descriptive name in the 'Variable ID' column. Any name is allowed, and the variable is case sensitive, meaning it distinguishes between uppercase and lowercase.

- To assign a field's value to a variable, place the variable name in the 'Variable ID' column.
- Conversely, to assign a variable's value to a field, place the variable name in the 'Formula' column.

In the following example template, we'll see how variables are used. Then, we'll replace a fixed value in a formula with an input parameter to make the template more flexible.

OE Template Ba <u>i</u> e CUSTNEWDISCOUNTS		
🖻 Copy-Paste 🔋 Export Data Excel 🛛 🔊 Run F	Process 📑 Save File 📲 Archive 🔝 Copy Template 🖷 Permissi	ions 🏾 💈 Vars & Formulas 🛛 🖀 Get and Enable All Fields 🛛 🖹 Delete Obsolete/Unexistent Field
General		
Code · · · · · CUSTNEWDISCOUNTS	Operation Insert-Modify	Enabled
Source Table ID	18 V Table Trigger	Use-Create Equivalence
Source Name · · · · · · · Customer	Permissions	0 Folder · · · · · · DEMO · ·
Description Increment discount to a cus	tomer Allow Requests	Template Use 1
Lines 🚽 🎬 New Line 🔅 Delete Line 🏻 🎰 Use Key	🔀 Clear Key 🗵 Excel Header 🕮 Excel Line 🔹 🕨 Indent	r 2
Target Table ID Target Table Name	Related Table Indentation ID Tag Operation	Table on Trig Key Fields Formu Filters
→ 18 : Customer	0 <u>18</u> Insert-	Mod 🗌 163 No 1
7004 Sales Line Discount	2 18 Insert-	Mod 🗋 11 Ves 1
 ↓ ↓ 		

In this example, we've defined two variables, **startDate** and **ldisc** for the discount line. In the first line, the **Starting Date** field's value is assigned to the **startDate** variable. When the **Ending Date** field is processed, the **startDate** variable, within the **Formula** column alongside a date expression, will perform a calculation to increase it by three months. The new value is then assigned to the **Ending Date** field.

Tab	TableData · 7004													
OE	Field Line	s	Ø~ ~ E -	+ New	🐯 Edit List	i D	elete 🔟 De	elete All	🔛 Get & En	abled All	🖃 Enat	oled All 🛛 🍾 D	isable All 🛯 📮 Check D	uplicates.
	Field		Field Name	Tag	Order †	Serie No.	Get Serie When	Get From	Field Value	Mapping (Imports)	Ena Map	Variable Id	Formula	Condi
	1		Code		10					0				No
	2		Sales Code		20					0				No
	3		Currency Code		30					0				No
	4		Starting Date		40					→ 0		startDate		No
	15		Ending Date		50	-				0		\rightarrow	startDate;+3M	No
	14		Minimum Quantity		55					0				No
\rightarrow	5	÷	Line Discount %		60					0		ldisc	ldisc + myincdisc	Yes
	5400		Unit of Measure Code		70					0				No
	5700		Variant Code		80					0				No
	21		Туре		90					0				No
	13		Sales Type		100					0				No

In the second case, the **ldisc** variable is assigned the field's value, then a formula is processed that increments this value using the previously defined parameter, **myincdisc**. Once the calculation is complete, the result is assigned to the **Line Discount %** field.

We'll likely use this second combination of field -> variable -> formula -> field frequently in a single line for auto-incrementing an entry number, document line, etc.

IMPORTANT: When processing a template line with level or **indentation** zero, all defined variables are initialized, whereas parameters retain their values throughout the entire execution process.

In this example, we are implementing a quarterly date increment and add **ten percentage points** directly to the discount after executing this process twice. Initially, only discounts for January to March existed.

TableData · 7004																
OE Field Lines	@ ∽	+ New 👿	Edit List 🧃)Delete 🗎	Delete All	👫 Get & I	Enabled All	🔚 Ena	bled All 🎽	o Disable All	📭 Check	Duplicates.				
Field ID T	Field Name	Tag	Order Serie † No.	e Get Serie When	Get From	Field Value	Mapping (Imports)	Ena Map	Variable Id	Formula		Condi				
\rightarrow <u>1</u> :	Code		10				0					No				
2	Sales Code		20				0					No				
3	Currency Code		30				0					No				
4	Starting Date		40				0		initDate	initDate;	;+3M	No				
15	Ending Date		50				0		finalDate	finalDate	e;+3M	No				
14	Minimum Quantity		55				0					No				
5	Line Discount %		60				0		ldisc	ldisc + n	nyincdisc	No				
5400	Unit of Measure Code		70				0					No				
5700	Variant Code		80			Custome	r 10000 Adat	um Cor	poration						✓ Saved	□ <i>2</i>
21	Туре		90													
13	Sales Type		100			Sales Line	e Discounts	Ð	< ∧ ∧		New 📑	Edit List	🗊 Delete		Ŀ	$\nabla \equiv$
						General		Г	\square							
						Sales Typ	e Filter		Customer		\sim	Type Fi	lter	None		\sim
						Sales Coo	le Filter		10000			Code F	ilter			
						Starting [Date Filter					Curren	cy Code Filter			
						0	Sales Type †		Code 1	Unit of Measure †	Code	Minimum Quantity↑	Line Discount %	Starting Date †	Ending Date	
						\rightarrow	Customer	÷	1896-S	PCS		1,00	1,05	01/01/2024	31/03/2024	4
						0	Customer		1896-S	UDS		5,00	2,50	01/01/2024	31/03/2024	4
						0	Customer		1896-S	PCS		1,00	11,05	01/04/2024	30/06/2024	4
						0	Customer		1896-S	UDS		5,00	12,50	01/04/2024	30/06/2024	4
						0	Customer		1896-S	PCS		1,00	21,05	01/07/2024	30/09/2024	4
						0	Customer		1896-S	UDS		5,00	22,50	01/07/2024	30/09/2024	4

PARAMETERS

Parameters are also variables but are defined in the **PARAMETERS** section since they are not tied to any specific field. They are used to set initial values for the template and can be referenced in the **Formula** column, like other variables, or in the **Filters** section under the **'Var or Formula'** column.

In the previous example, we can set a fixed discount increment value of **10**. When launching this template, the parameter values can still be adjusted if **visibility** has been enabled (in the **Requester Page** section).

The **PARAMETERS** section has four fields:

- 'Name': The variable name used to store the value, which can be referenced in a formula or as a variable-type filter in the template.
- 'Name for Users': A name or description that the user will see, for example, when launching the template.
- 'Value': The value of the parameter.
- 'Enabled': if activated, it will be processed; otherwise, the process will disregard it.

Note: If it is not enabled and we have the parameter in a 'Formula' column, the parameter will not be replaced by any value other than its own written text string (name). In this case, if the formula is mathematical, the calculation will fail.

Para	meters 🗦 🦉 New Line	i x	Delete Line	Ŀ	E2	
	Name 🕈		Name for Users	Value	Enabled	1
\rightarrow	myincdisc	÷	Increment Discount in	10	v	

WARNING: Template parameters are **ALWAYS READ**. If we have parameters with a fixed value defined in the template and it is enabled, <u>any internal or external process will</u> <u>read and process them accordingly</u>. If an entity is received from another company because it was launched with the **'Transform at'** field set to **Target** or **Mixed**, the fixed parameter value configured at the destination will be factored into the calculations where applicable.

FORMULAS

The application supports three types of formulas:

- 1. **Text:** Allows concatenation of fixed text with a text variable, such as an invoice number or any other data.
 - Example: Factura myInv -> Factura INV-0001
 Here, myInv is a variable to which the template process has previously assigned a value (in this case, INV-0001).
- 2. **Date:** There are three options for calculating dates:
 - today and workdate: Business Central variables that provide the current and work date, respectively.
 - Date expressions: A date can be calculated using Business Central date expressions. The base date can be placed in the Variable Id column or included directly in the formula, separated by a semicolon.
 - Example: startDate;+1M-1D \rightarrow Adds one month and subtracts one day from the date in the startDate variable.
 - \circ $\;$ You can combine reserved words like today with date formulas:
 - Example: today;+1Y \rightarrow One year from today.
- 3. **Mathematical:** Arithmetic calculations use Business Central functions, which support basic operations like addition, subtraction, multiplication, division, and exponentiation. It's recommended to first test formulas on the **Check Formulas** page (in the **Inspect & Wizards** menu) or use the action of the same name in the field configuration page, substituting variables with numerical values.
 - Example: netAmt + (netAmt * taxPer / 100)
 - Here, **netAmt** is the amount, and **taxPer** is the tax percentage. The result is the total amount.
 - Note: Business Central's calculation function doesn't allow consecutive operators (e.g., 100/-10). To prevent errors, the application automatically wraps the second operator with parentheses (100/(-10)), which Business Central accepts. Ensure that opening and closing parentheses match, as mismatched parentheses also cause errors. If an error occurs, the field's value will revert to its original value (before attempting the calculation).

GARAGE

Garage is a utility that simulates processing selected data with a template, displaying the data read, records generated, variable values used, and calculations of formulas and their conditions—all without making actual changes to the database.

In Open Entity, our template acts like an engine that can be tested by providing the right "fuel."

There is no direct access to Garage from the Control Panel or Dashboard. Instead, there are two Take to Garage actions located in the Inbound and Outbound folders (within Imports and Exports) and one more in the Archive List within Archive & Equivalences. Clicking Take to Garage while positioned on an entity line or file line will take you directly to the Garage page.



To test an entity with **Garage**, it's essential that a <u>template is assigned</u>. If the entity comes from the **Inbound** or **Outbound** folders, the destination template will be selected automatically. For an entity from **Archive**, if the template code wasn't propagated (during **Load File**), you can manually edit the **'Template Source'** field and add one.

After selecting **Take to Garage**, an empty **Garage** screen will appear. Click **'Start Engine'** to begin data processing. Once finished, a screen with three main sections will display the results.

Note: Certain operations with specific tables may generate "false errors." This typically happens during write and validation operations where Business Central's internal code attempts to directly read a record that, theoretically, doesn't yet exist (like a newly created sales order), since the simulator doesn't make actual database changes.

In such cases, it's best to test the operation in a **sandbox** environment before running it in production.

SECTIONS

- 1. **GENERAL:** Contains the basic data of the **Archive** line or the **Entity** line if the source is from the **Inbound/Outbound Folders**.
- 2. **DATAITEMS & BUFFER:** This section is divided into two parts:
 - **DataItems**: Represents the records read according to the template schema.
 - DataBuffer: Shows the data read and processed for each line of the DataItem. Switching from one DataItem to another will also update the data displayed in each section accordingly.

Garage	- 0			€}									~ ×
Manage	🕞 Start Engine	View Records	🕞 View Replaces	🏷 View Vars	🔢 View Formulas	∫₀ View (Conditions	🛄 View Schema and Index	es Page				0
General												$Record{\smallsetminus}$	
Entry No.						6	Source · ·		ARCHIVE		- 1	Value	
Template ·			CUSTNEWDISCOUN	NTS			Perspectiv	/e	Target		- 1	Sales Line Discount: Item	n,1896-S,Custo
Datalten	ns & Buffer										_		
Dataltems	/						Data Buffe	er∨					
	Major		Level Value					Major	Level	Value		Replacements \smallsetminus	
	1		0 Custom	ner			\rightarrow	<u>2</u> :	2	170 - Code : 1896-S			
\rightarrow	2	: :	2 Sale	s Line Discount				2	2	171 - SalesCode : 10000		Value	
	3		2 Sale	s Line Discount				2	2	172 - CurrencyCode :		(There is nothing to	show in this view)
								2	2	173 - StartingDate : 01/01/23		(mere is nothing to .	sion in allo neury
								2	2	174 - EndingDate :			
								2	2	175 - MinimumQuantity : 10			
								2	2	176 - LineDiscount : 13,5			
								2	2	177 - UnitofMeasureCode : UDS		Vars∨	
								2	2	178 - VariantCode :		vars -	
								2	2	179 - Type : Item		Value	
								2	2	180 - SalesType : Customer		-ttD-t- 04/04/02	
								2	2	181 - SalesLineDiscount :	•	startDate = 01/01/23	:
													Classic
													Close

- 3. FORMULA & CONDITIONS: Also divided into two panels:
 - The **left panel** displays the formulas that have been processed.
 - The **right panel** shows the defined conditions, indicating whether each condition was met. It includes both the values obtained during processing and those defined in each condition group.

On the **right side** of the main page, three **FactBoxes** provide details about:

- The **Record** that would have been saved.
- Substitutions or Replacements made.
- Variable values for the currently selected DataItem.

These FactBoxes can be shown or hidden using Business Central's standard icon in the upper-right corner of the page, represented by an 'i' inside a circle. If you want to view all records, substitutions, formulas, etc., you can use the **View** actions at the top of the page.

Garage - 0							$_{P}$ $\stackrel{\scriptstyle \swarrow}{\times}$
Manage 🗔 Start Engine 😥 View Records 🐘 View Replaces 🥻 View Vars 🔚 View Formulas	s f o Vie	ew Con	ndition	s 🗰 View Schema and Indexes Page			0
General >							Record \lor
DataItems & Buffer >							Value
Formula & Conditions							Sales Line Discount: Item, 1896-S, Custo
Formulas V		-	Condi	tions∨			
Value				Value			
today = 28/01/24			\rightarrow	GROUP CUSTDISCCAMPAIGN :	- E	*	Replacements \vee
today;+3M = 28/04/24				BEGIN			
→ Idisc + myIncrement = 17	÷			City: Atlanta Atlanta Chicago => YesOr			Value
Ν				Sales (LCY): 223.598,4 > 30.000 => Yes			
6				END			(There is nothing to show in this view)
				Condition GROUP = Yes			
				GROUP ITEMDISCLIMITATED :		1	
				BEGIN		_	
				Code: 1896-S 1896-S 1900-S => YesAnd		-	Vore
				Minimum Quantity: 10 >= 10 => Yes		-	vars ~
				END		•	
				Condition GROUP = Yes			Value
							startDate = 01/01/23
				CUSTDISCCAMPAIGN(Yes) And ITEMDISCLIMITATED(Yes) => Yes			[
				CONDITION FULFILLED = Yes			Close
				-			