20 Tips to Level Up Your Business Central

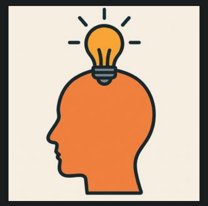
TEDDY HERRYANTO

Teddy Herryanto

- D365 Business Central Lead Consultant at Theta NZ
- thatnavguy.com
- https://www.linkedin.com/in/teddyherryanto/

Teddy Herryanto

20 Tips to Level Up Your Business Central


Know-How


Performance


Interface

Testing

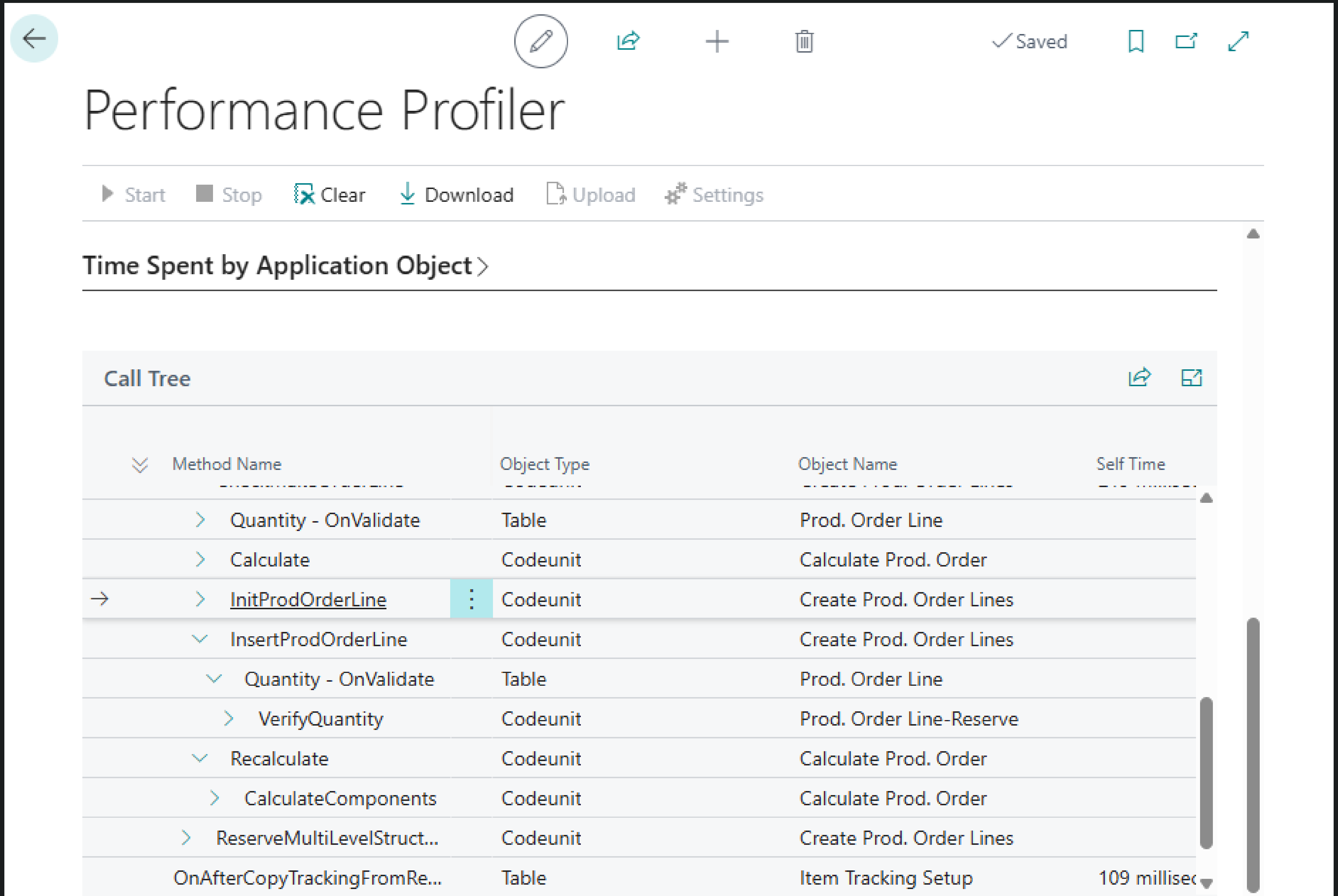

Feedback

#1 Master Your BC Tool

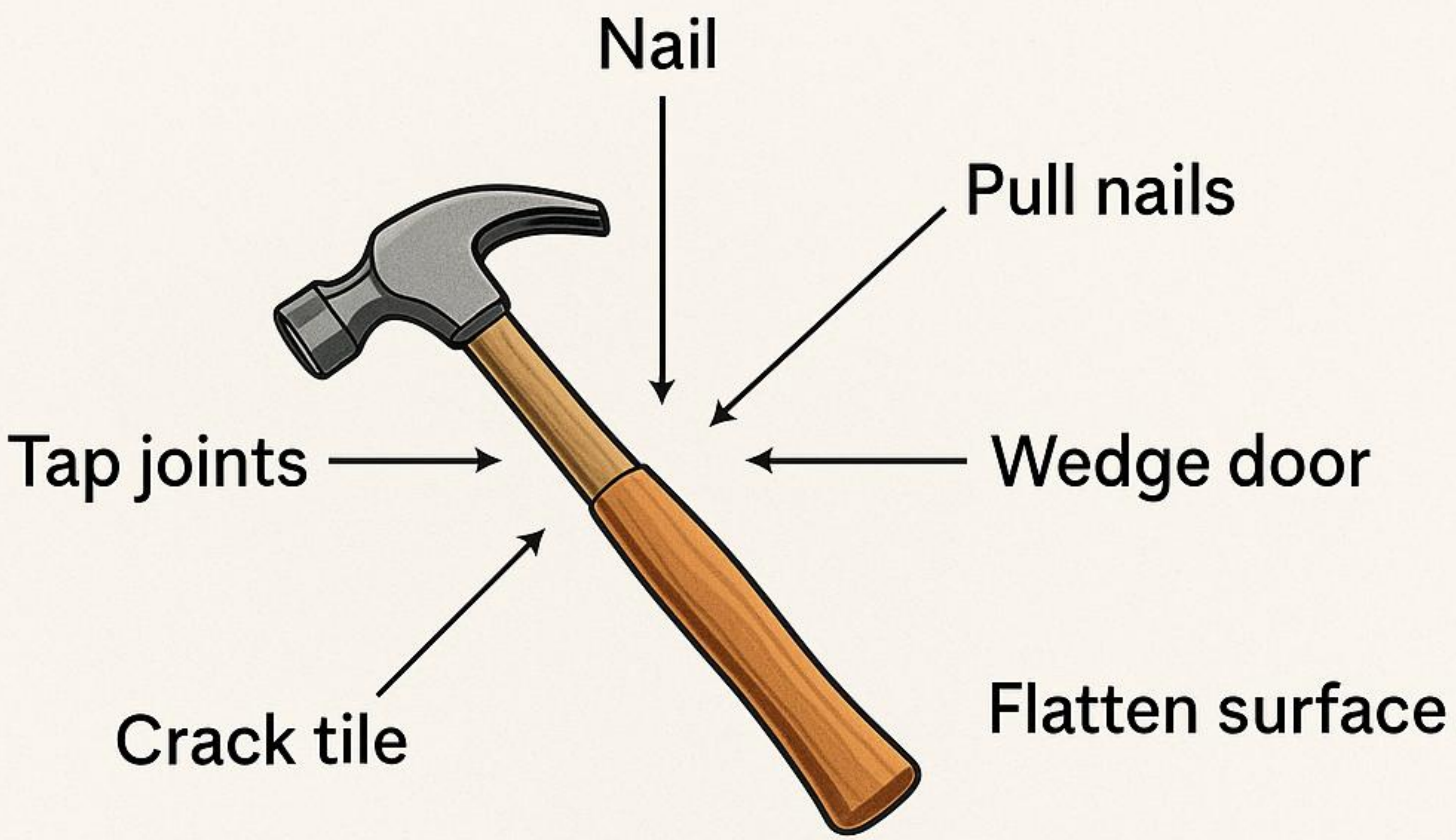
BC Tools and Utilities

- Configuration Package
- Configuration Template
- Admin Center
- Open / Edit in Excel
- Event Recorder
- Data Exchange Definition
- Profile and Personalisation
- Retention Policy
- Performance Profiler
- Page Inspection

Performance Profiler



Don’t Just Use the Tool. Master It.



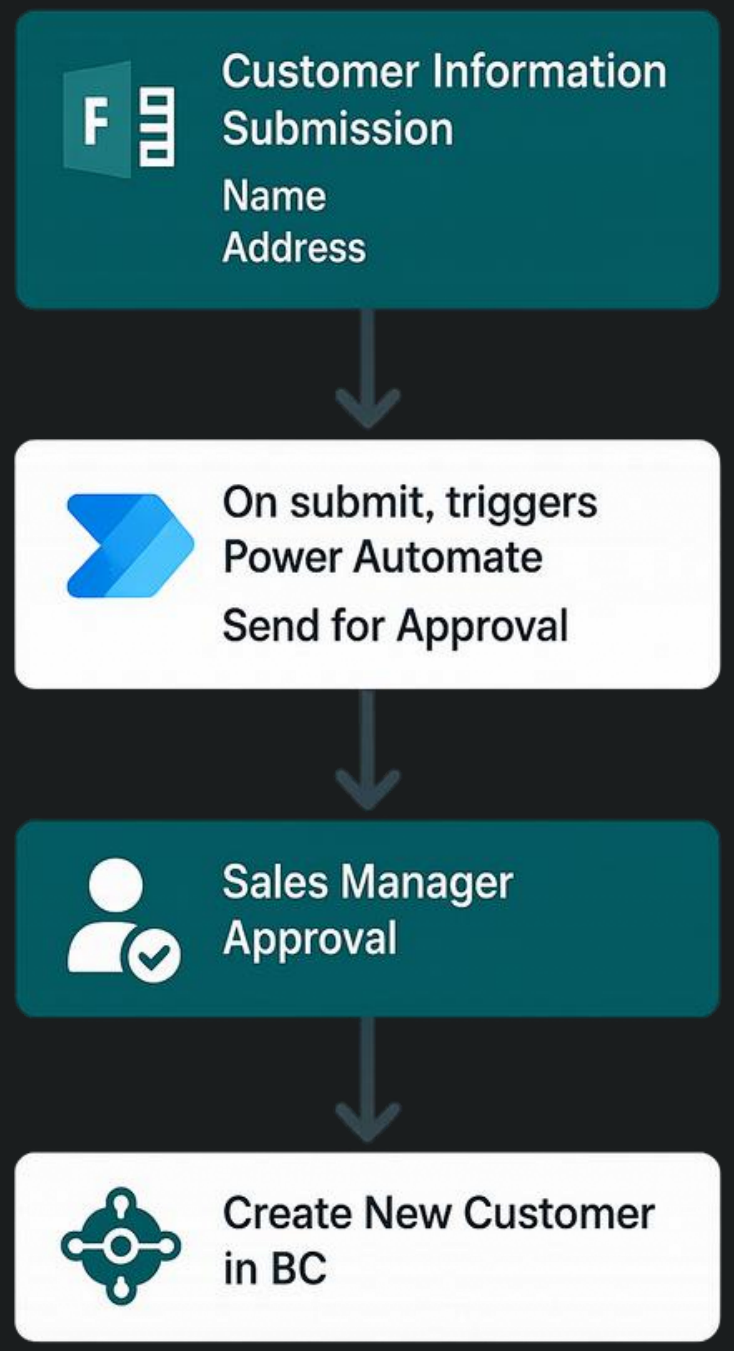
One tool. Many possibilities.

#2 Don't Force BC to Do Everything

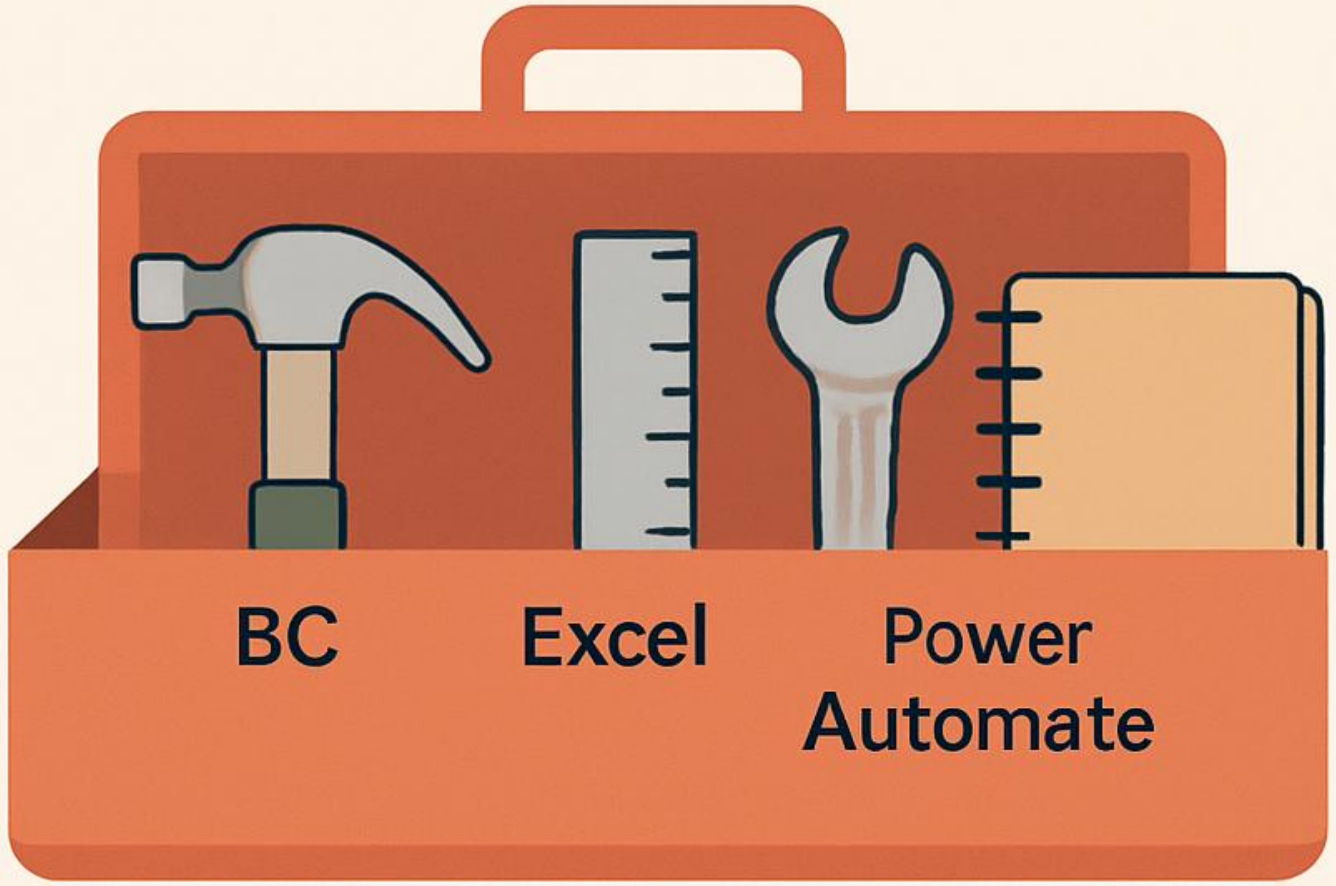
Is BC the right tool for the requirement?

- Data Entry
- Document and Reporting
- File Storage
- Workflow Automation
- Scheduling and Planning
- Integration

Customer Onboarding System



- Microsoft Form: Create a form to collect customer information.
- Power Automate: Set up a flow that triggers when the form is submitted. This flow would send an approval request to the Sales Manager and wait for the Sales Manager's response.
- Sales Manager Approval: The Sales Manager can approve or reject the request via email.
- Business Central: Once approved, the flow proceeds to automatically create a new customer in Microsoft Dynamics 365 Business Central using the information provided.



Pick the Right Tool for the Task

#3 Know where to use Copilot

Create Documentation

Prompt:
Convert below steps into a manual on how to create sales order in Business Central. Make sure that the steps are easy to understand.

- Type "Sales Order" in the search bar.
- Select the "Sales Orders".
- Click New
- Select the customer
- Enter the item or select from the list
- Fill in Quantity
- Confirm the price
- Release the sales order

Create Documentation

Here is a simple manual on how to create a sales order in Business Central:

How to Create a Sales Order in Business Central

1. **Open the Search Bar**
   - At the top of the Business Central window, click on the search icon or press `Alt + Q`.
2. **Search for Sales Orders**
   - Type "Sales Order" in the search bar.
   - Select "Sales Orders" from the search results.
3. **Create a New Sales Order**
   - Click the **New** button to start a new sales order.
4. **Select the Customer**
   - In the Customer field, enter the customer's name or select a customer from the list.
5. **Add Items**
   - In the lines section, enter the item number or select an item from the list.
6. **Enter Quantity**
   - Fill in the quantity for each item you want to sell.
7. **Confirm the Price**
   - Check and confirm the price for each item. Adjust if necessary.
8. **Release the Sales Order**
   - When all details are correct, click **Release** to finalize the sales order.

These steps will help you create a sales order in Business Central quickly and easily.

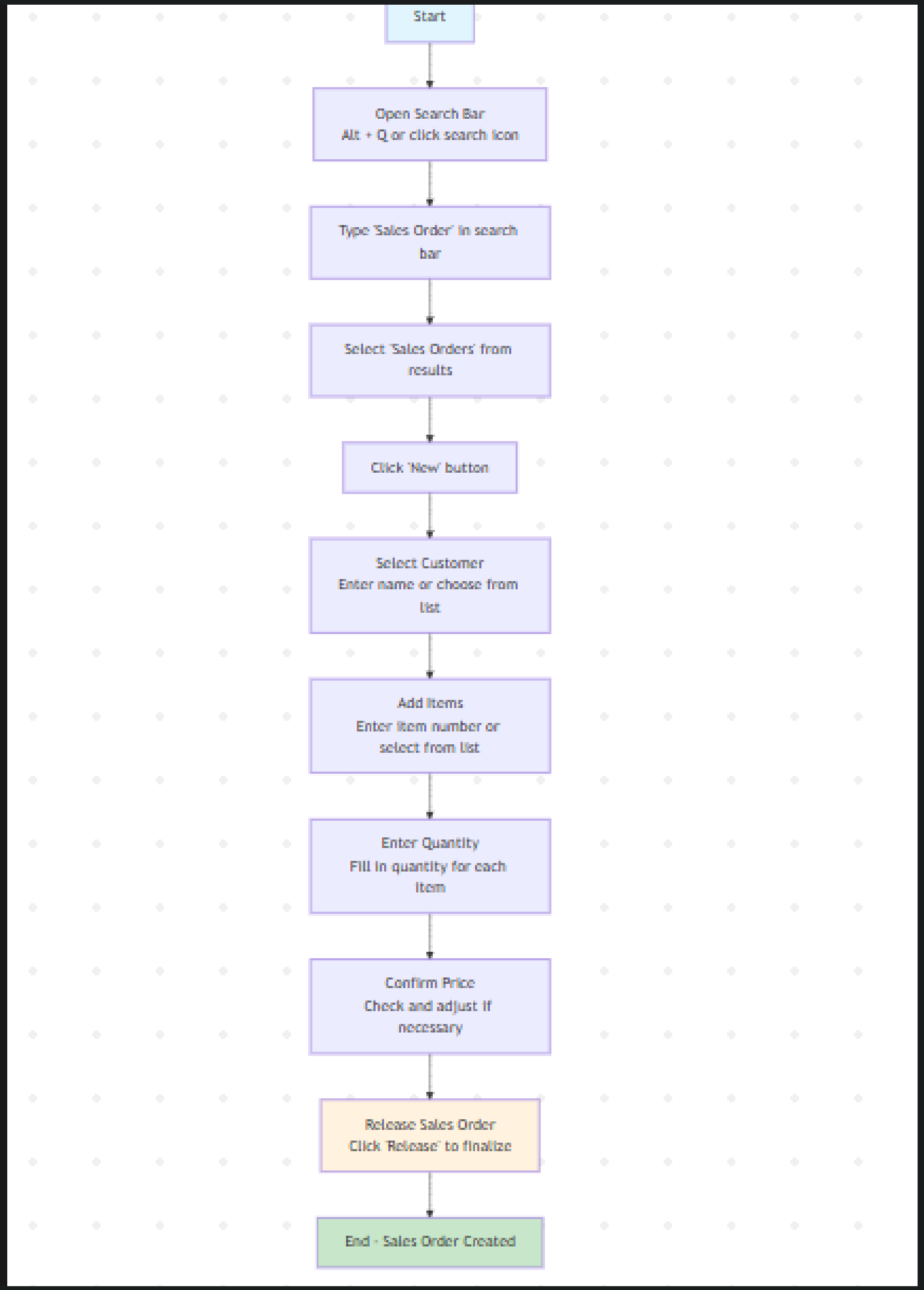
Create Flow Diagram

Generate mermaid diagram from it

Here's a Mermaid diagram based on the sales order creation process:

```
flowchart TD
    A[Start] --> B[Open Search Bar<br/>Alt + Q or click search icon]
    B --> C[Type 'Sales Order' in search bar]
    C --> D[Select 'Sales Orders' from results]
    D --> E[Click 'New' button]
    E --> F[Select Customer<br/>Enter name or choose from list]
    F --> G[Add Items<br/>Enter item number or select from list]
    G --> H[Enter Quantity<br/>Fill in quantity for each item]
    H --> I[Confirm Price<br/>Check and adjust if necessary]
    I --> J[Release Sales Order<br/>Click 'Release' to finalize]
    J --> K[End - Sales Order Created]

    style A fill:#e1f5fe
    style K fill:#c8e6c9
    style J fill:#fff3e0
```

This flowchart diagram shows the step-by-step process for creating a sales order in Business Central, with each step flowing logically to the next. The start and end points are highlighted in different colors for clarity.



Review Code

esCOGS.Page.al U ×   AL Page Designer

AL SalesInvoicesCOGS.Page.al > ...

```
e 60251 "Sales Invoices COGS_TNG"

0 references
trigger OnAfterGetRecord()
begin
    AverageUnitCost := 0;
    TotalCost := 0;
    SalesInvoiceLines.SetRange(salesInvoiceNo, Rec."Document No.");
    SalesInvoiceLines.SetRange(salesInvoiceLineNo, Rec."Line No.");
    if SalesInvoiceLines.Open() then begin
        while SalesInvoiceLines.Read() do begin
            UnitCost := (SalesInvoiceLines.costAmountActual + SalesInv
            TotalCost += UnitCost * -SalesInvoiceLines.InvoicedQuantit
        end;
        SalesInvoiceLines.Close();
    end;

    AverageUnitCost := TotalCost / Rec.Quantity;
end;

var
    9 references
    SalesInvoiceLines: Query "Sales Invoice Lines_TNG";
    2 references
    UnitCost: Decimal;
    3 references
    AverageUnitCost: Decimal;
    4 references
    TotalCost: Decimal;
```

CHAT   EXPLORER

```
TotalCost: Decimal;
```

2. **Query Column Naming**

Suggestion: Use PascalCase for query columns to match AL conventions:

```
column(SalesInvoiceNo; "Document No.")
column(SalesInvoiceLineNo; "Line No.")
column(ItemLedgerEntryQuantity; Quantity)
```

Potential Bugs

1. **Critical Bug: Division by Zero**

Location: Line 83

```
AverageUnitCost := TotalCost / Rec.Quantity;
```

Issue: If `Rec.Quantity` is 0, this will cause a runtime error. Although you filter for `Quantity = filter(<> 0)`, it's still a potent

Suggestion: Add a safety check:

```
if Rec.Quantity <> 0 then
    AverageUnitCost := TotalCost / Rec.Quantity
else
    AverageUnitCost := 0;
```

2. **Potential Bug: Division by Zero in Unit Cost Calculation**

Location: Line 76

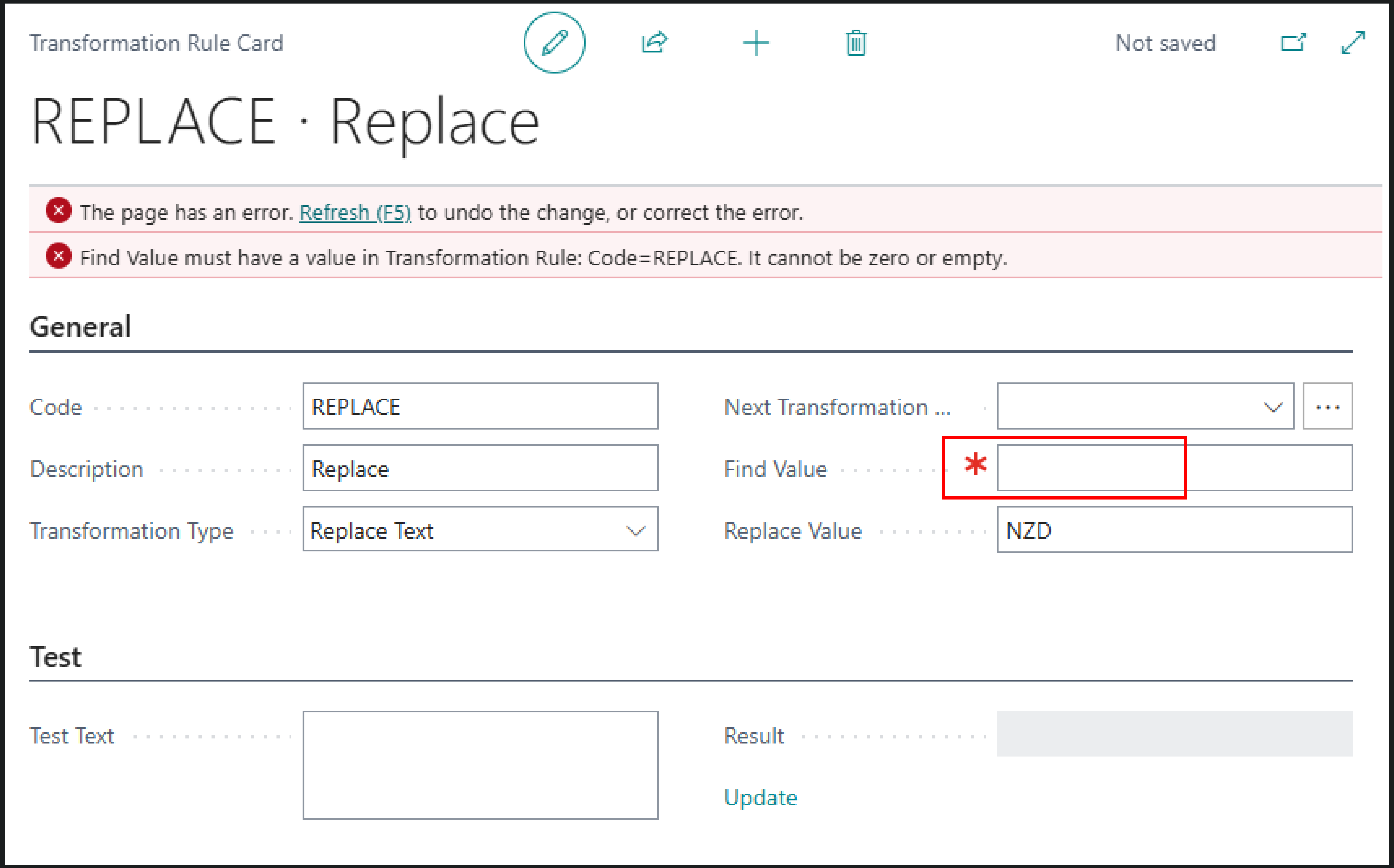
Don't outsource your thinking and creativity to AI.
Let it improve your way of thinking, not replace it.

#4 Learn Regex

What is Regex (Regular Expression)?

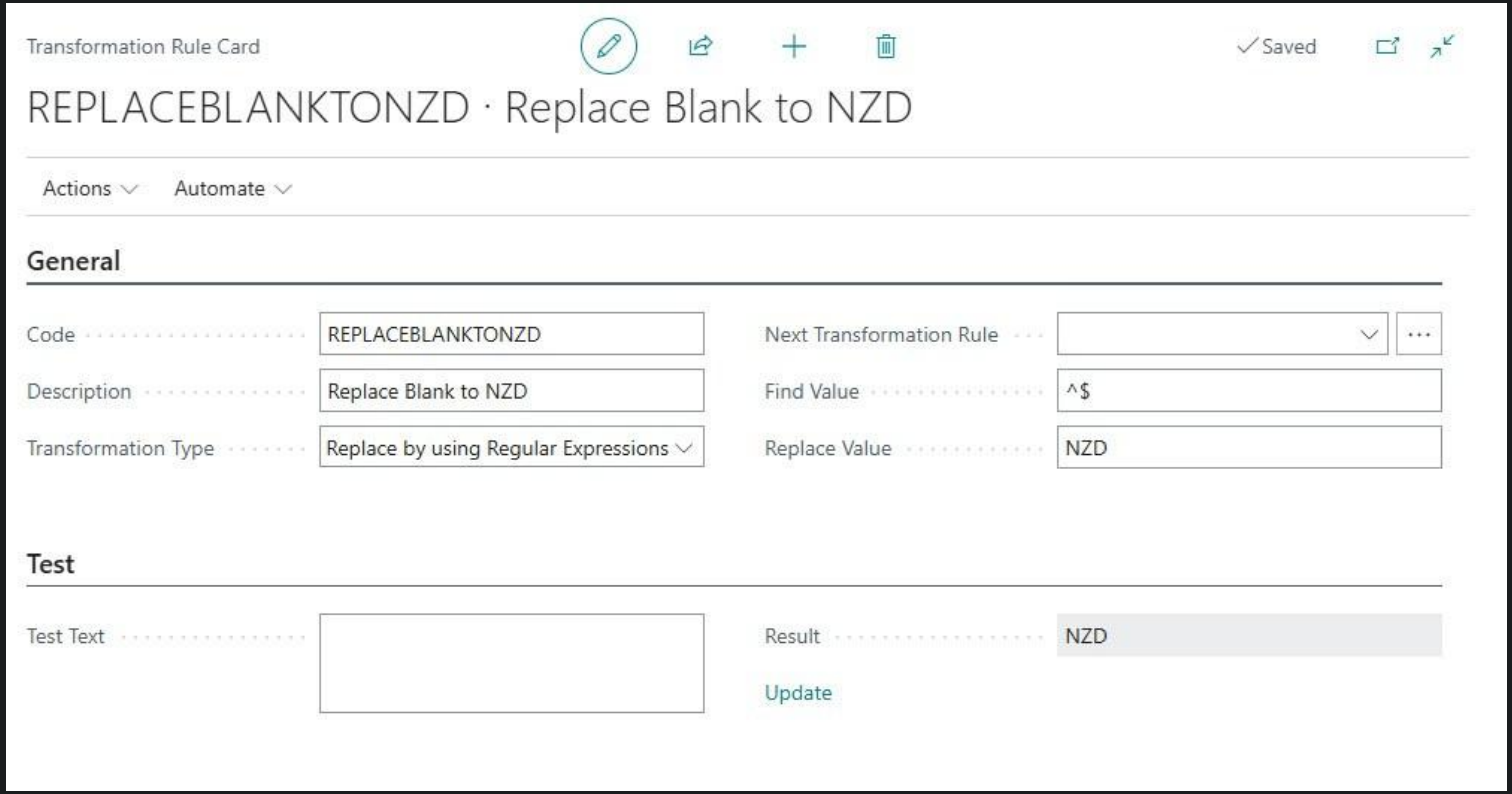
- Search for a pattern in text.
- Useful to validate, clean or transform text.
- More powerful than the normal Find and Replace.

Replace Blank



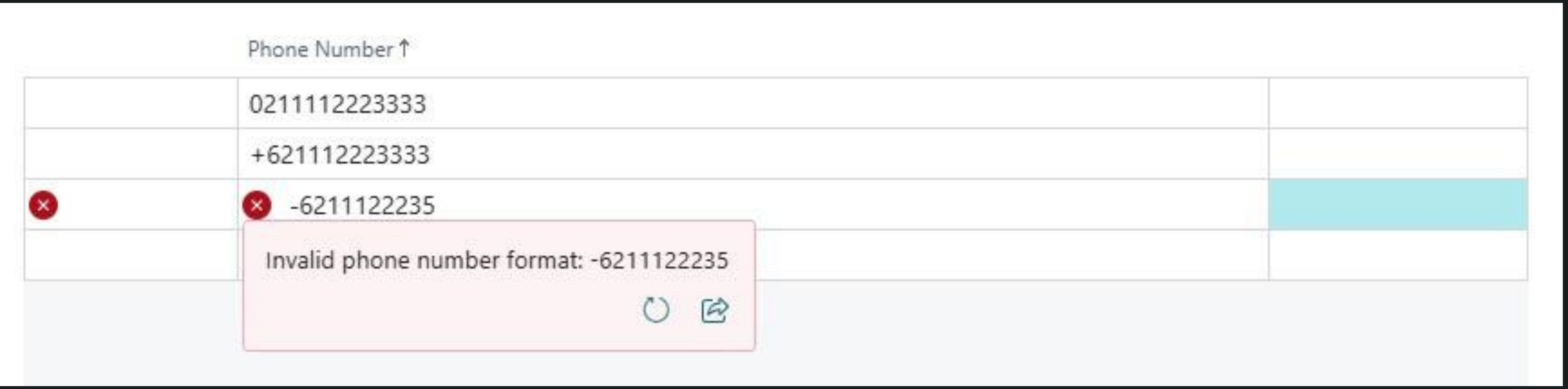
Regex to Replace Blank

^ → find the start of a string.
$ → find the end of a string.



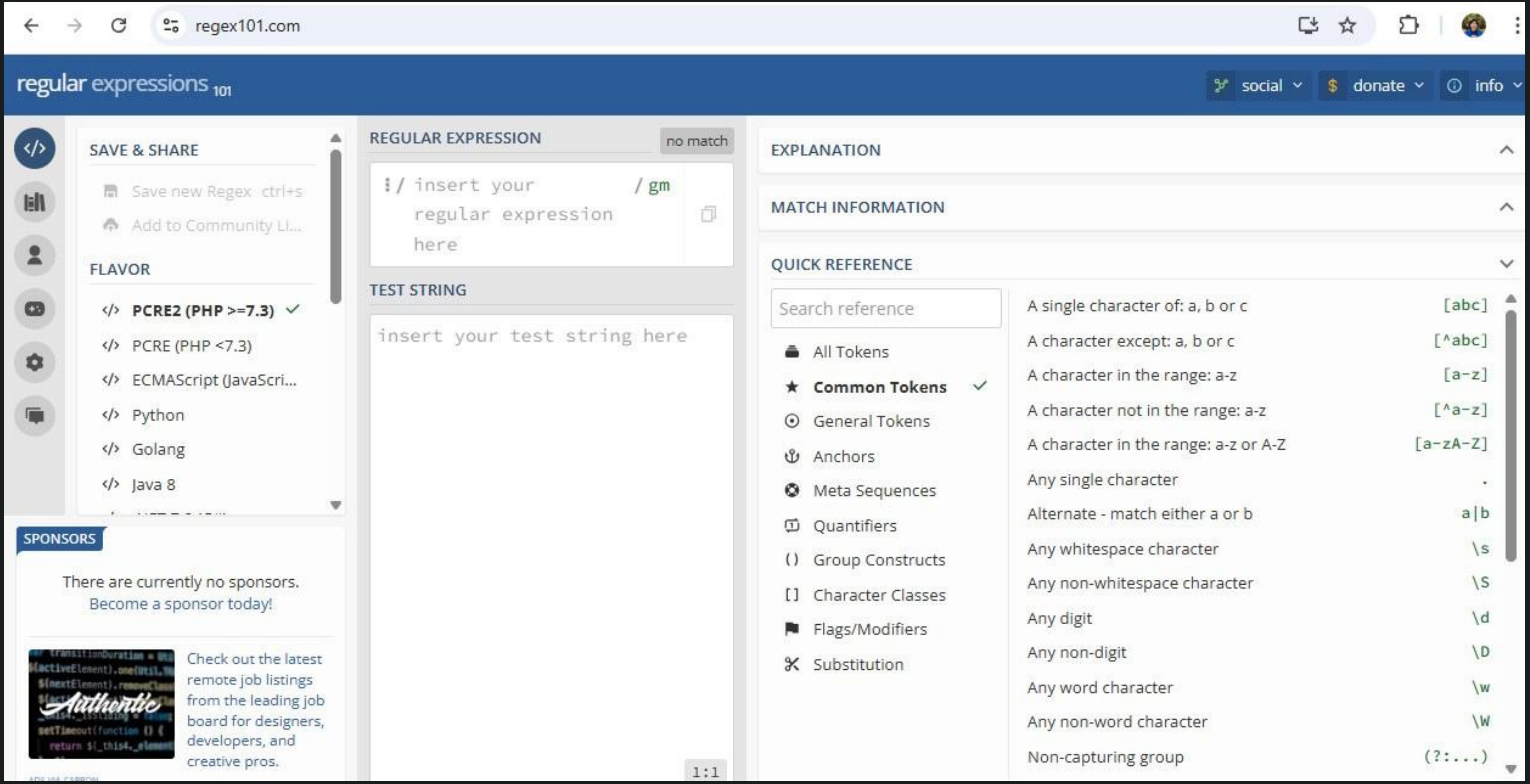
Regex Example

```
You, 9 minutes ago | 1 author (You)
fields
{
    4 references
    field(1; "Phone Number"; Text[30])
    {
        Caption = 'Phone Number';
        0 references
        trigger OnValidate()
        var
            Regex: Codeunit Regex;
        begin
            if not Regex.IsMatch("Phone Number", '^\+?[0-9]\d{1,14}$') then
                Error('Invalid phone number format: %1', "Phone Number");
        end;
    }
```



Regex

https://regex101.com/



Your Regex Friend

CHAT EXPLORER

help me create a regex for email validation

Here is a commonly used regular expression for basic email validation:

```
^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$
```

This regex checks for:

- One or more alphanumeric characters (including ., _, %, +, -) before the @
- An @ symbol
- One or more alphanumeric characters (including . and -) for the domain
- A dot (.)
- At least two alphabetic characters for the domain extension

This pattern works for most standard email addresses. If you need to implement this in AL code or want a more advanced pattern, let me know!

#5 Specify Data Access Intent

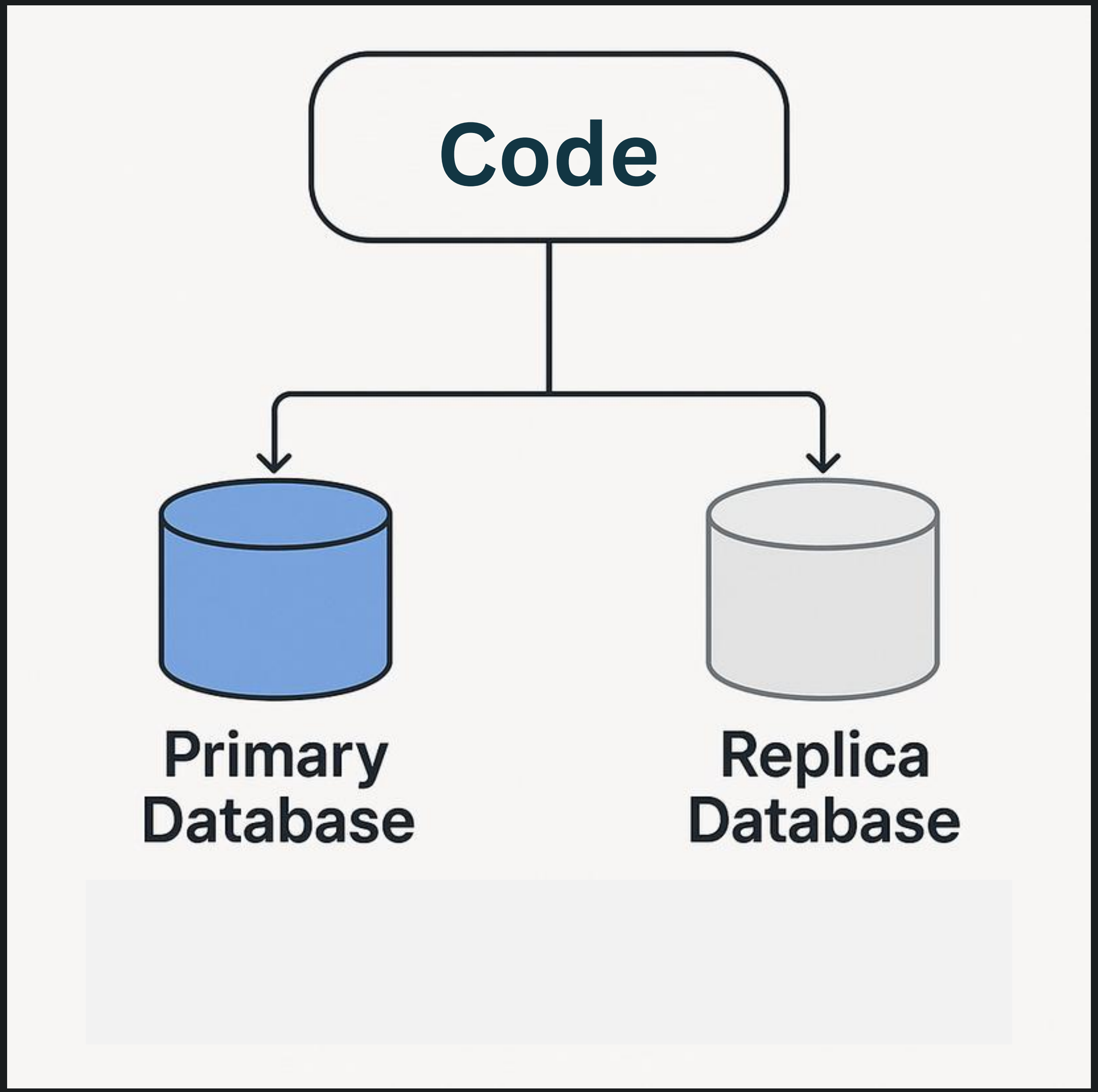
What is Data Access Intent?

- A setting that tells BC how you are going to access data
  - ReadOnly
  - ReadWrite

- ReadOnly signals the server to use a replica database when available.

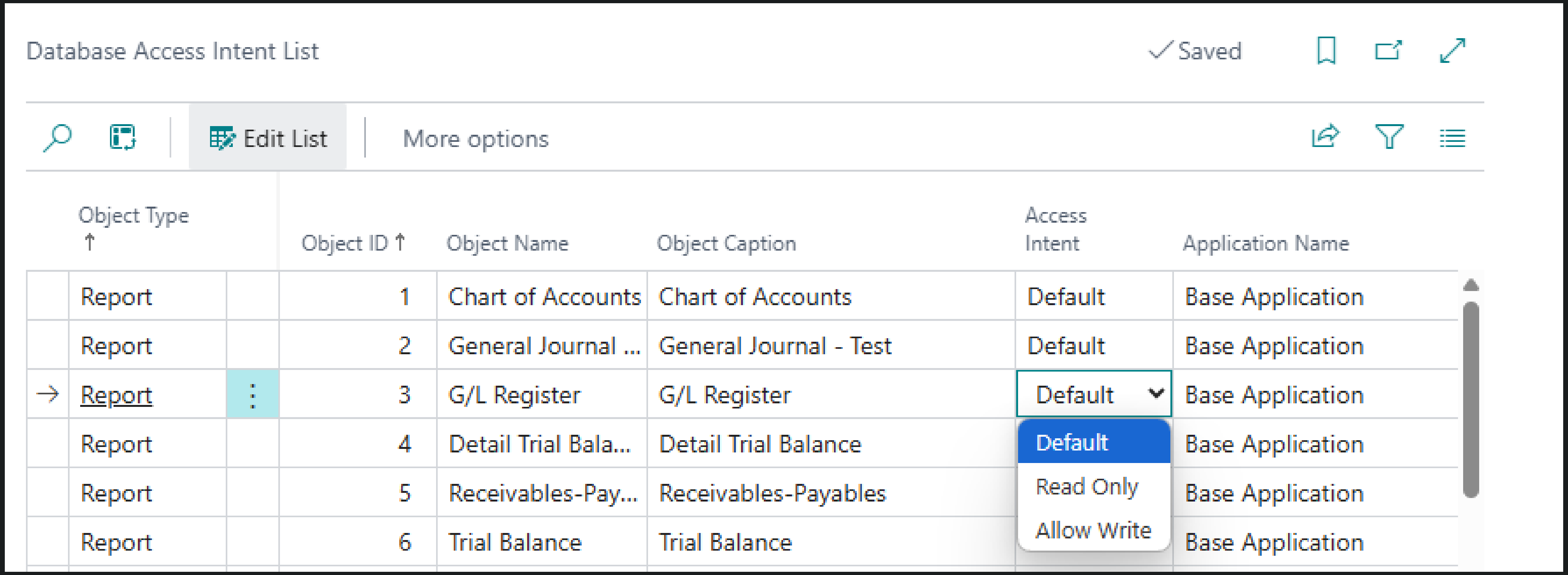
- Using the replica reduces the load on the primary database, ensuring better operation for others.

- Works on reports, API pages, and API queries.



How to use Data Access Intent?

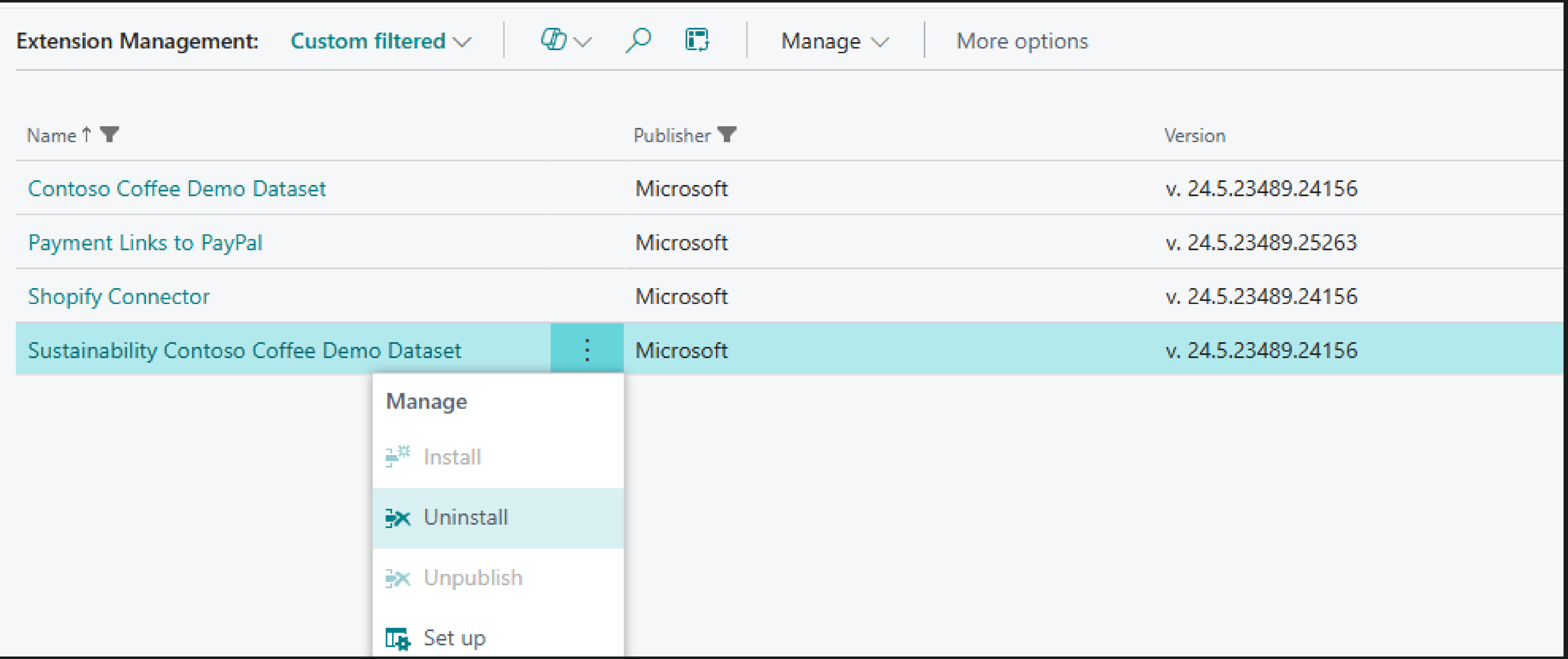
```
report 6 "Trial Balance"
{
    DefaultLayout = RDLC;
    RDLCLayout = './Finance/GeneralLedger/Reports/TrialBalance.rdlc';
    AdditionalSearchTerms = 'year closing,close accounting period,close fiscal year';
    ApplicationArea = Basic, Suite;
    Caption = 'Trial Balance';
    PreviewMode = PrintLayout;
    UsageCategory = ReportsAndAnalysis;
    DataAccessIntent = ReadOnly;

    dataset
    {
```



#6 Uninstall Unused Extensions

Unused Extensions

- Business Central environment comes with many pre-installed extensions.
- If you're not using them, consider uninstalling them.



Why Uninstall Unused Extensions?

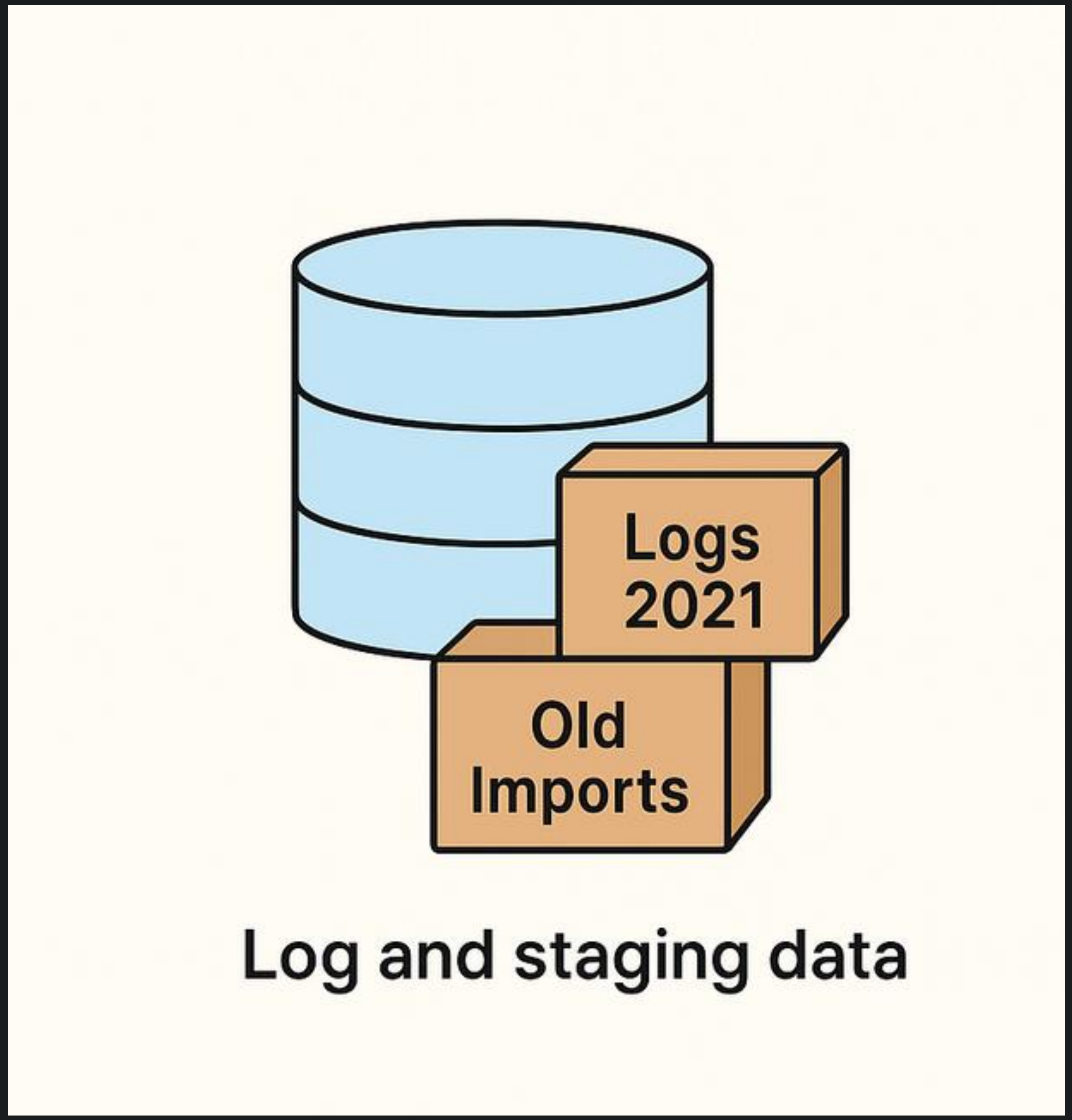
✅ Improve performance.
Less code means better performance.

✅ Avoid Errors.
Unused extensions can cause unexpected issues if accidently run.

#7 Schedule Data CleanUp

What to clean up?

- Log tables: Change Log Entries, Job Queue Log Entries, Integration Sync. Job.
- Staging Tables.
- Orphan medias.



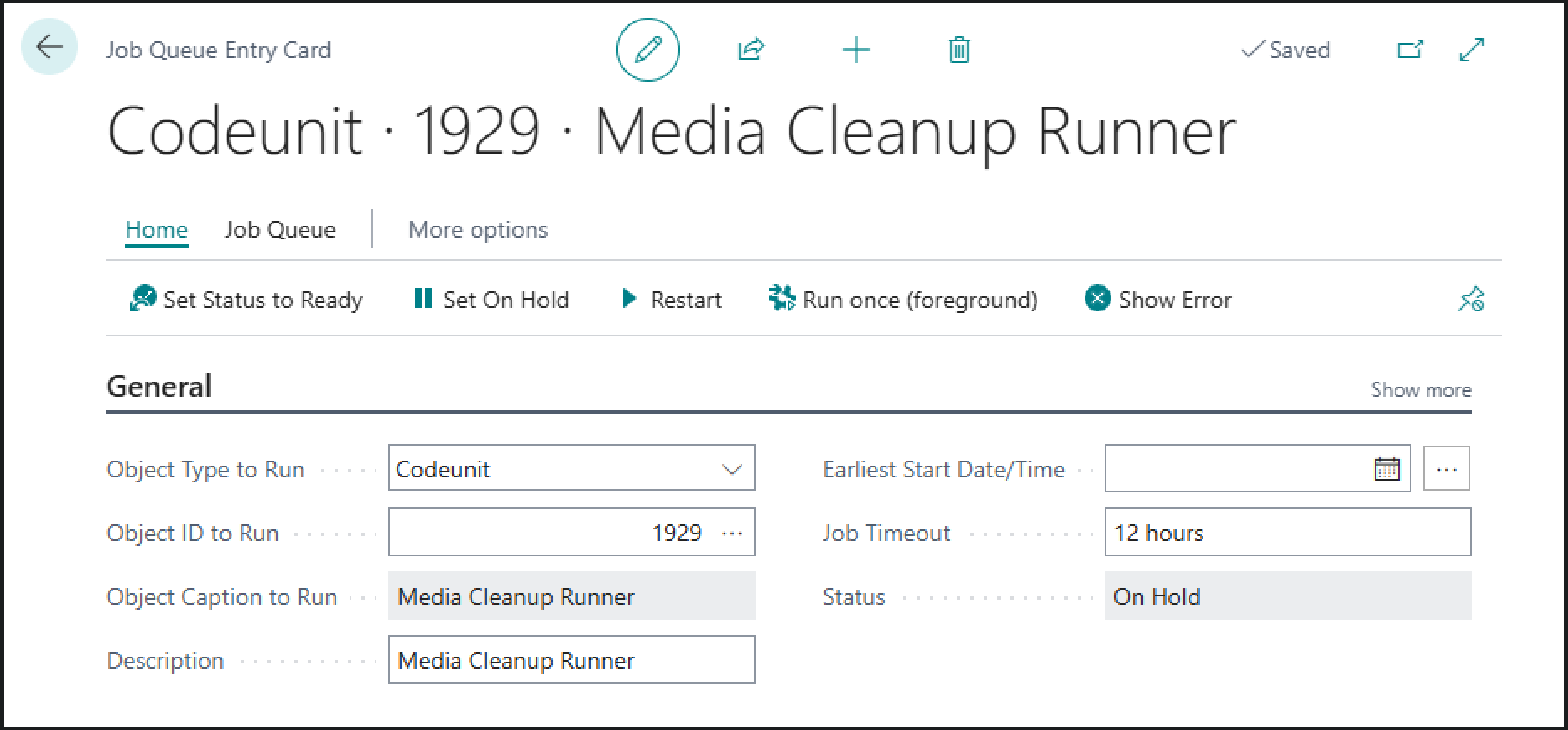
Why cleanup?

✅ Save database space.
Free up valuable space from unnecessary data. You spend less on database cost.

✅ Improve performance.
Smaller data = faster queries.

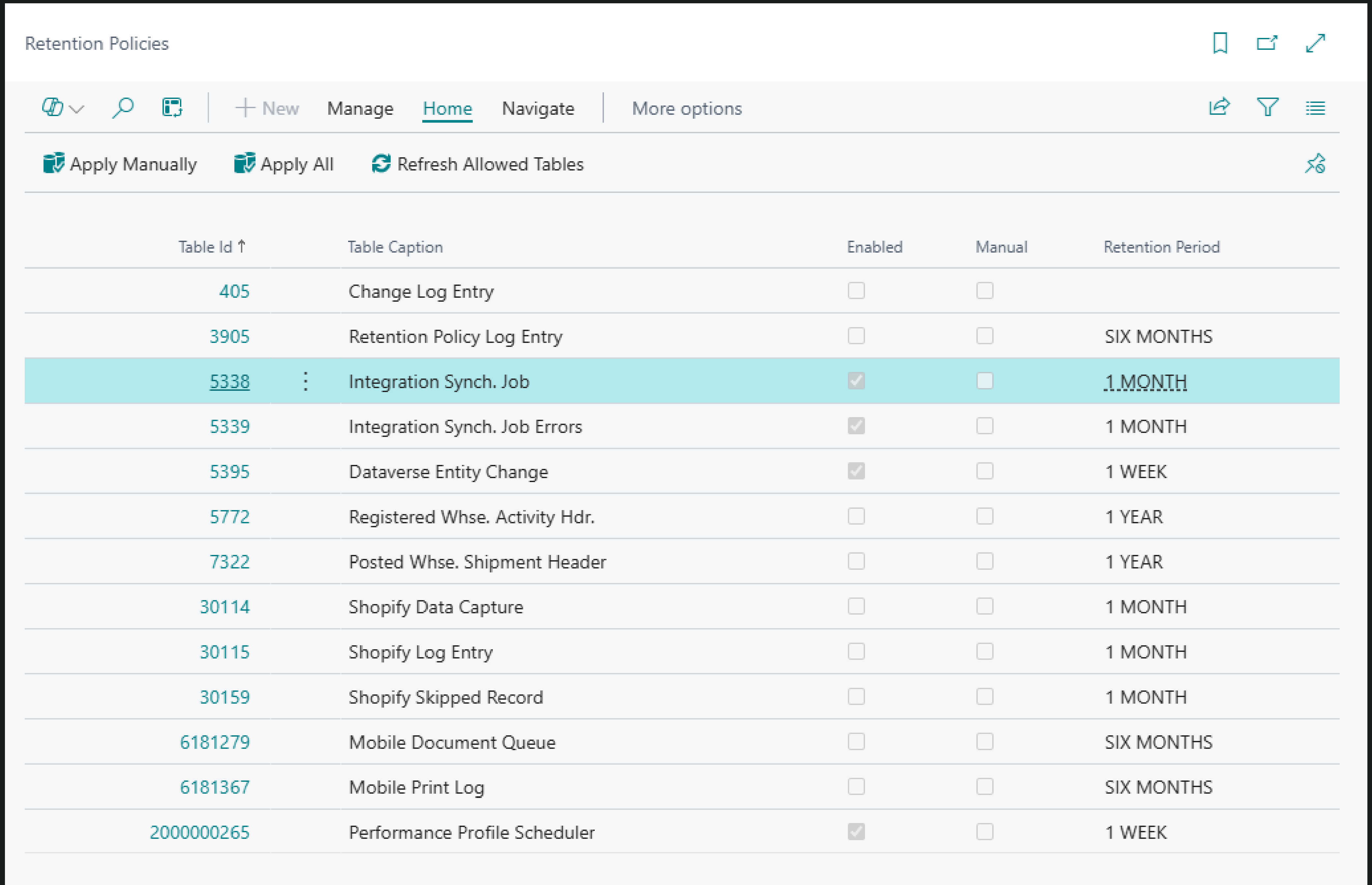
How to cleanup?

- Media Cleanup Runner



How to cleanup?

- Retention Policy


Retention Policies

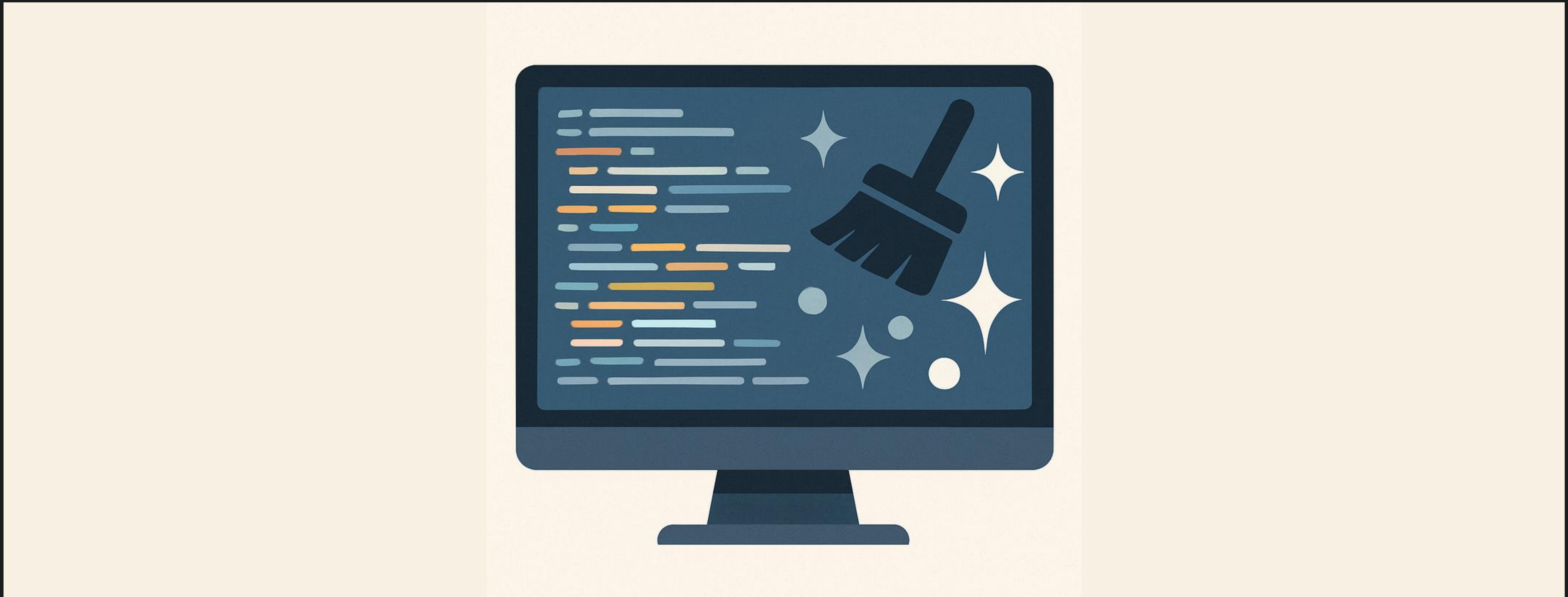
New | Manage | Home | Navigate | More options

Apply Manually | Apply All | Refresh Allowed Tables

Table Id ↑	Table Caption	Enabled	Manual	Retention Period
405	Change Log Entry	☐	☐	
3905	Retention Policy Log Entry	☐	☐	SIX MONTHS
5338	Integration Synch. Job	☑	☐	1 MONTH
5339	Integration Synch. Job Errors	☑	☐	1 MONTH
5395	Dataverse Entity Change	☑	☐	1 WEEK
5772	Registered Whse. Activity Hdr.	☐	☐	1 YEAR
7322	Posted Whse. Shipment Header	☐	☐	1 YEAR
30114	Shopify Data Capture	☐	☐	1 MONTH
30115	Shopify Log Entry	☐	☐	1 MONTH
30159	Shopify Skipped Record	☐	☐	1 MONTH
6181279	Mobile Document Queue	☐	☐	SIX MONTHS
6181367	Mobile Print Log	☐	☐	SIX MONTHS
2000000265	Performance Profile Scheduler	☑	☐	1 WEEK

How to cleanup?

- Develop your own data cleaner.



#8 Use Commit Wisely

What is Commit (Database Commit)?

- Save all changes so far into the database permanently.
- Before COMMIT, if something goes wrong, Business Central can roll back.

Why avoid Commit?

- Harder to roll back.

- Risk of data inconsistency.
  If you commit too early and something fails, you can end up with half-completed transactions.

Why use Commit?

✅ Prevents total data loss.
If something goes wrong in the process, you don’t lose everything.

✅ Avoids duplicates.
If your process crashes after sending data out, the retry will not send the same data twice.

✅ Improve performance.
Any locks by the transaction are automatically released.

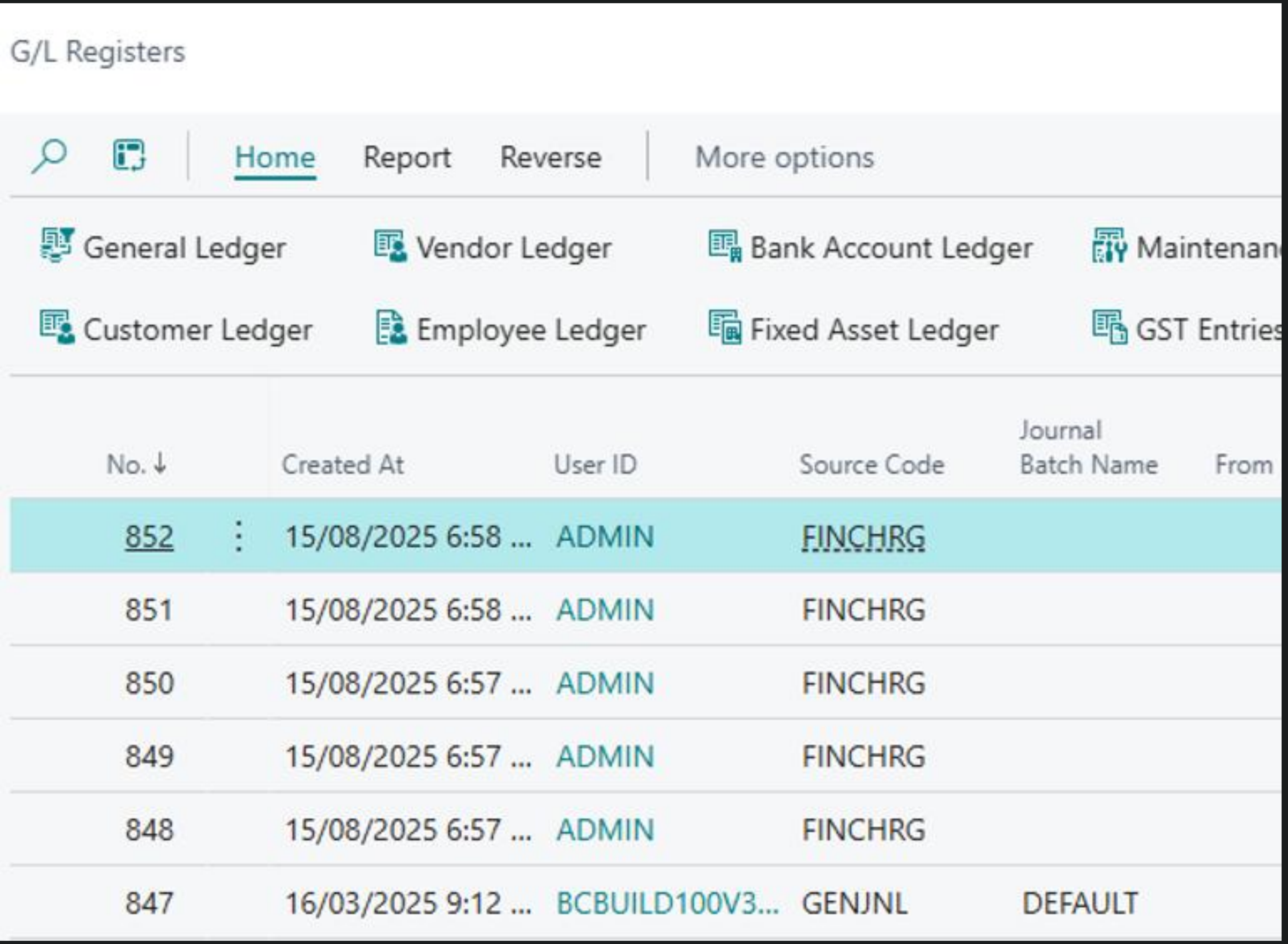
When to use Commit?

- Batch processing.
- Multi-step processes.
- Sending data to an external source.
- Only commit when you’ve completed a safe milestone.

#9 Descending sorting on the historical and log page

Descending Order

Descending = latest entries at the top



Why descending?

✅ Relevance

Most users are interested in the most recent entries.

✅ User Expectations

Users expect to see the latest updates first.

✅ Time-Saving

No scrolling.

How to change the sorting?

```
page 116 "G/L Registers"
{
    AdditionalSearchTerms = 'general ledger registers';
    ApplicationArea = Basic, Suite;
    Caption = 'G/L Registers';
    Editable = false;
    PageType = List;
    SourceTable = "G/L Register";
    SourceTableView = sorting("No.")
                      order(descending);
    UsageCategory = History;

    layout
    {
```

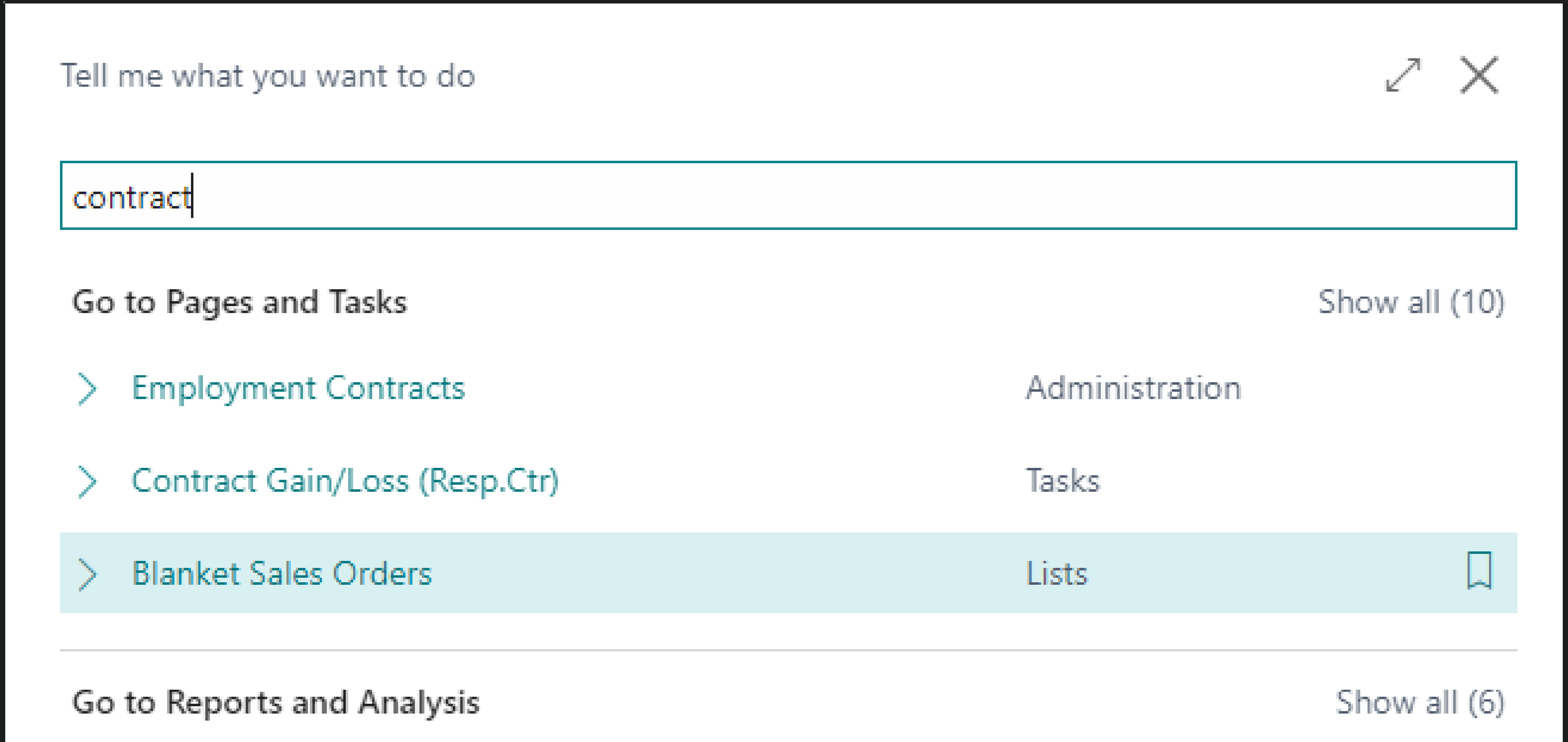
#10 Add Additional Search Term

What is Additional Search Term?

- Additional keywords, synonyms, nicknames, abbreviations.
- To help user find the right page easily.

How to use?

```
pageextension 70001 "Blanket Sales Orders_TNG" extends "Blanket Sales Orders"
{
    AdditionalSearchTerms = 'Contracts';
}
```



Why use it?

✅ Provide multiple terms to search.

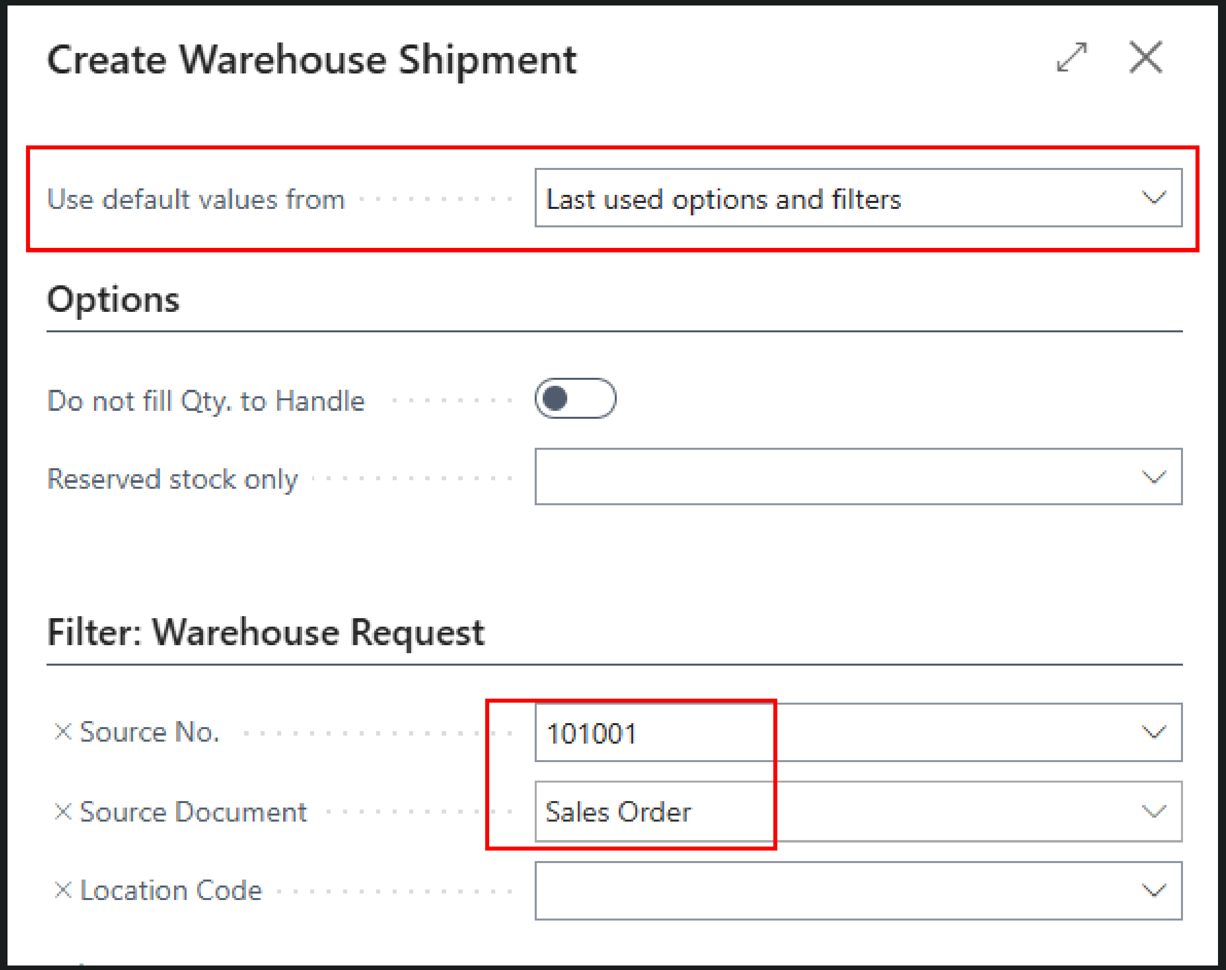
✅ Improve User Adoption.

A positive onboarding experience encourages users to embrace the new system.

#11 Enable Save Values

What is Save Values?

- BC remembers the last filter, option, or setting you used.
- Next time you open it, you start exactly where you left off.



Why use it?

✅ Better user experience.

Users like it when the system remembers their previous choices.

✅ Time-Saving.

Users don’t have to reconfigure the parameters.

How to use it?

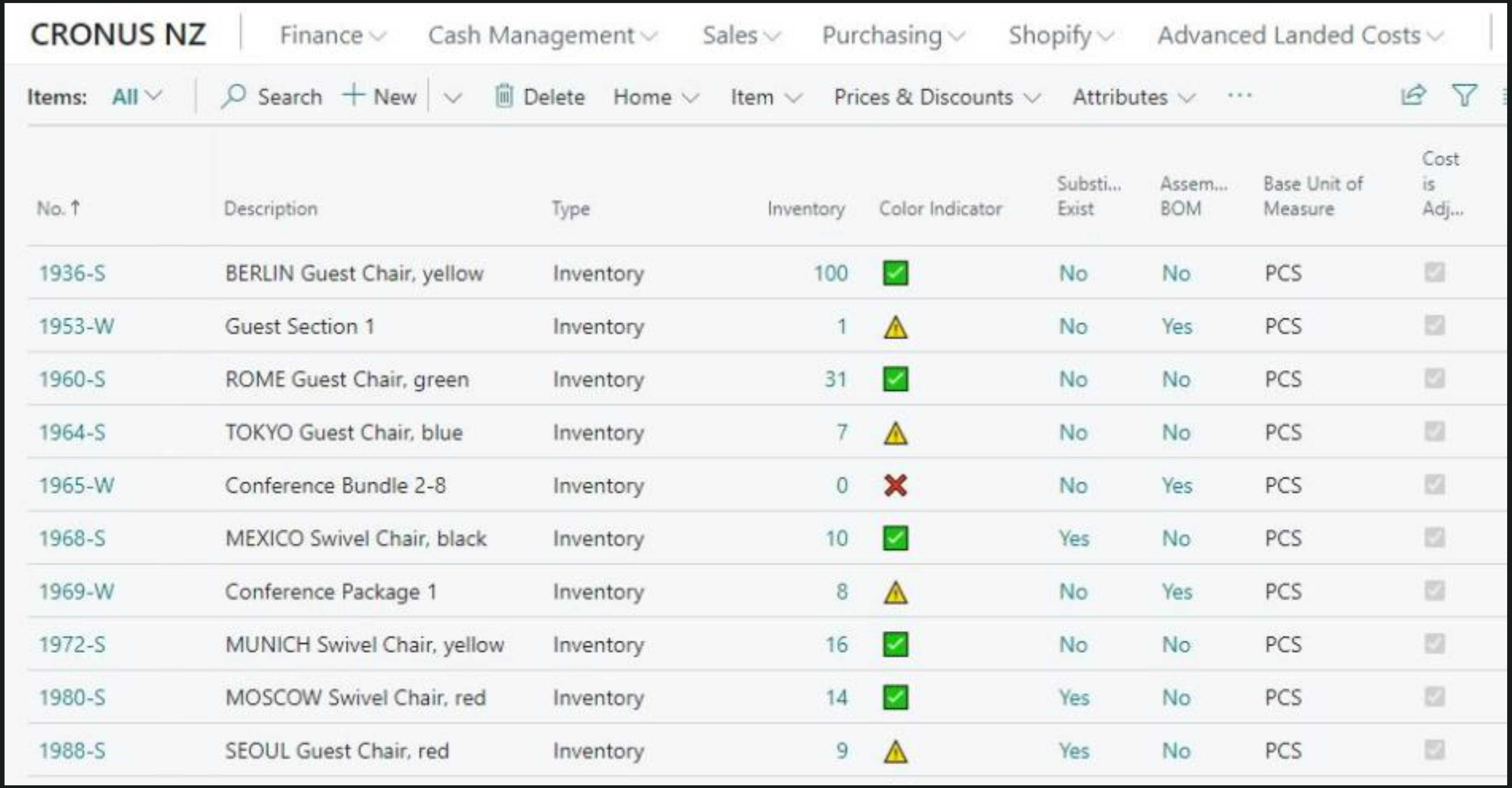
Create+Warehouse+Shipment.dal

```
{
    }

    requestpage
    {
        SaveValues = true;

        layout
        {
            area(content)
            {
                group(Options)
                {
                    Caption = 'Options';

                    field("Do Not Fill Qty. to Handle"; DoNotFillQtytoHandle)
                    {
                        Caption = 'Do not fill Qty. to Handle';
                        ToolTip = 'Specifies if the Quantity to Handle field in the warehouse
```

#12 Emoji as Visual Indicator

Emoji as Visual Indicator

- Use Emoji to represent statuses, warnings, or categories at a glance.



CRONUS NZ | Finance | Cash Management | Sales | Purchasing | Shopify | Advanced Landed Costs

Items: All | Search | + New | Delete | Home | Item | Prices & Discounts | Attributes | ···

No. ↑	Description	Type	Inventory	Color Indicator	Substi... Exist	Assem... BOM	Base Unit of Measure	Cost is Adj...
1936-S	BERLIN Guest Chair, yellow	Inventory	100	✅	No	No	PCS	☑
1953-W	Guest Section 1	Inventory	1	⚠️	No	Yes	PCS	☑
1960-S	ROME Guest Chair, green	Inventory	31	✅	No	No	PCS	☑
1964-S	TOKYO Guest Chair, blue	Inventory	7	⚠️	No	No	PCS	☑
1965-W	Conference Bundle 2-8	Inventory	0	❌	No	Yes	PCS	☑
1968-S	MEXICO Swivel Chair, black	Inventory	10	✅	Yes	No	PCS	☑
1969-W	Conference Package 1	Inventory	8	⚠️	No	Yes	PCS	☑
1972-S	MUNICH Swivel Chair, yellow	Inventory	16	✅	No	No	PCS	☑
1980-S	MOSCOW Swivel Chair, red	Inventory	14	✅	Yes	No	PCS	☑
1988-S	SEOUL Guest Chair, red	Inventory	9	⚠️	Yes	No	PCS	☑

Why use it?

- ✅ Quick visual cues without reading text.
- ✅ Helps users spot issues or changes instantly.
- ✅ Makes your UI more engaging and user-friendly.

How to use it?

```
trigger OnAfterGetRecord()
begin
    if Rec.Inventory >= 10 then
        InventoryIndicator := '✅'
    else
        if Rec.Inventory > 0 then
            InventoryIndicator := '⚠️'
        else
            InventoryIndicator := '❌';
end;
```

#13 DTAP Sandbox Naming Convention

What is DTAP Naming Convention?

- Development, Testing, Acceptance, Production.

- Name the sandboxes for their purpose:
    - Sandbox-Dev
    - Sandbox-Test
    - Sandbox-Acc
    - Sandbox-UAT
    - Sandbox-Demo

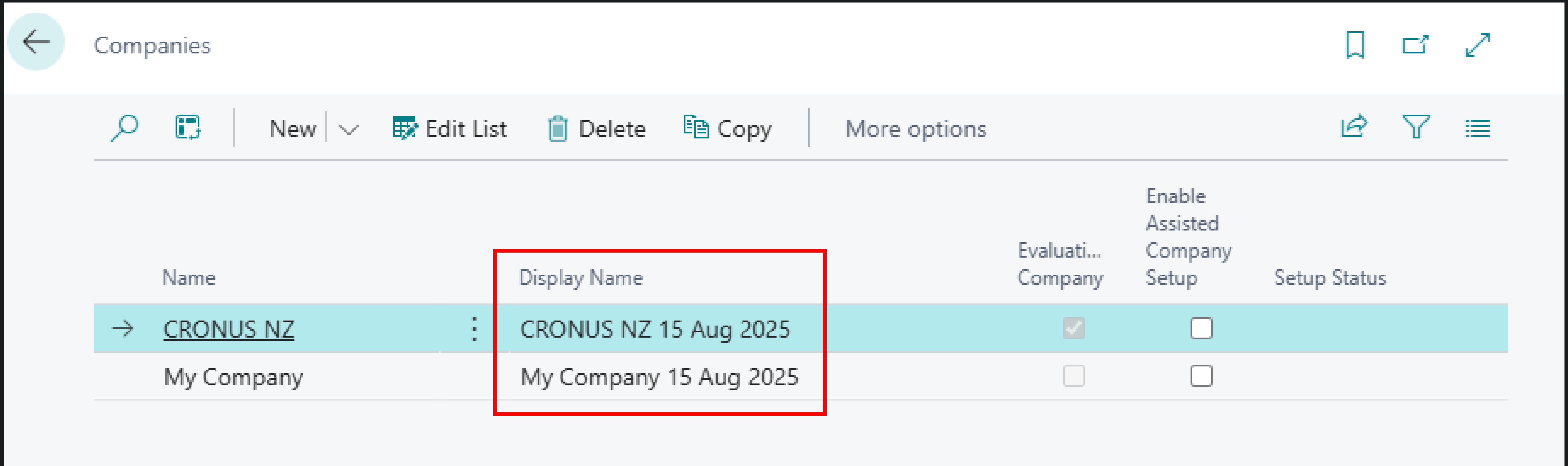
- No date and time in the Sandbox name
    - Sandbox-August2025
    - Sandbox-Dev1Aug25

Why use it?

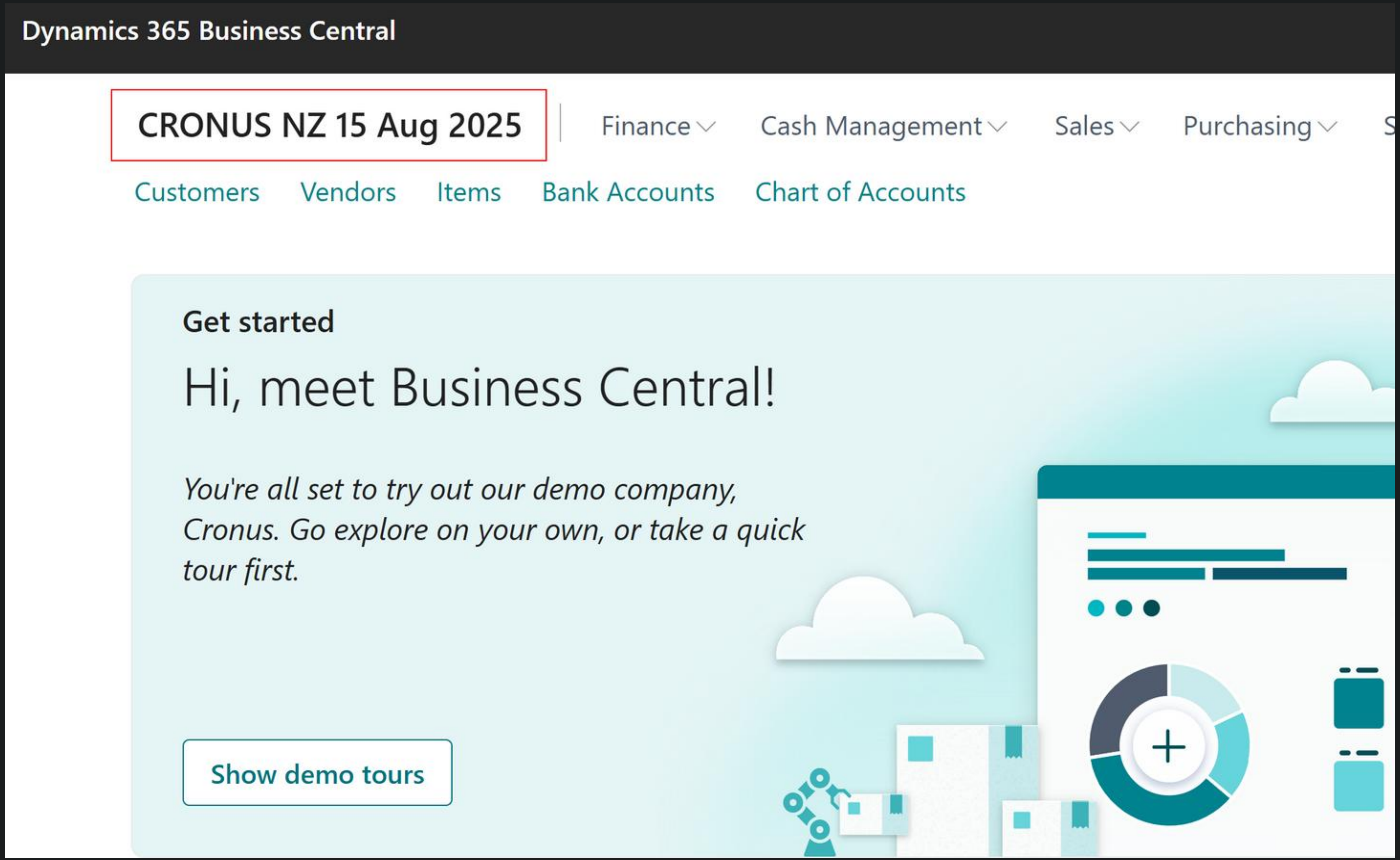
- ✅ Consistency.
- ✅ Easy access. Users don’t need to change their bookmarks.
- ✅ Work with integration.

#14 Sandbox Creation Date on the Company Display Name

How to use it?


Companies
New
Edit List
Delete
Copy
More options
Name
Display Name
Evaluati... Company
Enable Assisted Company Setup
Setup Status
CRONUS NZ
CRONUS NZ 15 Aug 2025
My Company
My Company 15 Aug 2025

Creation Date in Company Display Name



#15 Automate Sandbox Initialisation

Sandbox Initialisation

- Add creation date to Company Display Name.
- Set the company badge.
- Delete email accounts on masters.
- Replace or remove the setup.

Why automate it?

✅ Save manual time.

✅ Consistent and ready for testing.

How to automate it?

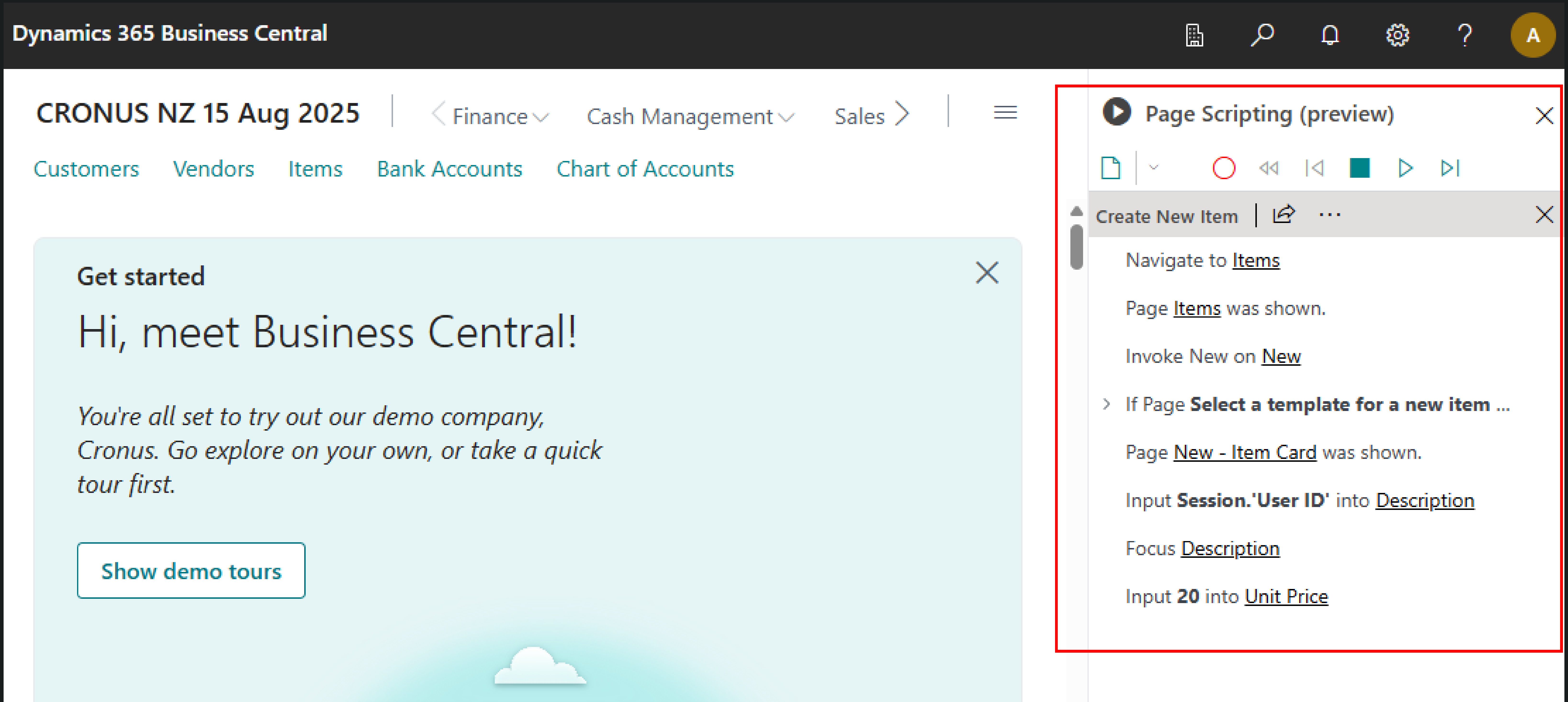
```
[EventSubscriber(ObjectType::Codeunit, Codeunit::"Environment Cleanup", OnClearCompanyConfig, '', false, false)]
0 references
local procedure EnvironmentCleanup_OnClearCompanyConfig(SourceEnv: Enum "Environment Type"; DestinationEnv: Enum "Environment Type"; CompanyName: Text)
begin
    if CanRunInitialisation(SourceEnv, DestinationEnv) then
        InitialiseSandbox(CompanyName);
end;

1 reference
local procedure CanRunInitialisation(var SourceEnv: Enum "Environment Type"; var DestinationEnv: Enum "Environment Type"): Boolean
begin
    if SourceEnv <> SourceEnv::Production then
        exit(false);

    if DestinationEnv <> DestinationEnv::Sandbox then
        exit(False);

    exit(true);
end;
```

#16 Use Page Scripting

What is Page Scripting?

- Record your BC interaction and automate repetitive clicks.
- Primary use is for automated testing.
- Record once. Run multiple times.

Page Scripting



Page Scripting File

Create New Item.yml U ×

C: > Personal > BC-Resources > Teddy Presentation > Create New Item.yml

```yaml
name: Create New Item
description: Test recording
start:
  profile: BUSINESS MANAGER
steps:
  - type: navigate
    target:
      - page: Business Manager Role Center
      - action: Items
    description: Navigate to <caption>Items</caption>
  - type: page-shown
    source:
      page: Item List
    modal: false
    runtimeId: b5dx
    description: Page <caption>Items</caption> was shown.
  - type: invoke
    target:
      - page: Item List
        runtimeRef: b5dx
      - action: Control_New
    invokeType: New
    description: Invoke New on <caption>New</caption>
  - type: scope
    condition:
      type: page-shown
      source:
        page: Select Item Templ. List
      runtimeId: b5ro
```

Convert to Documentation

Read this yml file and convert it into step to step documentation on how to create a new item

✓ Read Create New Item.yml

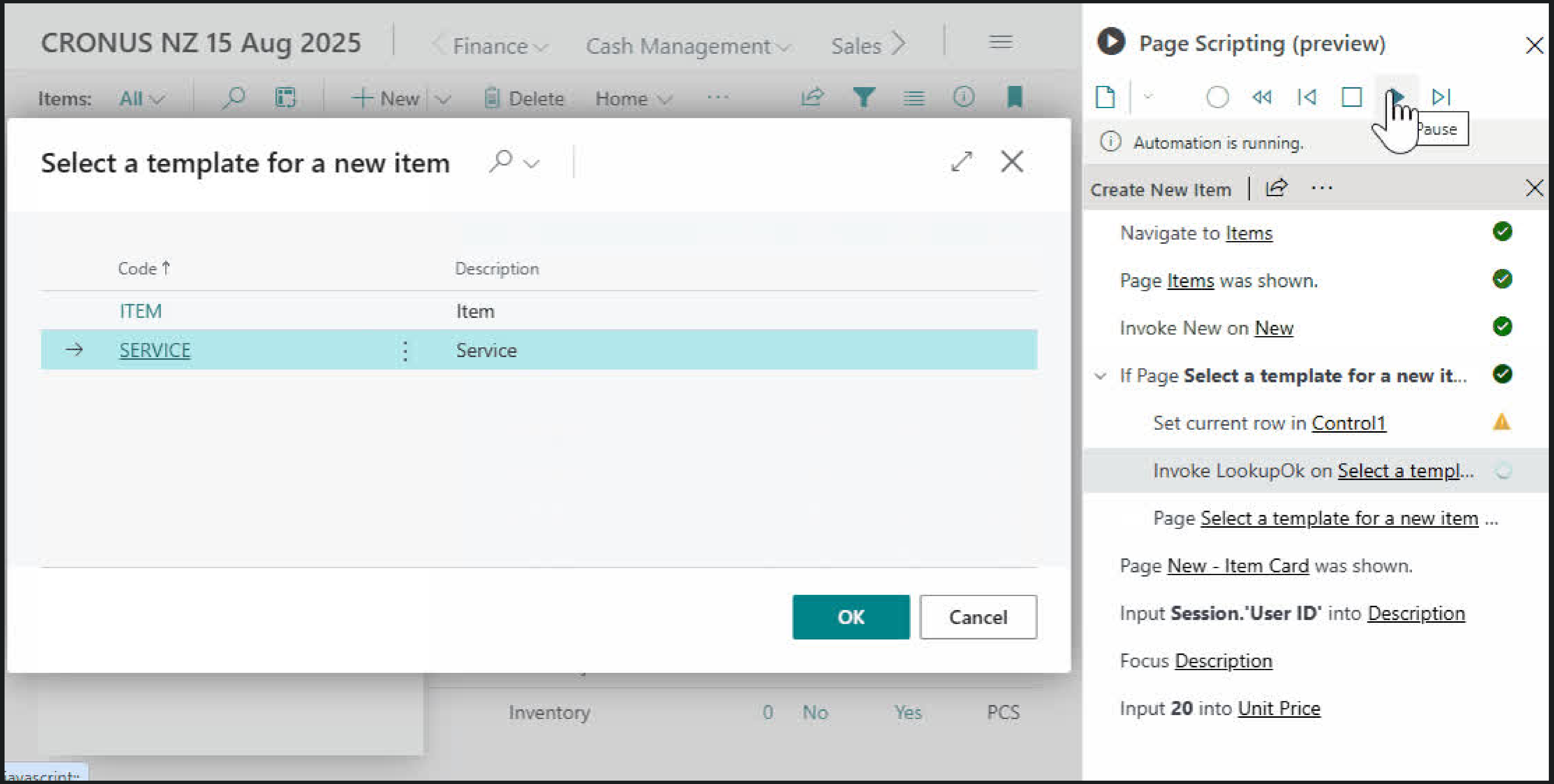
Here is a step-by-step guide on how to create a new item, based on your YAML file:

How to Create a New Item

1. **Start as Business Manager**
   - Make sure you are logged in with the "BUSINESS MANAGER" profile.
2. **Navigate to Items**
   - From the Business Manager Role Center, select the "Items" action.
3. **Wait for the Item List Page**
   - The "Items" list page should appear.
4. **Create a New Item**
   - Click the "New" button to start creating a new item.
5. **(If prompted) Select an Item Template**
   - If the "Select Item Template List" page appears:
     - Select the second row in the template list.
     - Click "OK" to confirm your template selection.
     - The template selection page will close.
6. **Wait for the Item Card Page**
   - The "New - Item Card" page will open.
7. **Enter Item Details**
   - In the "Description" field, enter your User ID (or the value shown in your session).
   - In the "Unit Price" field, enter "20".
8. **Focus on Description (Optional)**
   - Click into the "Description" field to focus it, if needed.

Let me know if you want this formatted differently or need it as a document!

Record Instructional Video



#17 Add Successful Feedback

Without Successful Feedback

Did it work? Should I click again? Did I break something?

Why success feedback matter?

- ✅ Reassures users that their action was completed.
- ✅ Acknowledges and builds their trust in the system.
- ✅ Keeps users moving without hesitation.

How to write Success Messages?

- Tell the users what they need to know. Provide enough context to be helpful.
- Tell it in easy-to-understand language. Avoid technical jargon.
- Summarise instead of repeating.
- Guide users to next steps whenever possible.
- Use GuiAllowed for Schedule or Background Tasks.

Example Success Messages

❌ Sales Order has been posted.
✅ Sales Order 101234 has been posted successfully.

❌ Purchase Order has been released.
✅ Purchase Order 101234 has been released and emailed to the vendor.

❌ File has been imported.
✅ 250 customer records were imported successfully.

❌ Item was added.
✅ New item “Coffee Maker” (Item No. 70021) was added.

❌ Purchase Order has been approved.
✅ All selected purchase orders have been approved. 5 records processed.

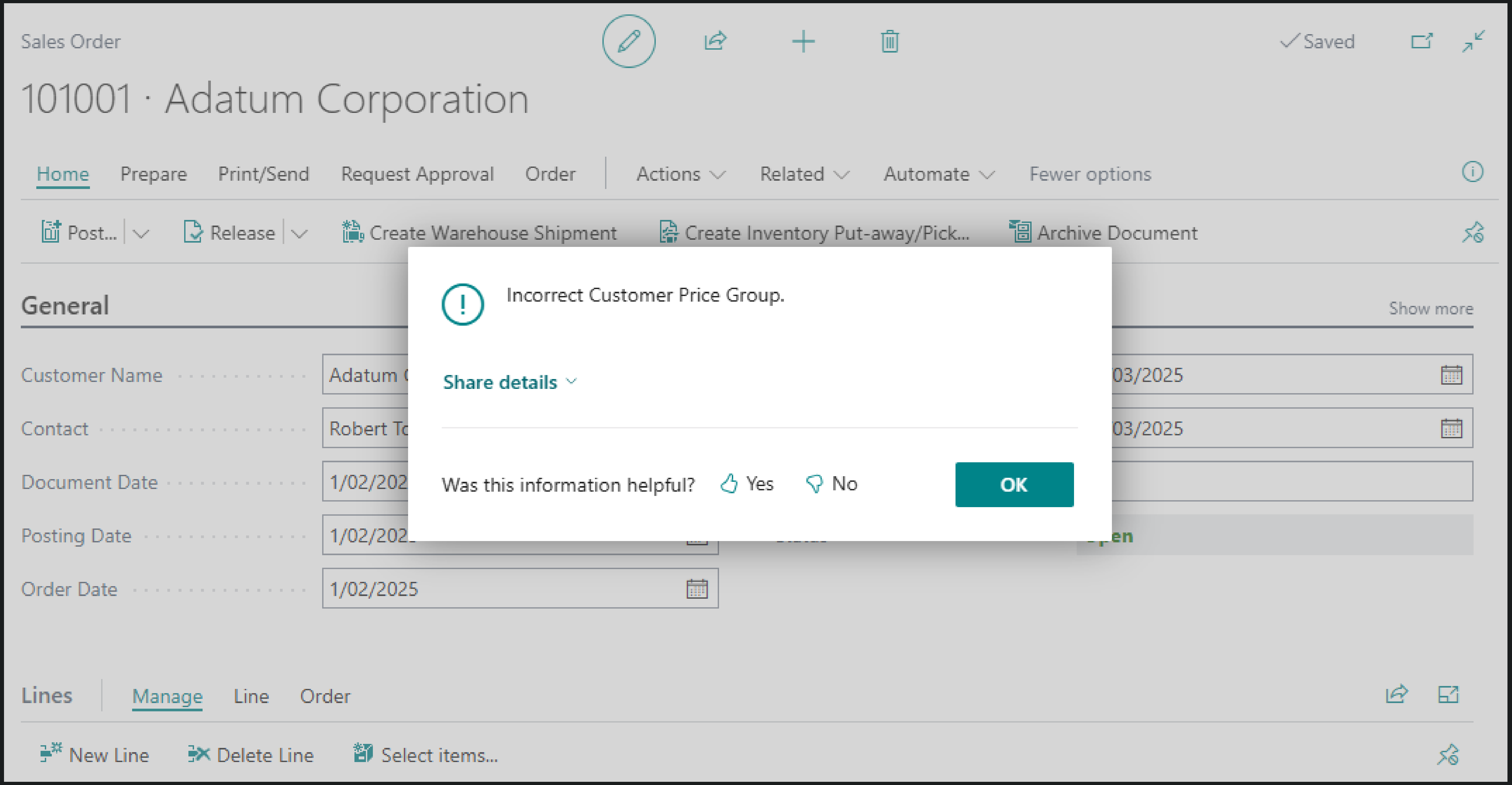
❌ Sales Invoice prints.
✅ Sales Invoice INV-123456 was printed successfully.

#18 Context on Error Message

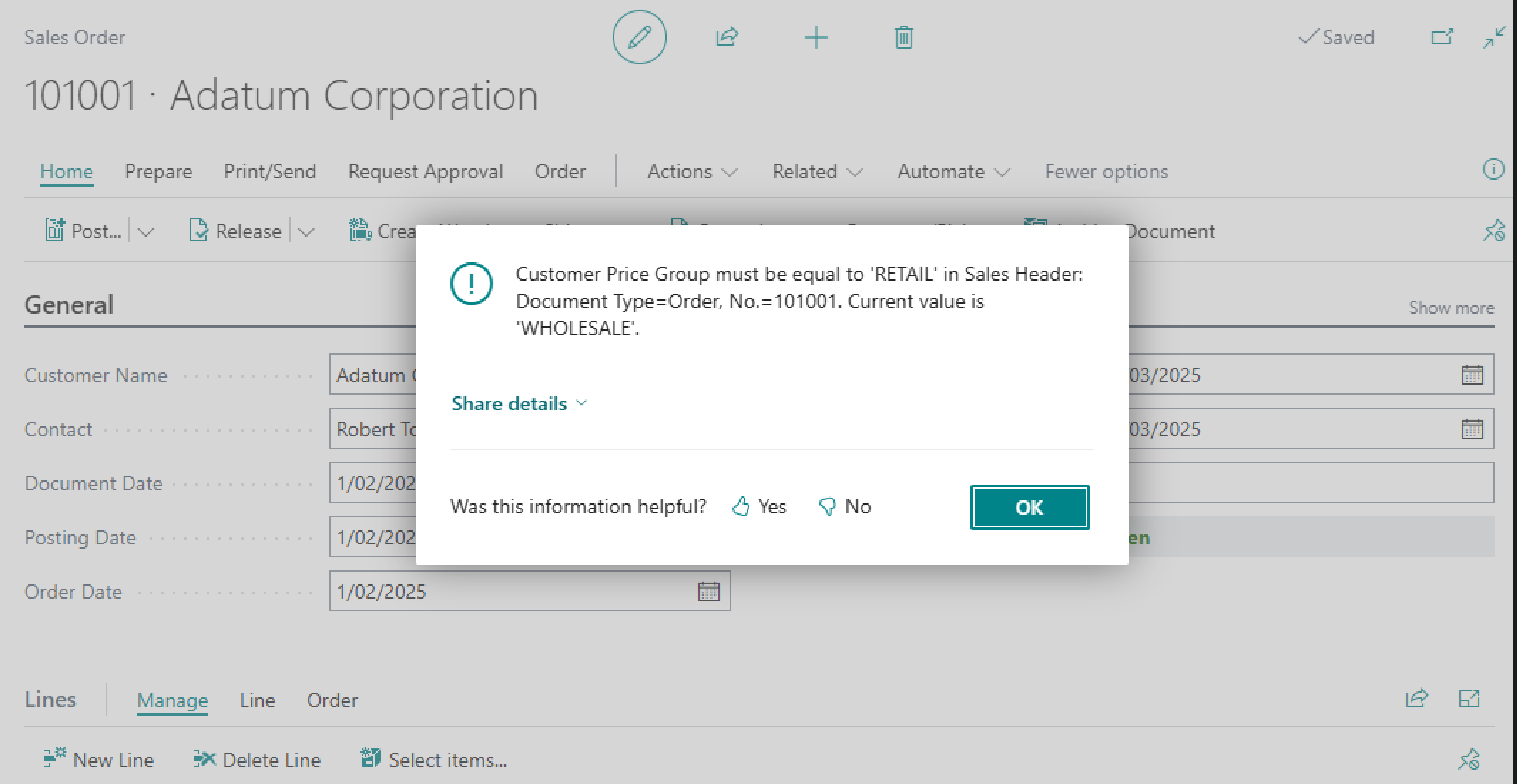
Importance of Context

When it comes to Error message, context is important.

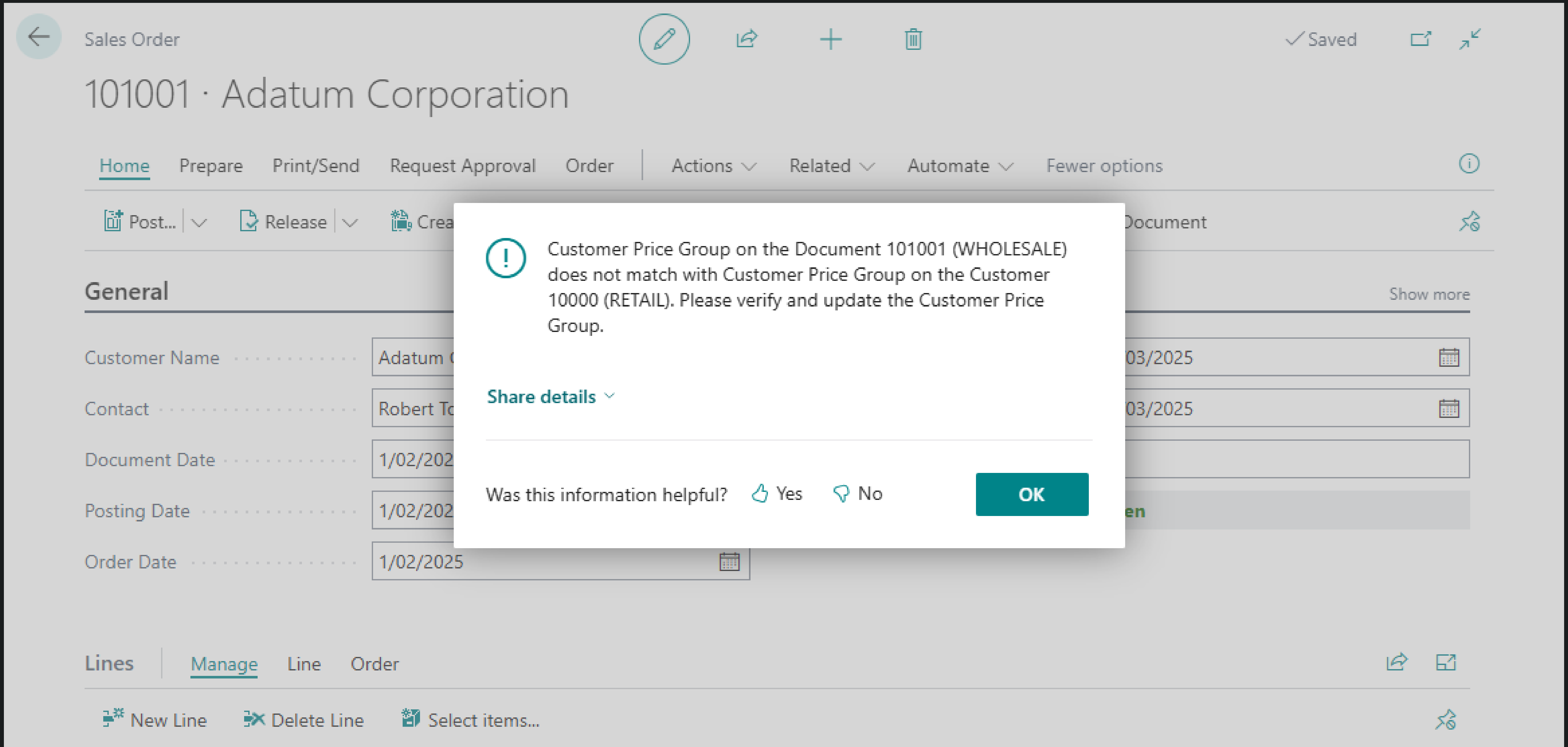
Error without Context



Error with Context



Error with Why Context



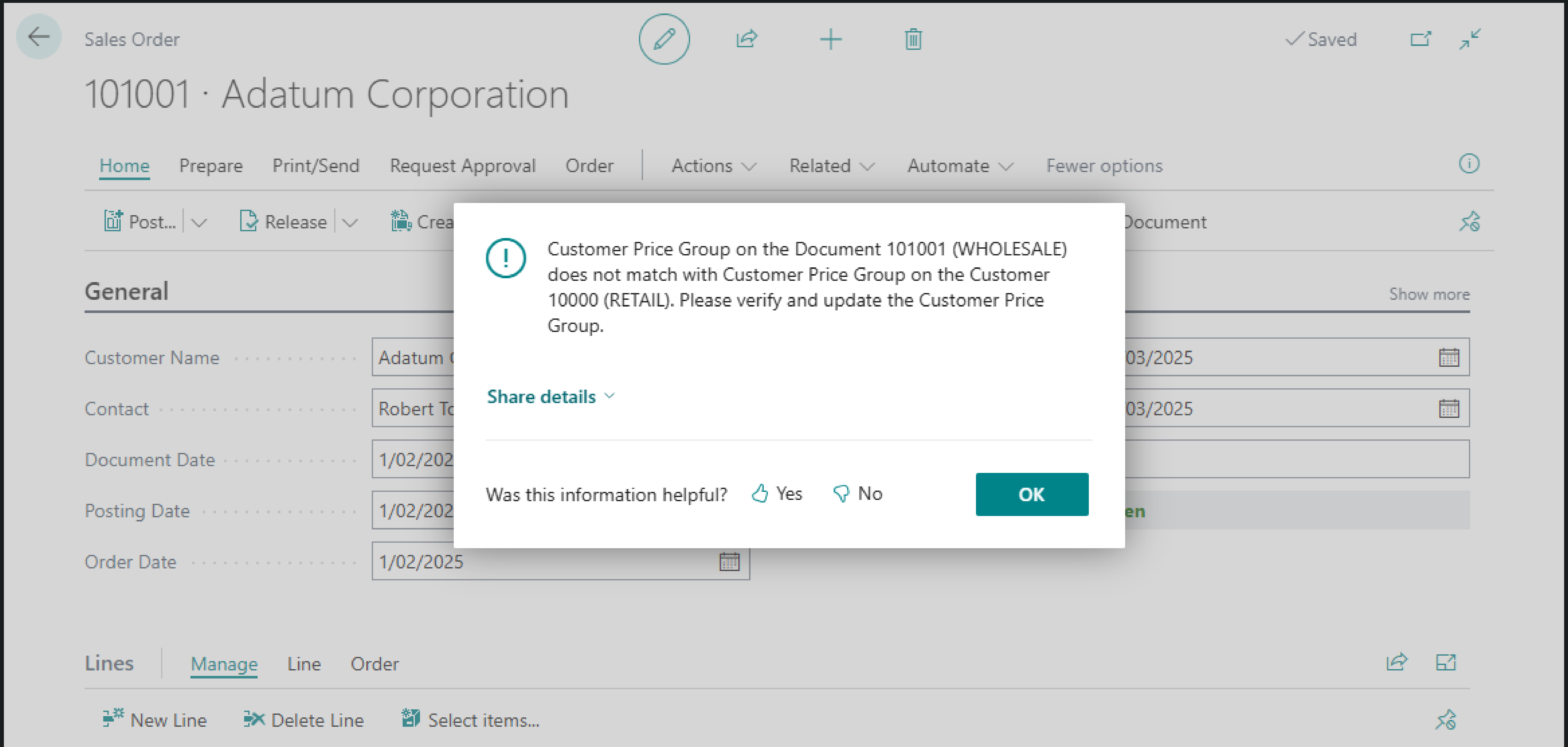
Why context is important?

✅ Avoid confusing Error Messages.

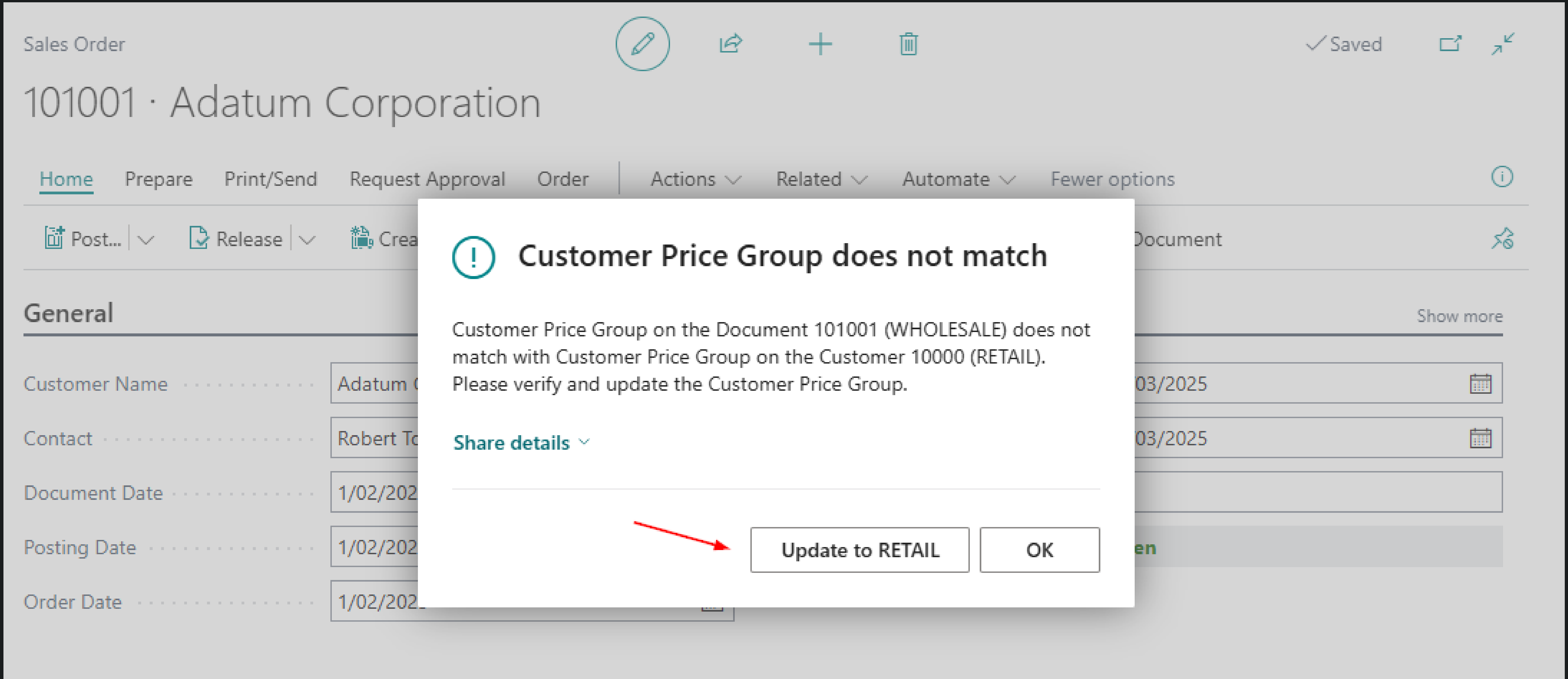
✅ Provide clear guidance to make an informed decision.

#19 Add Actionable Error

Error Message with Context



Actionable Error



Why use Actionable Error?

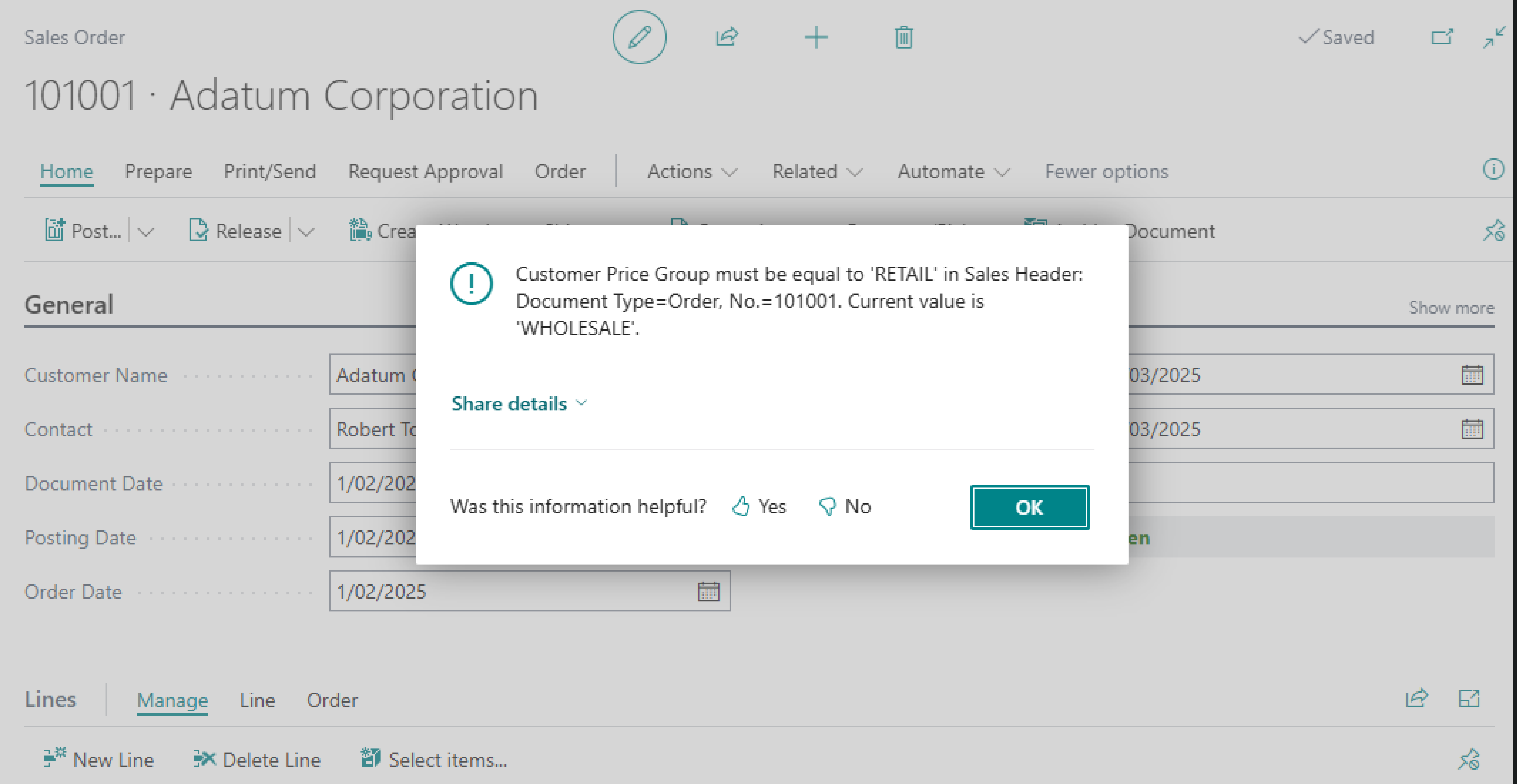
- ✅ Give guidance to the user.
- ✅ Save user time.

How to use it?

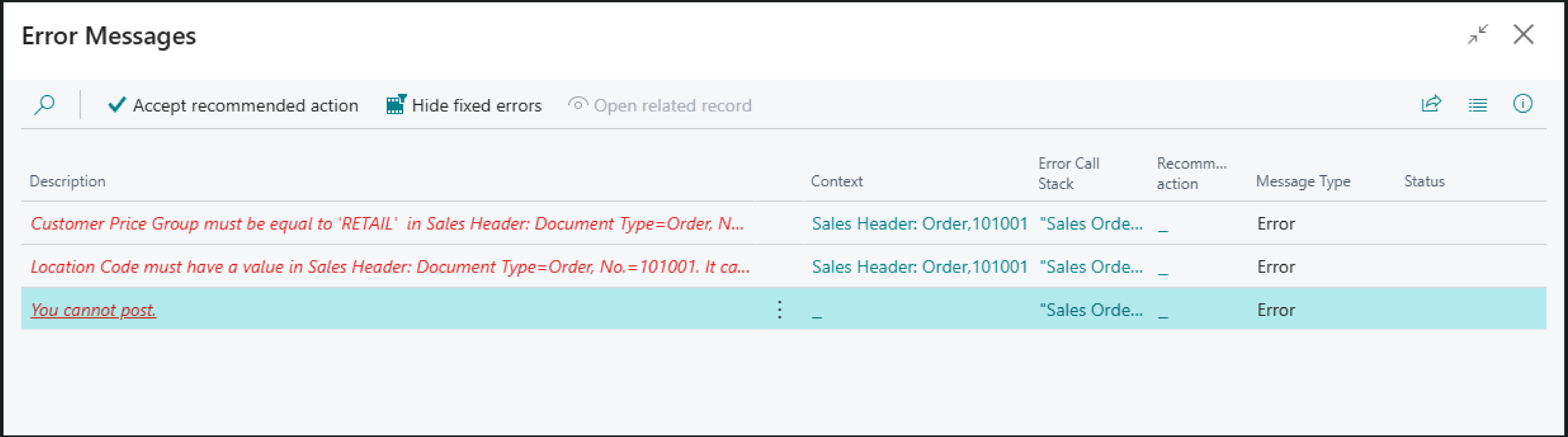
```
1 reference
local procedure SampleActionableError()
var
    Customer: Record Customer;
    CustomerPriceGroupTitleMsg: Label 'Customer Price Group does not match';
    CustomerPriceGroupDetailedMsg: Label 'You are not allowed to choose different Customer Price Group tl
    UpdateActionMsg: Label 'Update to %1', Comment = '%1=Customer."Customer Price Group"';
    MyErrorInfo: ErrorInfo;
begin
    Customer.Get(Rec."Sell-to Customer No.");
    if Rec."Customer Price Group" <> Customer."Customer Price Group" then begin
        MyErrorInfo.Title(CustomerPriceGroupTitleMsg);
        MyErrorInfo.Message(StrSubstNo(CustomerPriceGroupErr, Rec."Customer Price Group", Customer."Custo
        MyErrorInfo.DetailedMessage(CustomerPriceGroupDetailedMsg);
        MyErrorInfo.RecordId := Rec.RecordId;

        MyErrorInfo.AddAction(StrSubstNo(UpdateActionMsg, Customer."Customer Price Group"),
            Codeunit::"Update Customer", 'UpdateCustomerPriceGroupOnDocument');

        Error(MyErrorInfo);
    end;
end;
```

#20 Use Collectable Error

First Error Stop



Collectible Errors

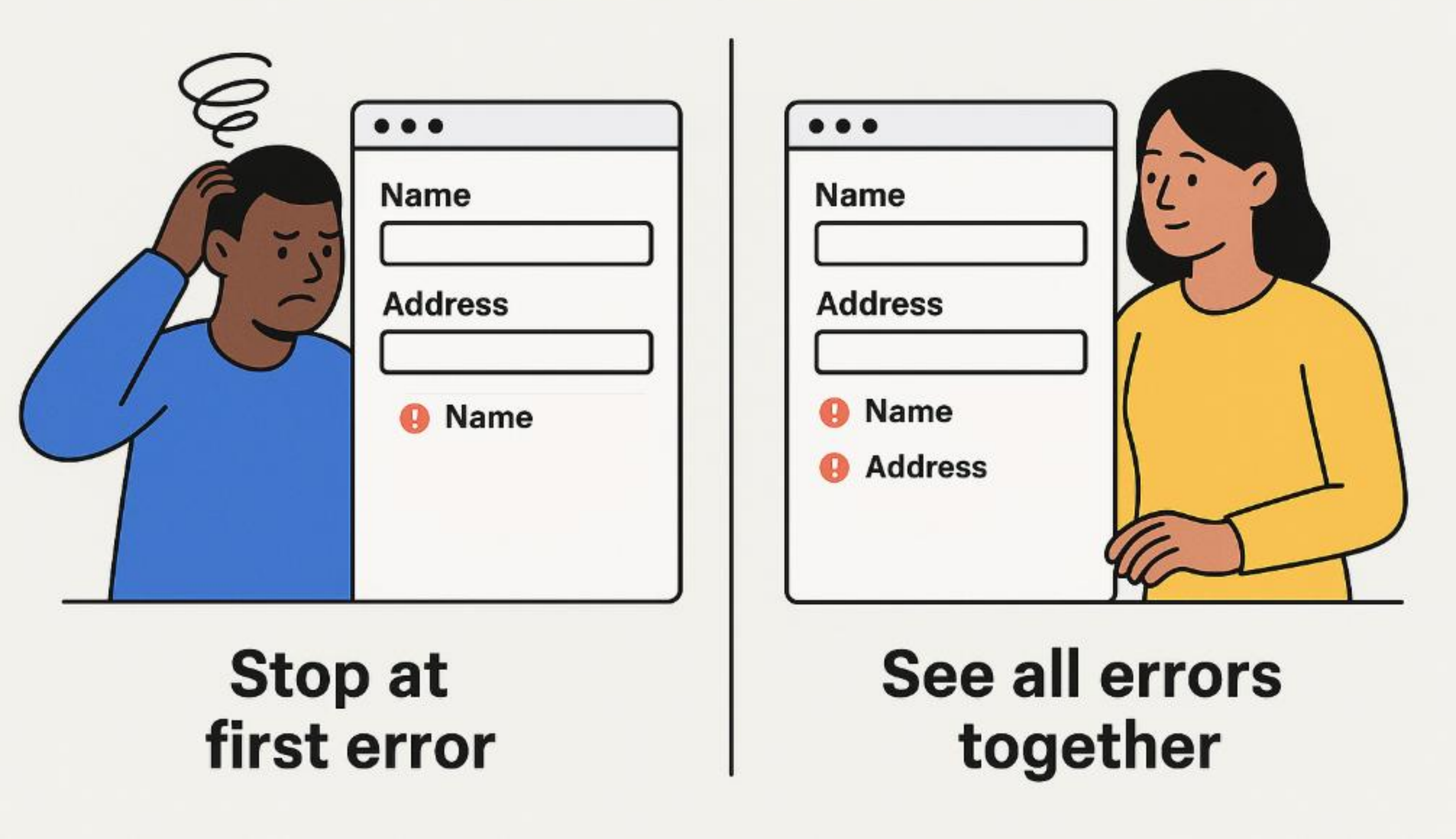

Error Messages

Description	Context	Error Call Stack	Recomm... action	Message Type	Status
Customer Price Group must be equal to 'RETAIL' in Sales Header: Document Type=Order, N...	Sales Header: Order,101001	"Sales Orde...	_	Error	
Location Code must have a value in Sales Header: Document Type=Order, No.=101001. It ca...	Sales Header: Order,101001	"Sales Orde...	_	Error	
You cannot post.	_	"Sales Orde...	_	Error	

Why use Collectable Errors?

✅ Save user time. Users can fix multiple problems in one go.

✅ Better user experience. Users feel supported instead of being blocked repeatedly.



How to use it?

```
[ErrorBehavior(ErrorBehavior::Collect)]
1 reference
local procedure ValidateOrderWithCollectableError()
var
    TempErrorMessage: Record "Error Message" temporary;
    MyErrorInfo: ErrorInfo;
begin
    ValidateOrder();

    if not HasCollectedErrors then
        exit;

    //Show the errors on page
    foreach MyErrorInfo in system.GetCollectedErrors() do begin
        TempErrorMessage.ID := TempErrorMessage.ID + 1;
        TempErrorMessage.Message := CopyStr(MyErrorInfo.Message, 1, 2048);
        TempErrorMessage."Context Record ID" := MyErrorInfo.RecordId;
        TempErrorMessage.SetErrorCallStack(MyErrorInfo.Callstack());
        TempErrorMessage.Insert(false);
    end;

    ClearCollectedErrors();

    Page.RunModal(Page::"Error Messages", TempErrorMessage);
    Error('');
end;

1 reference
local procedure ValidateOrder() //throw multiple errors here
var
    Customer: Record Customer;
begin
    Customer.Get(Rec."Sell-to Customer No.");
    Rec.TestField("Customer Price Group", Customer."Customer Price Group", ErrorInfo.Create());
    Rec.TestField("Location Code", ErrorInfo.Create());
    Error(ErrorInfo.Create('You cannot post.', true));
end;
```

Thank you!