ciellos
EVERYTHING DYNAMICS

BC Add-in for Excel – Overview Services

Contents

Introduction

In this article, we would like to discuss the nuances and pitfalls when working with "Excel Add-In". We will also look behind the scenes and consider the mechanism of work and whether it is possible to make changes in the code to meet the requirements of the customer.

One way to use the Add-in for Excel is to load data as part of a migration project from another system or reimplementation. In this case, the most effective method is to split the data into several files and download them simultaneously.

We hope this will be useful for both the business analysts and the developers.

How to install and configure for Business Central On-Premises

The installation procedure is described in detail on the Microsoft Learn, so here we will only provide some links:

Setting up the Business Central Add-in for Excel in Business Central On-premises (Opens in new window or tab)

Get the Business Central Add-in for Excel (Opens in new window or tab)

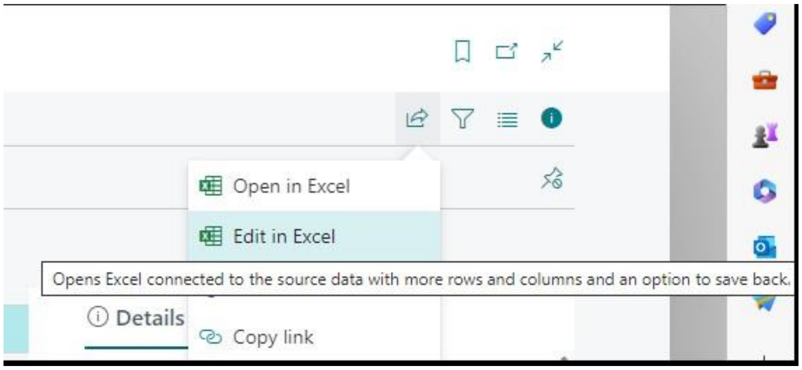
Viewing and Editing in Excel From Business Central (Opens in new window or tab)

Notes

- The add-in works with Excel for the web (online) from any device as long as as use a supported browser. It also works with the Excel app for Windows (PC); **but not for macOS**. Unfortunately, we didn't have a chance to check if "Add-Ins Excel" really doesn't work on Mac.
- If you configured the "Excel Add-In" to work with Business Central on premises, don't forget modify configuration as described below: Modify the Excel Add-in Configuration to Support July 2022 Update (Opens in new window or tab)

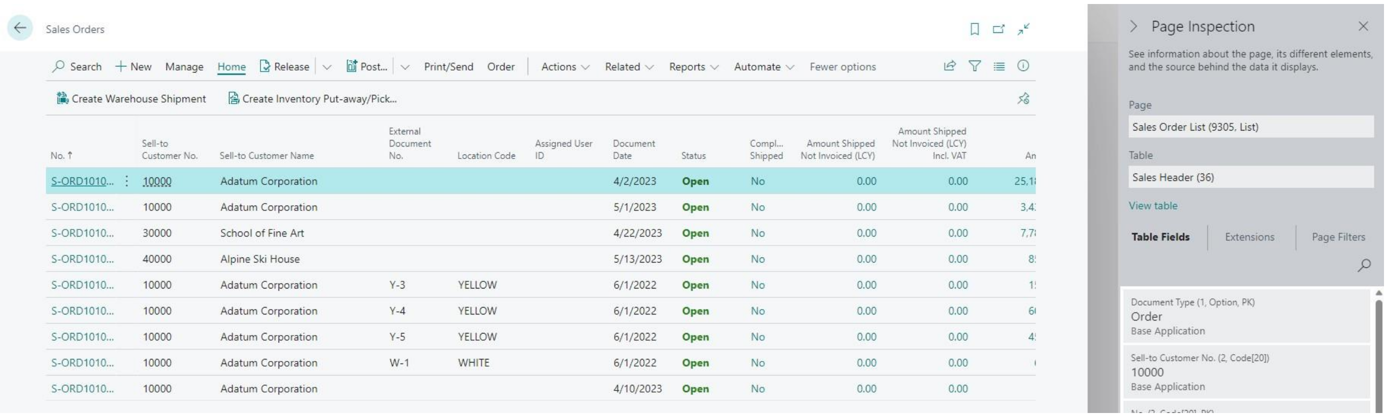
"Add-Ins Excel". How it works.

Every time when you click on "**Edit in Excel**"



Opens Excel connected to the source data with more rows and columns and an option to save back.

In this example, I pressed "Edit in Excel" on the page "Sales Order List (9305, List)"



No. ↑	Sell-to Customer No.	Sell-to Customer Name	External Document No.	Location Code	Assigned User ID	Document Date	Status	Compl... Shipped	Amount Shipped Not Invoiced (LCY)	Amount Shipped Not Invoiced (LCY) Incl. VAT	An
S-ORD1010...	10000	Adatum Corporation				4/2/2023	Open	No	0.00	0.00	25,1
S-ORD1010...	10000	Adatum Corporation				5/1/2023	Open	No	0.00	0.00	3,4
S-ORD1010...	30000	School of Fine Art				4/22/2023	Open	No	0.00	0.00	7,7
S-ORD1010...	40000	Alpine Ski House				5/13/2023	Open	No	0.00	0.00	8
S-ORD1010...	10000	Adatum Corporation	Y-3	YELLOW		6/1/2022	Open	No	0.00	0.00	1
S-ORD1010...	10000	Adatum Corporation	Y-4	YELLOW		6/1/2022	Open	No	0.00	0.00	6
S-ORD1010...	10000	Adatum Corporation	Y-5	YELLOW		6/1/2022	Open	No	0.00	0.00	4
S-ORD1010...	10000	Adatum Corporation	W-1	WHITE		6/1/2022	Open	No	0.00	0.00	
S-ORD1010...	10000	Adatum Corporation				4/10/2023	Open	No	0.00	0.00	

Page Inspection

See information about the page, its different elements, and the source behind the data it displays.

Page: Sales Order List (9305, List)

Table: Sales Header (36)

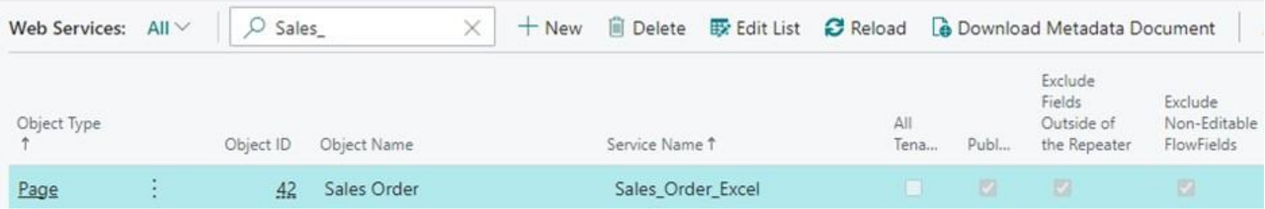
View table

Table Fields | Extensions | Page Filters

Document Type (1, Option, PK)
Order
Base Application

Sell-to Customer No. (2, Code[20])
10000
Base Application

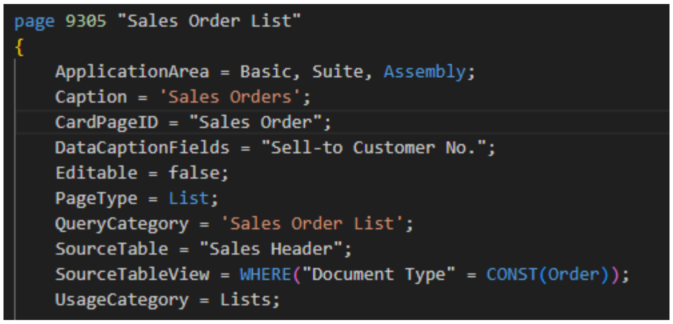
The system creates **web service** if it doesn't exist.



Object Type ↑		Object ID	Object Name	Service Name ↑	All Tena...	Publ...	Exclude Fields Outside of the Repeater	Exclude Non-Editable FlowFields
Page	⋮	42	Sales Order	Sales_Order_Excel	☐	☑	☑	☑

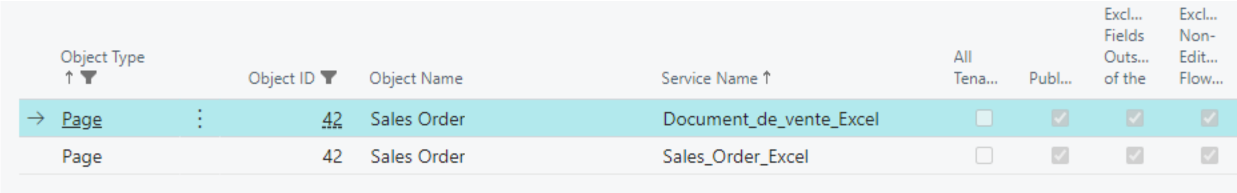
But the page used in web service is "Sales Order (42, Document)" and not "Sales Order List (9305, List)"! The same thing for Purchase Order List (9307, List) -> Purchase Order (50, Document) Item List (31, List) - >Item Card (30, Card) etc.

Why so? For these pages (there are a number of tables for which this works differently, but I'll cover that later) Web service is created based on page from **CardPageID**.



```
page 9305 "Sales Order List"
{
    ApplicationArea = Basic, Suite, Assembly;
    Caption = 'Sales Orders';
    CardPageID = "Sales Order";
    DataCaptionFields = "Sell-to Customer No.";
    Editable = false;
    PageType = List;
    QueryCategory = 'Sales Order List';
    SourceTable = "Sales Header";
    SourceTableView = WHERE("Document Type" = CONST(Order));
    UsageCategory = Lists;
```

Let's now take a look at the name of the service - Sales_Order_Excel **If we change language in our parameters on French (for example), as result we will have two web services Sales_Order_Excel Document_de_vente_Excel The reason for this is that the name of the service is generated from the Page Caption and not the Page Name. This must be taken into account when configuring "Excel Add-In"**



Object Type	Object ID	Object Name	Service Name	All Tena...	Publ...	Excl... Fields Outs... of the	Excl... Non-Edit... Flow...
Page	42	Sales Order	Document_de_vente_Excel	☐	☑	☑	☑
Page	42	Sales Order	Sales_Order_Excel	☐	☑	☑	☑

Let's go ahead and look at some of the issues I've encountered while working with "Add-Ins Excel"

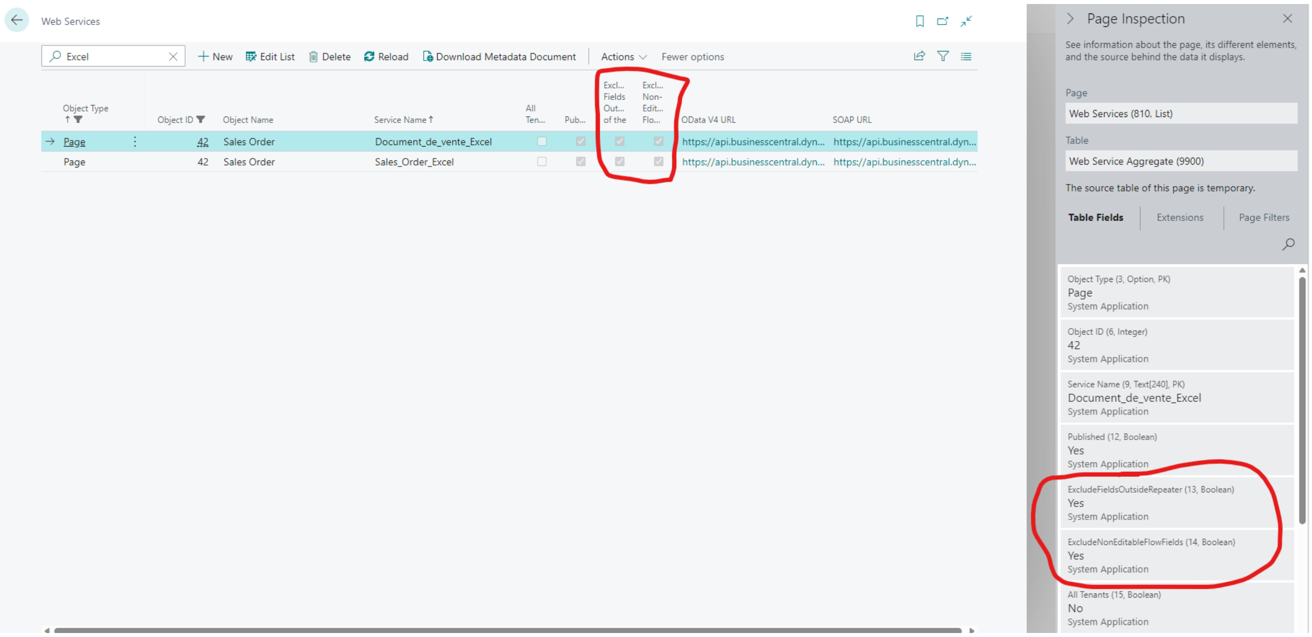
Troubleshooting

Sometimes, users run into problems with the "Add-In Excel". This section gives some tips for how to unblock users in certain circumstances.

Issue	Solution or workaround	Comments
The add-in doesn't start	Check whether the add-in is deployed centrally. Or, check whether the user is blocked from installing it locally.	The admin can configure Office so that users can't acquire add-ins. In those cases, the admin must deploy the add-in centrally. For more information, see Deploy add-ins in the admin center (Opens in new window or tab)
Data doesn't load into Excel	Test the connection by opening another list in Excel from Business Central. Or, open the workbook in Excel in a browser.	**If the user has specified a company name that contains special characters, the add-in can't connect.**
Data can't publish back to Business Central.	Test the connection by opening the workbook in Excel in a browser.	Sometimes an extension can block the publishing job. If the page is extended or customized, remove the extensions, and then try again.
The dates are wrong	Excel might show times and dates in a different format than Business Central. This condition doesn't make them wrong, and the data in Business Central won't get messed up.	

Issue:

For some list pages, editing multiple lines in Excel consistently causes errors. This condition can occur if OData calls include FlowFields and fields outside of the repeater control. **Solution or workaround:** On the **Web Services** page, select the **Exclude Non-Editable FlowFields** and **Exclude Fields Outside of the Repeater** check boxes for the published page. Selecting these check boxes excludes non-editable FlowFields and field from the eTag calculation. Comments: These check boxes are hidden by default. To show them on the **Web Services** page, use **Personalization**.



Web Services

Excel | + New | Edit List | Delete | Reload | Download Metadata Document | Actions | Fewer options

Object Type ↑	Object ID	Object Name	Service Name ↑	All Ten...	Pub...	Excl... Fields Out... of the	Excl... Non-Edit... Flo...	OData V4 URL	SOAP URL
Page	42	Sales Order	Document_de_vente_Excel	☐	☑	☑	☑	https://api.businesscentral.dyn...	https://api.businesscentral.dyn...
Page	42	Sales Order	Sales_Order_Excel	☐	☑	☑	☑	https://api.businesscentral.dyn...	https://api.businesscentral.dyn...

Page Inspection

See information about the page, its different elements, and the source behind the data it displays.

Page: Web Services (810, List)

Table: Web Service Aggregate (9900)

The source table of this page is temporary.

Table Fields | Extensions | Page Filters

Object Type (3, Option, PK)
Page
System Application

Object ID (6, Integer)
42
System Application

Service Name (9, Text[240], PK)
Document_de_vente_Excel
System Application

Published (12, Boolean)
Yes
System Application

ExcludeFieldsOutsideRepeater (13, Boolean)
Yes
System Application

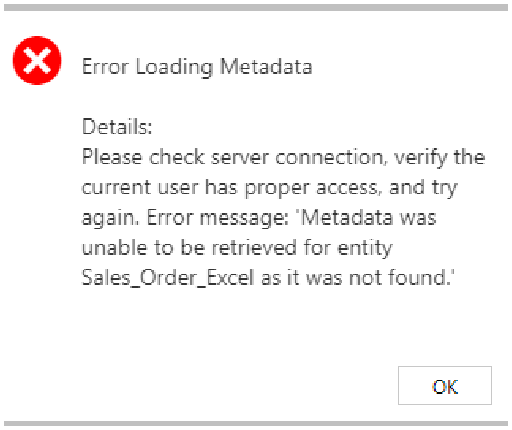
ExcludeNonEditableFlowFields (14, Boolean)
Yes
System Application

All Tenants (15, Boolean)
No
System Application

By the way, Microsoft plans to change this functionality based on information from [Exclude FlowFields and fields outside repeater from ETag (OData) (Opens in new window or tab)](#) So stay tuned for new releases.

Issue:

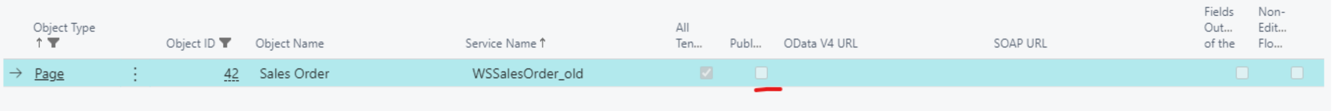
Metadata was enable to retrieved for entry Sales_Order_Excel as it was not found



Error Loading Metadata

Details:
Please check server connection, verify the current user has proper access, and try again. Error message: 'Metadata was unable to be retrieved for entity Sales_Order_Excel as it was not found.'

Solution or workaround: Check if web service with the same **[Object Type]** and **[Object ID]** exists but not published. Delete or publish this web service.

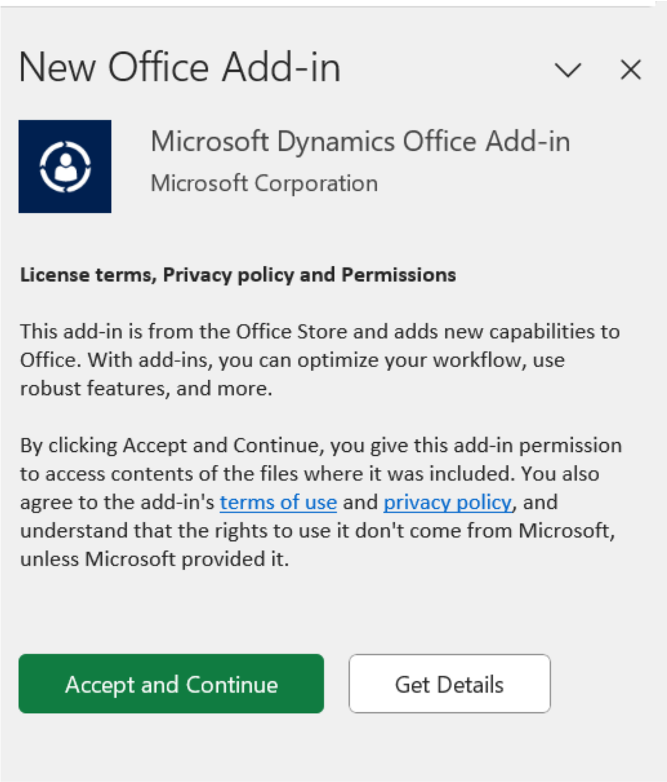


Object Type ↑	Object ID	Object Name	Service Name ↑	All Ten...	Publ...	OData V4 URL	SOAP URL	Fields Out... of the	Non-Edit... Flo...
Page	42	Sales Order	WSSalesOrder_old	☑	☐			☐	☐

Now let's take a look at what's going on in Excel

Excel

When we open the file generated by "Edit in Excel" and "Excel Add-In" is not installed, the system display page to install "Excel Add-In"



New Office Add-in

Microsoft Dynamics Office Add-in

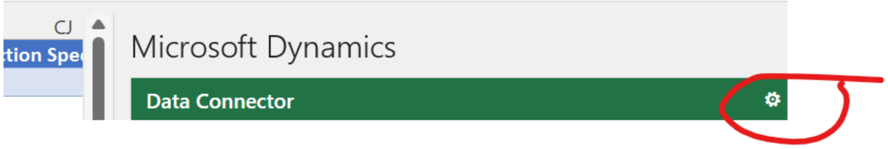
Microsoft Corporation

License terms, Privacy policy and Permissions

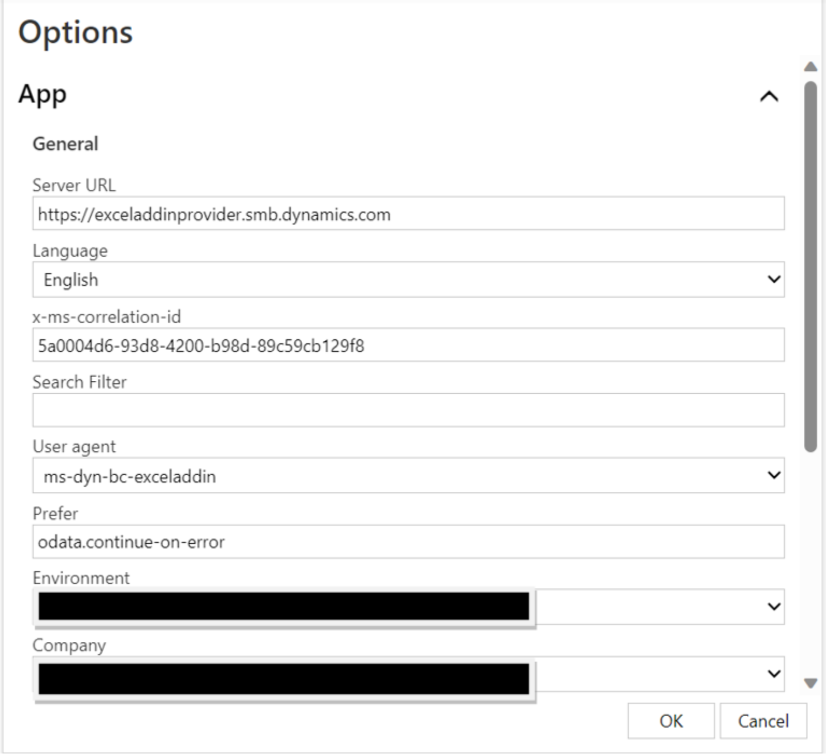
This add-in is from the Office Store and adds new capabilities to Office. With add-ins, you can optimize your workflow, use robust features, and more.

By clicking Accept and Continue, you give this add-in permission to access contents of the files where it was included. You also agree to the add-in's terms of use and privacy policy, and understand that the rights to use it don't come from Microsoft, unless Microsoft provided it.

Let's look at options



where we can change environment and company



Options

App

General

Server URL: https://exceladdinprovider.smb.dynamics.com

Language: English

x-ms-correlation-id: 5a0004d6-93d8-4200-b98d-89c59cb129f8

Search Filter:

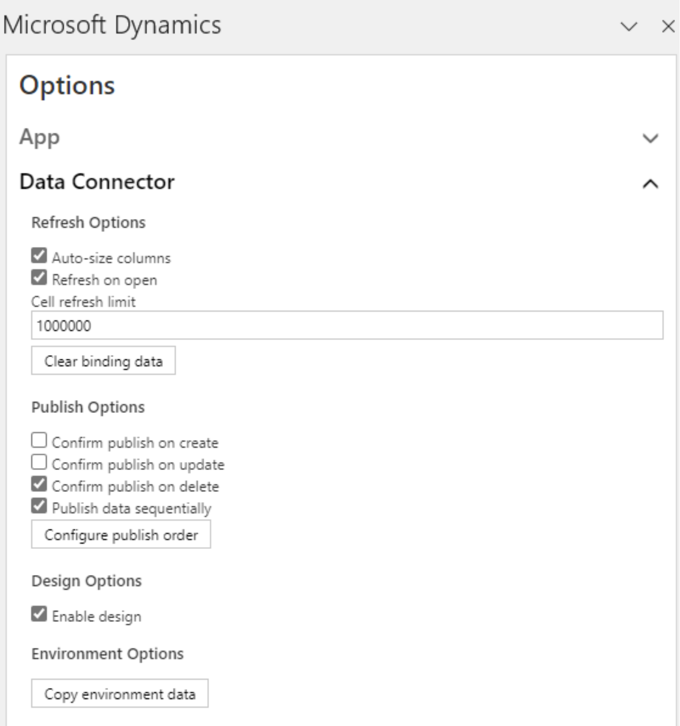
User agent: ms-dyn-bc-exceladdin

Prefer: odata.continue-on-error

Environment: [redacted]

Company: [redacted]

and change some parameters. The purpose of the parameters is intuitive from their names, so I will not dwell on them.



Microsoft Dynamics

Options

App

Data Connector

Refresh Options

☑ Auto-size columns
☑ Refresh on open

Cell refresh limit: 1000000

Publish Options

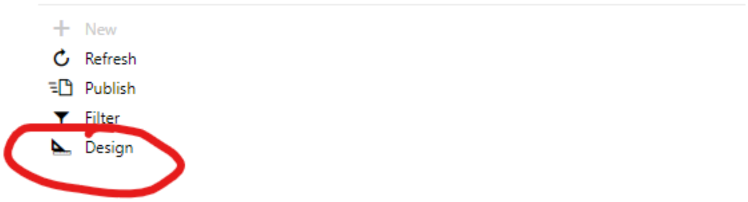
□ Confirm publish on create
□ Confirm publish on update
☑ Confirm publish on delete
☑ Publish data sequentially

Design Options

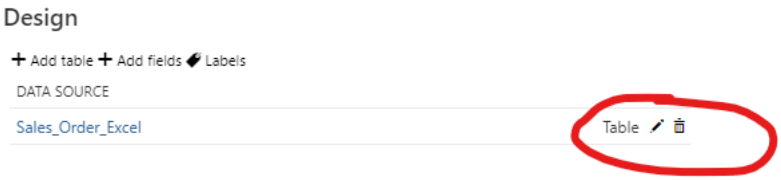
☑ Enable design

Environment Options

Let's click on **Design** button

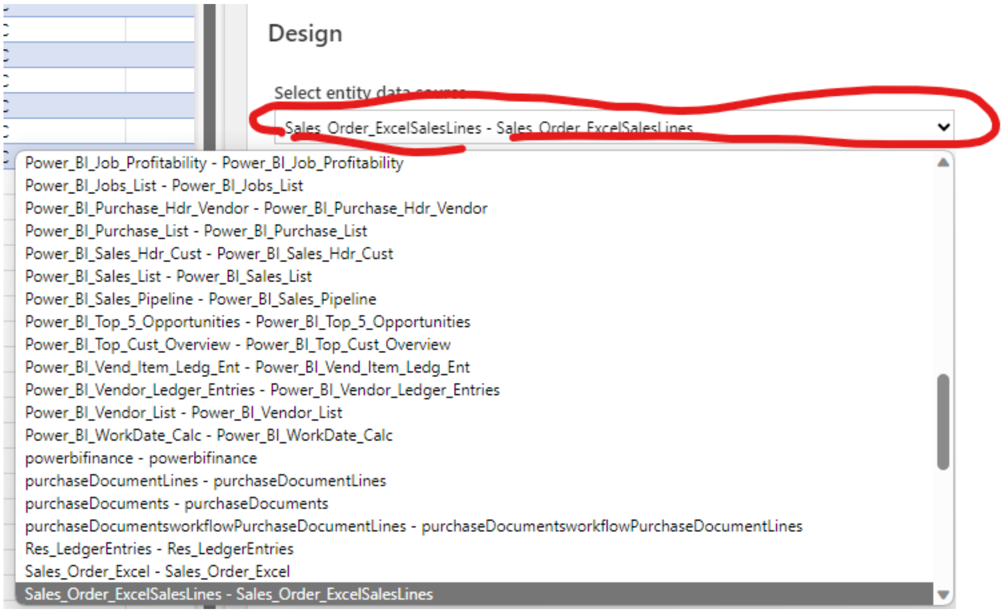


we can see the data source for this Excel file - Sales_Order_Excel



But what should we do if we want to add another table?

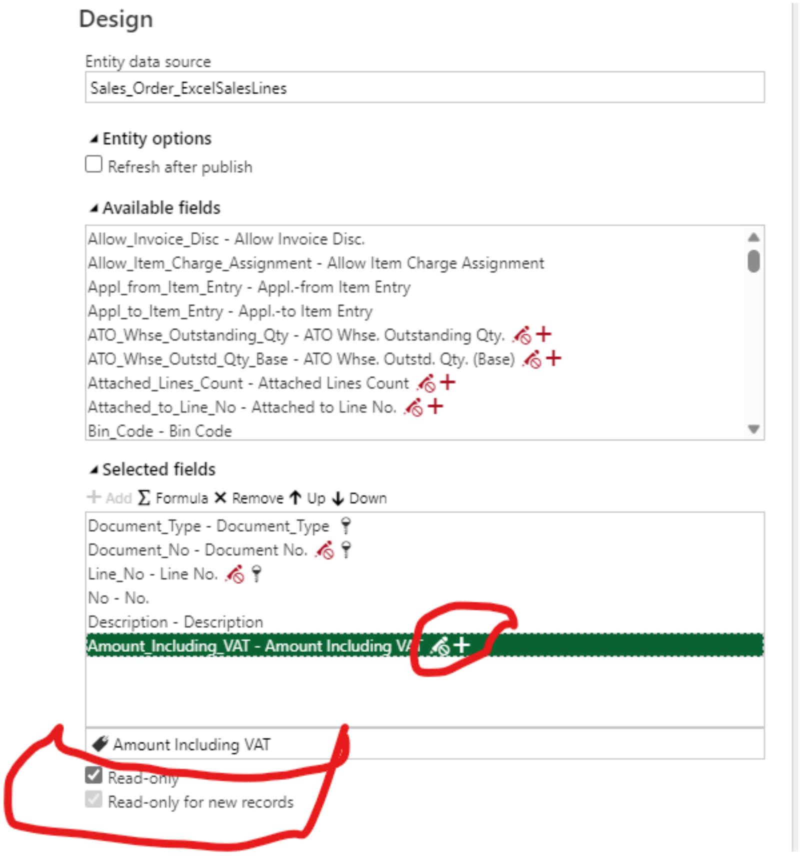
Just click + Add Table. Select from drop-down list the web service you want to use as data source. In our case I selected SalesLines.



Sales_Order_ExcelSalesLines - Sales_Order_ExcelSalesLines

Power_BI_Job_Profitability - Power_BI_Job_Profitability
Power_BI_Jobs_List - Power_BI_Jobs_List
Power_BI_Purchase_Hdr_Vendor - Power_BI_Purchase_Hdr_Vendor
Power_BI_Purchase_List - Power_BI_Purchase_List
Power_BI_Sales_Hdr_Cust - Power_BI_Sales_Hdr_Cust
Power_BI_Sales_List - Power_BI_Sales_List
Power_BI_Sales_Pipeline - Power_BI_Sales_Pipeline
Power_BI_Top_5_Opportunities - Power_BI_Top_5_Opportunities
Power_BI_Top_Cust_Overview - Power_BI_Top_Cust_Overview
Power_BI_Vend_Item_Ledg_Ent - Power_BI_Vend_Item_Ledg_Ent
Power_BI_Vendor_Ledger_Entries - Power_BI_Vendor_Ledger_Entries
Power_BI_Vendor_List - Power_BI_Vendor_List
Power_BI_WorkDate_Calc - Power_BI_WorkDate_Calc
powerbifinance - powerbifinance
purchaseDocumentLines - purchaseDocumentLines
purchaseDocuments - purchaseDocuments
purchaseDocumentsworkflowPurchaseDocumentLines - purchaseDocumentsworkflowPurchaseDocumentLines
Res_LedgerEntries - Res_LedgerEntries
Sales_Order_Excel - Sales_Order_Excel
Sales_Order_ExcelSalesLines - Sales_Order_ExcelSalesLines

Double-click on the fields you want to add from **Available Fields** to **Selected Fields**. ***Notice the special symbols next to the field name, which indicate that this is a field from the primary key, read-only, and read-only for new records***



Design

Entity data source

Sales_Order_ExcelSalesLines

Entity options

□ Refresh after publish

Available fields

Allow_Invoice_Disc - Allow Invoice Disc.
Allow_Item_Charge_Assignment - Allow Item Charge Assignment
Appl_from_Item_Entry - Appl.-from Item Entry
Appl_to_Item_Entry - Appl.-to Item Entry
ATO_Whse_Outstanding_Qty - ATO Whse. Outstanding Qty.
ATO_Whse_Outstd_Qty_Base - ATO Whse. Outstd. Qty. (Base)
Attached_Lines_Count - Attached Lines Count
Attached_to_Line_No - Attached to Line No.
Bin_Code - Bin Code

Selected fields

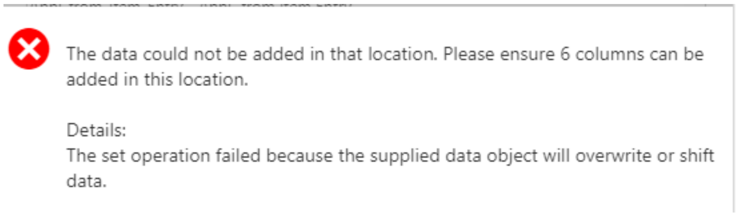
Document_Type - Document_Type
Document_No - Document No.
Line_No - Line No.
No - No.
Description - Description
Amount_Including_VAT - Amount Including VA

Amount Including VAT

☑ Read-only

☑ Read-only for new records

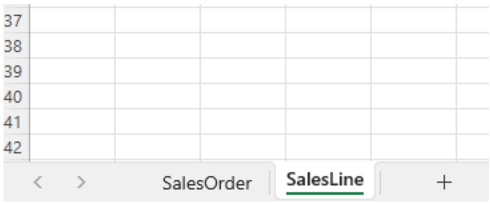
Push **Done** and we get error



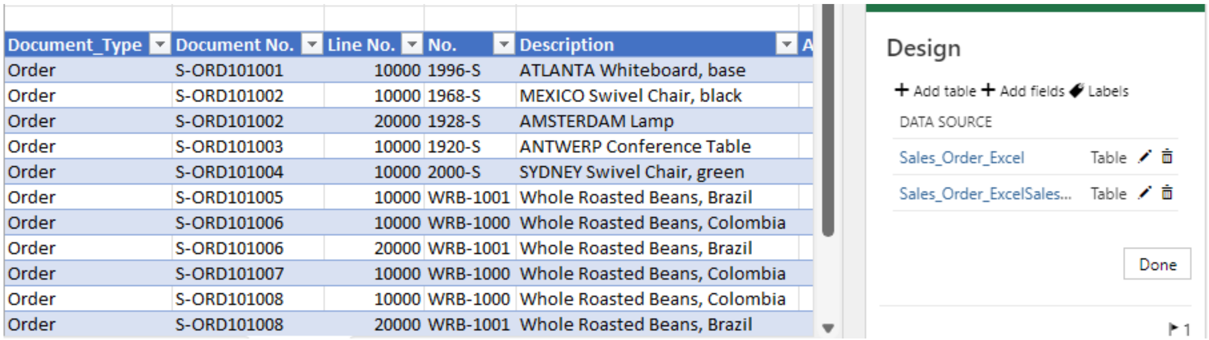
The data could not be added in that location. Please ensure 6 columns can be added in this location.

Details:
The set operation failed because the supplied data object will overwrite or shift data.

No problem, create new tab like this



and, Vois-Là, we have two tabs - Sales Orders and Sales Lines



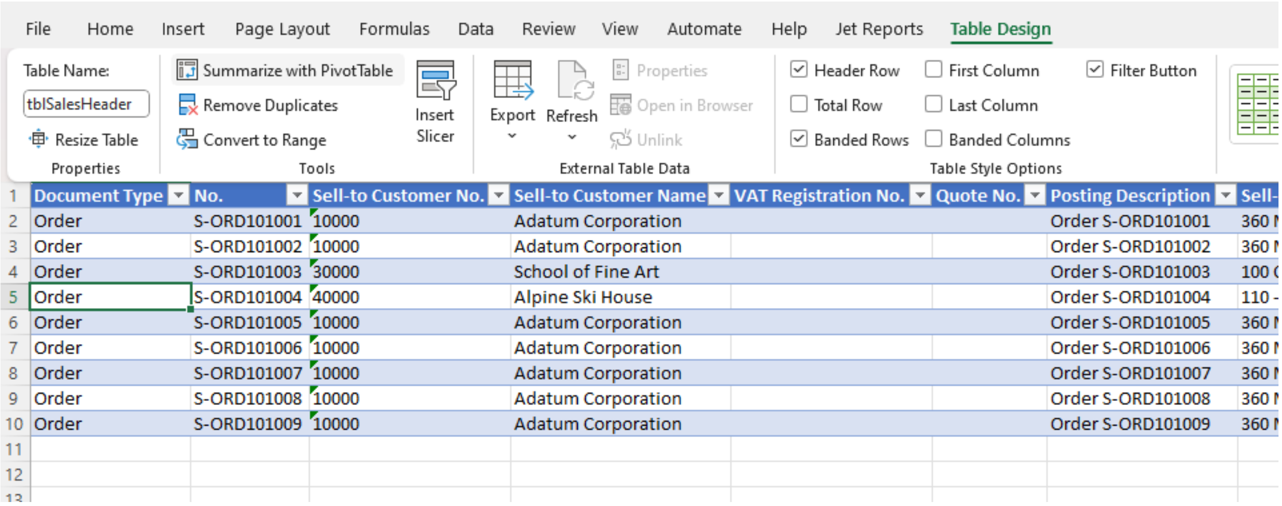
Document_Type	Document No.	Line No.	No.	Description	A
Order	S-ORD101001	10000	1996-S	ATLANTA Whiteboard, base	
Order	S-ORD101002	10000	1968-S	MEXICO Swivel Chair, black	
Order	S-ORD101002	20000	1928-S	AMSTERDAM Lamp	
Order	S-ORD101003	10000	1920-S	ANTWERP Conference Table	
Order	S-ORD101004	10000	2000-S	SYDNEY Swivel Chair, green	
Order	S-ORD101005	10000	WRB-1001	Whole Roasted Beans, Brazil	
Order	S-ORD101006	10000	WRB-1000	Whole Roasted Beans, Colombia	
Order	S-ORD101006	20000	WRB-1001	Whole Roasted Beans, Brazil	
Order	S-ORD101007	10000	WRB-1000	Whole Roasted Beans, Colombia	
Order	S-ORD101008	10000	WRB-1000	Whole Roasted Beans, Colombia	
Order	S-ORD101008	20000	WRB-1001	Whole Roasted Beans, Brazil	

Design

DATA SOURCE	
Sales_Order_Excel	Table
Sales_Order_ExcelSales...	Table

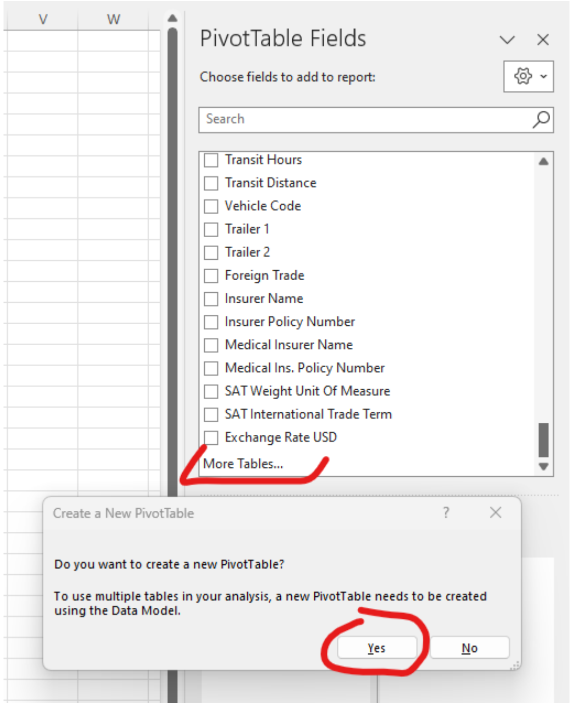
Note. It does not always make sense to allow editing certain fields or even tables as a whole. For example, "Sales Lines" - editing fields such as "Quantity" or "Price" separately from "Sales Header" can lead to a violation of the system business logic. At the same time, if these are fields like Comment or some instructions, then why not. So, a preliminary analysis is necessary in most cases.

What else can we do? Ok. Let's create PivotTable To do this, go to the tab "Table Design" and click on "Summarize with PivotTable"



1	Document Type	No.	Sell-to Customer No.	Sell-to Customer Name	VAT Registration No.	Quote No.	Posting Description	Sell-
2	Order	S-ORD101001	10000	Adatum Corporation			Order S-ORD101001	360 I
3	Order	S-ORD101002	10000	Adatum Corporation			Order S-ORD101002	360 I
4	Order	S-ORD101003	30000	School of Fine Art			Order S-ORD101003	100 (
5	Order	S-ORD101004	40000	Alpine Ski House			Order S-ORD101004	110 -
6	Order	S-ORD101005	10000	Adatum Corporation			Order S-ORD101005	360 I
7	Order	S-ORD101006	10000	Adatum Corporation			Order S-ORD101006	360 I
8	Order	S-ORD101007	10000	Adatum Corporation			Order S-ORD101007	360 I
9	Order	S-ORD101008	10000	Adatum Corporation			Order S-ORD101008	360 I
10	Order	S-ORD101009	10000	Adatum Corporation			Order S-ORD101009	360 I
11								
12								

In "PivotTable Fields" go to the bottom, click on "More Tables" and after that "Yes"



PivotTable Fields

Choose fields to add to report:

Search

□ Transit Hours
□ Transit Distance
□ Vehicle Code
□ Trailer 1
□ Trailer 2
□ Foreign Trade
□ Insurer Name
□ Insurer Policy Number
□ Medical Insurer Name
□ Medical Ins. Policy Number
□ SAT Weight Unit Of Measure
□ SAT International Trade Term
□ Exchange Rate USD

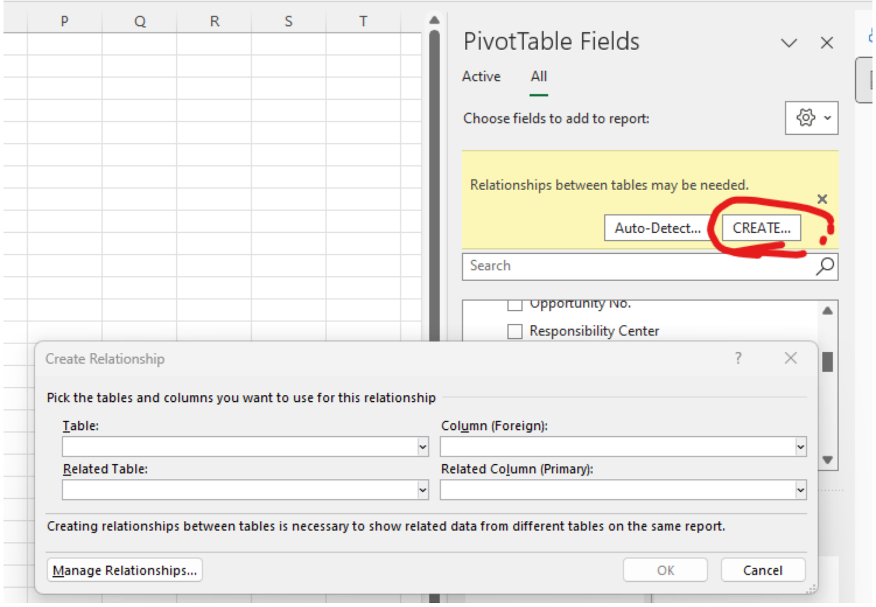
More Tables...

Create a New PivotTable

Do you want to create a new PivotTable?

To use multiple tables in your analysis, a new PivotTable needs to be created using the Data Model.

Then, create relationships between "tblSalesHeader" and "tblSalesLine" (We can use field "Document No." only, because there is filter "Document Type" = Order)



PivotTable Fields

Active All

Choose fields to add to report:

Relationships between tables may be needed.

Search

□ Opportunity No.
□ Responsibility Center

Create Relationship

Pick the tables and columns you want to use for this relationship

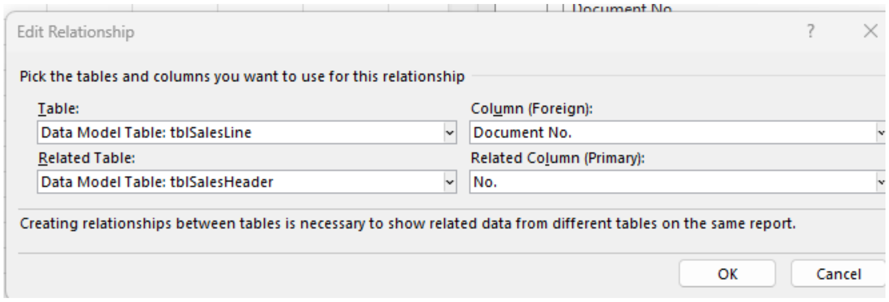
Table:

Column (Foreign):

Related Table:

Related Column (Primary):

Creating relationships between tables is necessary to show related data from different tables on the same report.

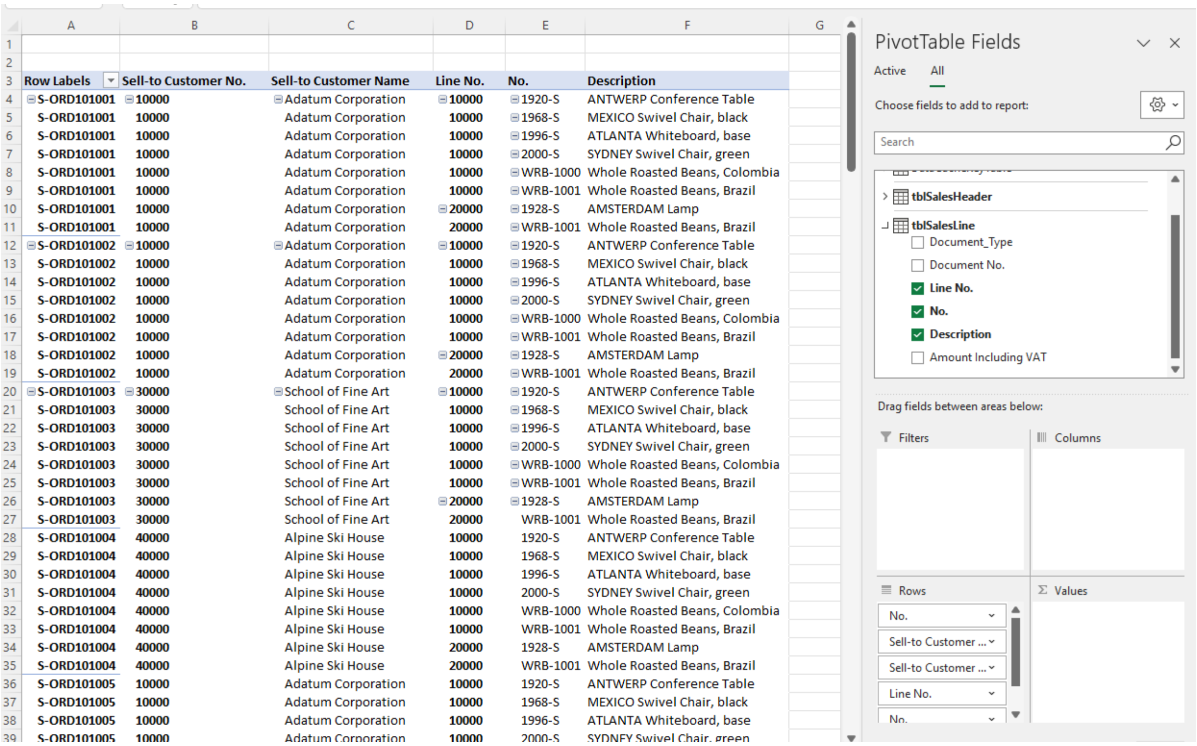


Pick the tables and columns you want to use for this relationship

Table:	Column (Foreign):
Data Model Table: tblSalesLine	Document No.
Related Table:	Related Column (Primary):
Data Model Table: tblSalesHeader	No.

Creating relationships between tables is necessary to show related data from different tables on the same report.

Finally, we have pivot table with fields from "Sales Header" and "Sales Line"



	A	B	C	D	E	F	G
1							
2							
3	Row Labels	Sell-to Customer No.	Sell-to Customer Name	Line No.	No.	Description	
4	S-ORD101001	10000	Adatum Corporation	10000	1920-S	ANTWERP Conference Table	
5	S-ORD101001	10000	Adatum Corporation	10000	1968-S	MEXICO Swivel Chair, black	
6	S-ORD101001	10000	Adatum Corporation	10000	1996-S	ATLANTA Whiteboard, base	
7	S-ORD101001	10000	Adatum Corporation	10000	2000-S	SYDNEY Swivel Chair, green	
8	S-ORD101001	10000	Adatum Corporation	10000	WRB-1000	Whole Roasted Beans, Colombia	
9	S-ORD101001	10000	Adatum Corporation	10000	WRB-1001	Whole Roasted Beans, Brazil	
10	S-ORD101001	10000	Adatum Corporation	20000	1928-S	AMSTERDAM Lamp	
11	S-ORD101001	10000	Adatum Corporation	20000	WRB-1001	Whole Roasted Beans, Brazil	
12	S-ORD101002	10000	Adatum Corporation	10000	1920-S	ANTWERP Conference Table	
13	S-ORD101002	10000	Adatum Corporation	10000	1968-S	MEXICO Swivel Chair, black	
14	S-ORD101002	10000	Adatum Corporation	10000	1996-S	ATLANTA Whiteboard, base	
15	S-ORD101002	10000	Adatum Corporation	10000	2000-S	SYDNEY Swivel Chair, green	
16	S-ORD101002	10000	Adatum Corporation	10000	WRB-1000	Whole Roasted Beans, Colombia	
17	S-ORD101002	10000	Adatum Corporation	10000	WRB-1001	Whole Roasted Beans, Brazil	
18	S-ORD101002	10000	Adatum Corporation	20000	1928-S	AMSTERDAM Lamp	
19	S-ORD101002	10000	Adatum Corporation	20000	WRB-1001	Whole Roasted Beans, Brazil	
20	S-ORD101003	30000	School of Fine Art	10000	1920-S	ANTWERP Conference Table	
21	S-ORD101003	30000	School of Fine Art	10000	1968-S	MEXICO Swivel Chair, black	
22	S-ORD101003	30000	School of Fine Art	10000	1996-S	ATLANTA Whiteboard, base	
23	S-ORD101003	30000	School of Fine Art	10000	2000-S	SYDNEY Swivel Chair, green	
24	S-ORD101003	30000	School of Fine Art	10000	WRB-1000	Whole Roasted Beans, Colombia	
25	S-ORD101003	30000	School of Fine Art	10000	WRB-1001	Whole Roasted Beans, Brazil	
26	S-ORD101003	30000	School of Fine Art	20000	1928-S	AMSTERDAM Lamp	
27	S-ORD101003	30000	School of Fine Art	20000	WRB-1001	Whole Roasted Beans, Brazil	
28	S-ORD101004	40000	Alpine Ski House	10000	1920-S	ANTWERP Conference Table	
29	S-ORD101004	40000	Alpine Ski House	10000	1968-S	MEXICO Swivel Chair, black	
30	S-ORD101004	40000	Alpine Ski House	10000	1996-S	ATLANTA Whiteboard, base	
31	S-ORD101004	40000	Alpine Ski House	10000	2000-S	SYDNEY Swivel Chair, green	
32	S-ORD101004	40000	Alpine Ski House	10000	WRB-1000	Whole Roasted Beans, Colombia	
33	S-ORD101004	40000	Alpine Ski House	10000	WRB-1001	Whole Roasted Beans, Brazil	
34	S-ORD101004	40000	Alpine Ski House	20000	1928-S	AMSTERDAM Lamp	
35	S-ORD101004	40000	Alpine Ski House	20000	WRB-1001	Whole Roasted Beans, Brazil	
36	S-ORD101005	10000	Adatum Corporation	10000	1920-S	ANTWERP Conference Table	
37	S-ORD101005	10000	Adatum Corporation	10000	1968-S	MEXICO Swivel Chair, black	
38	S-ORD101005	10000	Adatum Corporation	10000	1996-S	ATLANTA Whiteboard, base	
39	S-ORD101005	10000	Adatum Corporation	10000	2000-S	SYDNEY Swivel Chair, green	

PivotTable Fields
Active
All
Choose fields to add to report:
Search
Drag fields between areas below:
Filters
Columns
Rows
Values

tblSalesHeader

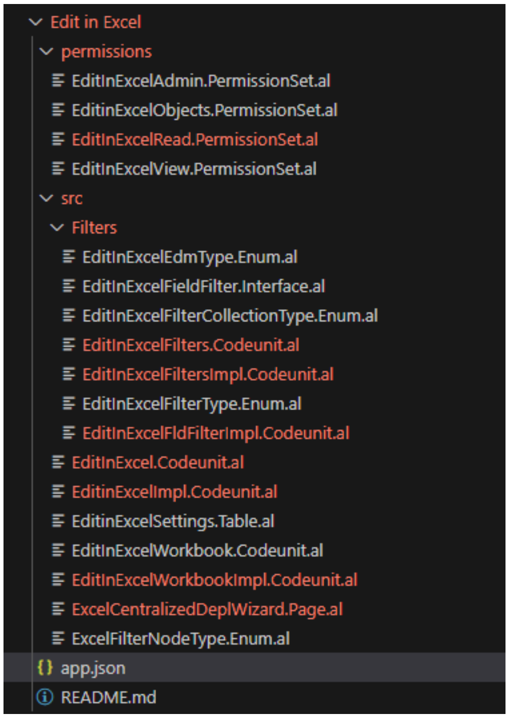
tblSalesLine
- ☐ Document_Type
- ☐ Document No.
- ☑ Line No.
- ☑ No.
- ☑ Description
- ☐ Amount Including VAT

Rows:
- No.
- Sell-to Customer ...
- Sell-to Customer ...
- Line No.
- No.

Now let's move on to the topic, how can we change the behavior or transfer it to new tables

Development

If we look at source code of "System Application" (you can find it in installation files for On-Prem "..\Applications\system application\source\System Application.Source.zip") we can find application "Edit in Excel"



```
Edit in Excel
  permissions
    EditInExcelAdmin.PermissionSet.al
    EditinExcelObjects.PermissionSet.al
    EditInExcelRead.PermissionSet.al
    EditInExcelView.PermissionSet.al
  src
    Filters
      EditInExcelEdmType.Enum.al
      EditInExcelFieldFilter.Interface.al
      EditInExcelFilterCollectionType.Enum.al
      EditInExcelFilters.Codeunit.al
      EditInExcelFiltersImpl.Codeunit.al
      EditInExcelFilterType.Enum.al
      EditInExcelFldFilterImpl.Codeunit.al
    EditInExcel.Codeunit.al
    EditinExcelImpl.Codeunit.al
    EditinExcelSettings.Table.al
    EditInExcelWorkbook.Codeunit.al
    EditInExcelWorkbookImpl.Codeunit.al
    ExcelCentralizedDeplWizard.Page.al
    ExcelFilterNodeType.Enum.al
  app.json
  README.md
```

README.md

In the README.md file you will find a detailed description of how the "Excel Add-In" can be modified. I will give here only some snippets from there.

How to download an Edit in Excel file

```
procedure Example()
var
    EditinExcel: Codeunit "Edit in Excel";
    EditinExcelFilters: Codeunit "Edit in Excel Filters";
    FileName: Text;
begin
    EditinExcelFilters.AddField('Journal_Batch_Name', Enum::"Edit in Excel
Filter Type"::Equal, JournalBatchName, Enum::"Edit in Excel Edm
Type"::"Edm.String");
    EditinExcelFilters.AddField('Journal_Template_Name', Enum::"Edit in Excel
Filter Type"::Equal, JournalTemplateName, Enum::"Edit in Excel Edm
Type"::"Edm.String");
    FileName := StrSubstNo('%1 (%2, %3)', CurrPage.Caption, JournalBatchName,
JournalTemplateName);
```

```
    EditinExcel.EditPageInExcel(CopyStr(CurrPage.Caption, 1, 240),
Page::"Example page", EditinExcelFilters, FileName);
end;
```

How to override Edit in Excel functionality

```
[EventSubscriber(ObjectType::Codeunit, Codeunit::"Edit in Excel",
'OnEditInExcelWithFilters', '', false, false)]
local procedure OnEditInExcelWithFilters(ServiceName: Text[240]; var
EditinExcelFilters: Codeunit "Edit in Excel Filters"; SearchFilter: Text; var
Handled: Boolean)
begin
    // Note: Since EditinExcelFilters is sent by var, you can simply modify
the filters and not handle the entire flow by not setting Handled := True
    if HandleOnEditInExcel(ServiceName, EditinExcelFilters, SearchFilter)
then
        Handled := True;
end;
```

How to generate your own Excel file

```
procedure CreateExcelFile(ServiceName: Text[250]; EditinExcelFilters:
Codeunit "Edit in Excel Filters"; SearchFilter: Text)
var
    EditinExcelWorkbook: Codeunit "Edit in Excel Workbook";
    FileName: Text;
begin
    // Initialize the workbook
    EditinExcelWorkbook.Initialize(ServiceName);

    // Add columns that should be shown to the user
    EditinExcelWorkbook.AddColumn(Rec.FieldCaption(Code), 'Code');
    EditinExcelWorkbook.AddColumn(Rec.FieldCaption(Name), 'Name');

    // Add any filters from the page (see below for how to create filters).
Note: It's allowed to filter on columns not added to the excel file
    EditinExcelWorkbook.SetFilters(EditinExcelFilters);

    // Download the excel file
    FileName := 'ExcelFileName.xlsx';
    DownloadFromStream(EditinExcelWorkbook.ExportToStream(), DialogTitleTxt,
'', '*.*', FileName);
end;
```

How to create filters

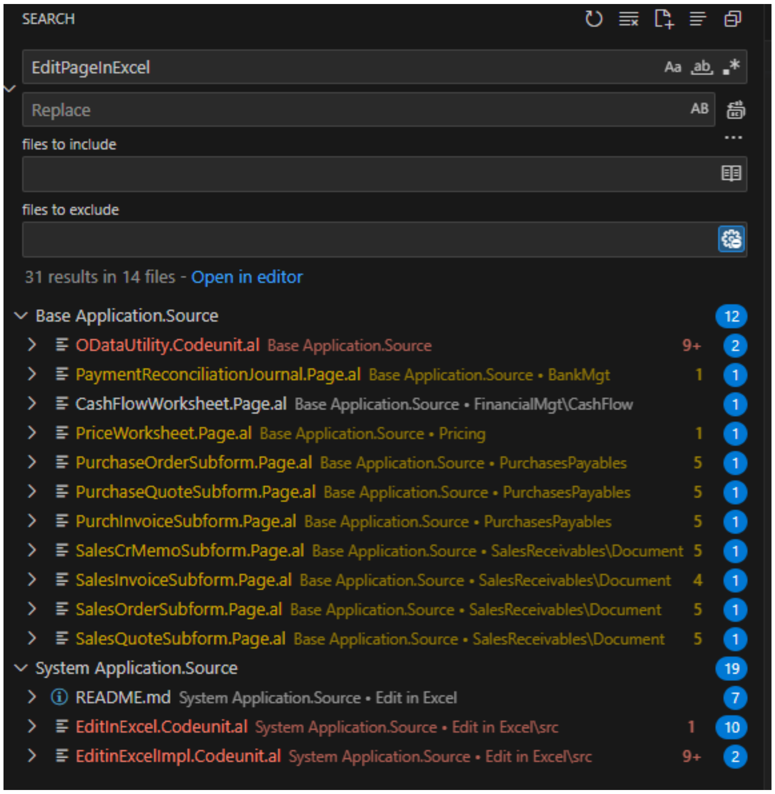
```
procedure CreateExcelFilters()
var
    EditinExcelFilters: Codeunit "Edit in Excel Filters";
begin
    // Let's add a simple filter "Blocked = False"
    EditinExcelFilters.AddField('Blocked', Enum::"Edit in Excel Filter
Type"::Equal, 'false', Enum::"Edit in Excel Edm Type"::"Edm.Boolean");
```

```
    // Now the filter "No. = 10000|20000"
    EditinExcelFilters.AddField('No_', Enum::"Edit in Excel Filter Collection
Type"::"or", Enum::"Edit in Excel Edm Type"::"Edm.String")
                        .AddFilterValue(Enum::"Edit in Excel Filter
Type"::Equal, '10000')
                        .AddFilterValue(Enum::"Edit in Excel Filter
Type"::Equal, '20000');

    // Finally let's add a range, "Amount = 1000..2000"
    EditinExcelFilters.AddField('Amount', Enum::"Edit in Excel Filter
Collection Type"::"and", Enum::"Edit in Excel Edm Type"::"Edm.Decimal")
                        .AddFilterValue(Enum::"Edit in Excel Filter
Type"::"Greater or Equal", '1000')
                        .AddFilterValue(Enum::"Edit in Excel Filter
Type"::"Less or Equal", '2000');

    // Since we did not clear EditinExcelFilters in between, the current
filter is "(Blocked = false) and (No_ = 10000|20000) and (Amount =
1000..2000)"
    // In other words, all the filters are added together.
end;
```

Some notes

The main procedure of the application is "EditPageInExcel", where web service for the specified page is created, and uses the web service to prepare and download an Excel file for the Edit in Excel functionality.

If we search where this function is referenced in "Base Application", we can see that it mostly Sales (Purchase) Subforms



```
EditPageInExcel
Replace

31 results in 14 files - Open in editor
Base Application.Source                                                              12
  ODataUtility.Codeunit.al Base Application.Source                              9+    2
  PaymentReconciliationJournal.Page.al Base Application.Source • BankMgt         1    1
  CashFlowWorksheet.Page.al Base Application.Source • FinancialMgt\CashFlow           1
  PriceWorksheet.Page.al Base Application.Source • Pricing                       1    1
  PurchaseOrderSubform.Page.al Base Application.Source • PurchasesPayables       5    1
  PurchaseQuoteSubform.Page.al Base Application.Source • PurchasesPayables       5    1
  PurchInvoiceSubform.Page.al Base Application.Source • PurchasesPayables        5    1
  SalesCrMemoSubform.Page.al Base Application.Source • SalesReceivables\Document 5    1
  SalesInvoiceSubform.Page.al Base Application.Source • SalesReceivables\Document 4   1
  SalesOrderSubform.Page.al Base Application.Source • SalesReceivables\Document  5    1
  SalesQuoteSubform.Page.al Base Application.Source • SalesReceivables\Document  5    1
System Application.Source                                                            19
  README.md System Application.Source • Edit in Excel                                 7
  EditInExcel.Codeunit.al System Application.Source • Edit in Excel\src          1   10
  EditinExcelImpl.Codeunit.al System Application.Source • Edit in Excel\src     9+    2
```

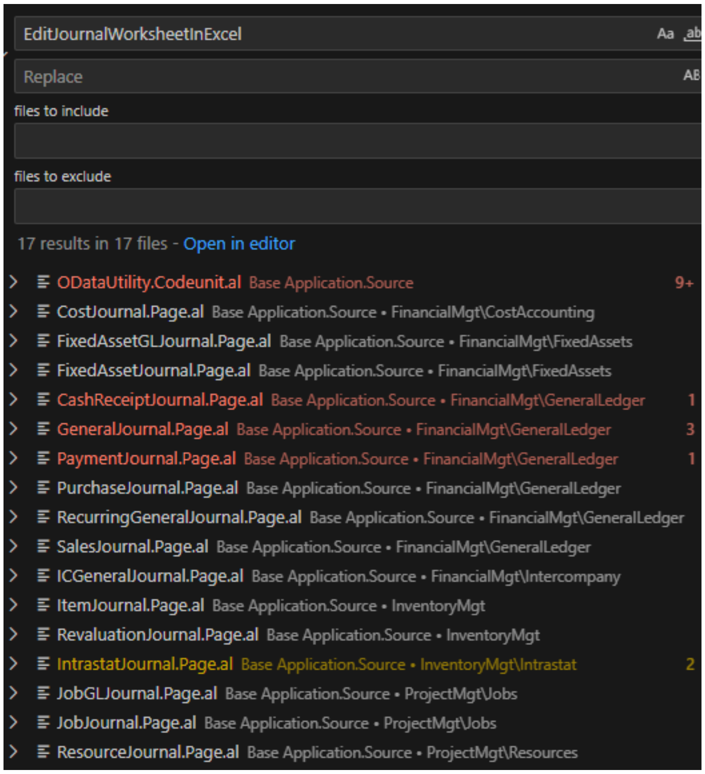
```
action(EditInExcel)
{
   ApplicationArea = Basic, Suite;
   Caption = 'Edit in Excel';
   Image = Excel;
   Promoted = true;
   PromotedCategory = Category8;
   PromotedIsBig = true;
   PromotedOnly = true;
   Visible = IsSaaSExcelAddinEnabled;
   ToolTip = 'Send the data in the sub page to an Excel file for analysis or
editing';
   AccessByPermission = System "Allow Action Export To Excel" = X;

   trigger OnAction()
   var
      EditinExcel: Codeunit "Edit in Excel";
      EditinExcelFilters: Codeunit "Edit in Excel Filters";
   begin
```

```
     EditinExcelFilters.AddField('Document_No', Enum::"Edit in Excel Filter
Type"::Equal, Rec."Document No.", Enum::"Edit in Excel Edm
Type"::"Edm.String");

      EditinExcel.EditPageInExcel(
        'Purchase_Order_Line',
        Page::"Purchase Order Subform",
        EditinExcelFilters,
        StrSubstNo(ExcelFileNameTxt, Rec."Document No."));
    end;
}
```

Also two procedures "EditJournalWorksheetInExcel" and "EditWorksheetInExcel" from codeunit 6710 ODataUtility are used in Journal Pages.

```
    procedure EditJournalWorksheetInExcel(PageCaption: Text[240]; PageId:
Text; JournalBatchName: Text; JournalTemplateName: Text)
    var
        EditinExcel: Codeunit "Edit in Excel";
        EditinExcelFilters: Codeunit "Edit in Excel Filters";
        ObjectId: Integer;
    begin
        EditinExcelFilters.AddField('Journal_Batch_Name', Enum::"Edit in
Excel Filter Type"::Equal, JournalBatchName, Enum::"Edit in Excel Edm
Type"::"Edm.String");
        EditinExcelFilters.AddField('Journal_Template_Name', Enum::"Edit in
Excel Filter Type"::Equal, JournalTemplateName, Enum::"Edit in Excel Edm
Type"::"Edm.String");

        Evaluate(ObjectId, CopyStr(PageId, 5));
        EditinExcel.EditPageInExcel(PageCaption, ObjectId,
EditinExcelFilters);
    end;

    procedure EditWorksheetInExcel(PageCaption: Text[240]; PageId: Text;
"Filter": Text)
    var
        EditinExcelHandler: Codeunit "Edit in Excel";
        ObjectId: Integer;
    begin
        Evaluate(ObjectId, CopyStr(PageId, 5));
        EditinExcelHandler.EditPageInExcel(PageCaption, ObjectId);
    end;
```

References to "EditJournalWorksheetInExcel"



EditJournalWorksheetInExcel

Replace

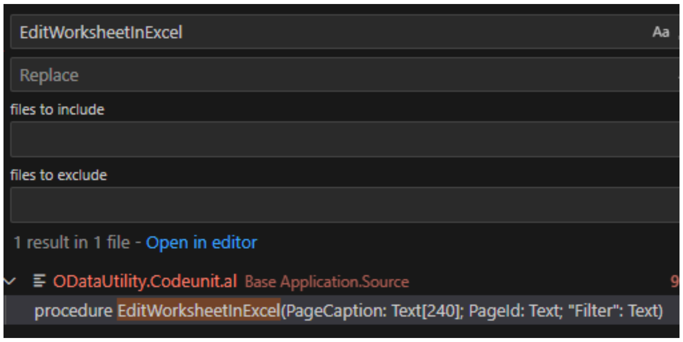
files to include

files to exclude

17 results in 17 files - Open in editor

- ODataUtility.Codeunit.al Base Application.Source 9+
- CostJournal.Page.al Base Application.Source • FinancialMgt\CostAccounting
- FixedAssetGLJournal.Page.al Base Application.Source • FinancialMgt\FixedAssets
- FixedAssetJournal.Page.al Base Application.Source • FinancialMgt\FixedAssets
- CashReceiptJournal.Page.al Base Application.Source • FinancialMgt\GeneralLedger 1
- GeneralJournal.Page.al Base Application.Source • FinancialMgt\GeneralLedger 3
- PaymentJournal.Page.al Base Application.Source • FinancialMgt\GeneralLedger 1
- PurchaseJournal.Page.al Base Application.Source • FinancialMgt\GeneralLedger
- RecurringGeneralJournal.Page.al Base Application.Source • FinancialMgt\GeneralLedger
- SalesJournal.Page.al Base Application.Source • FinancialMgt\GeneralLedger
- ICGeneralJournal.Page.al Base Application.Source • FinancialMgt\Intercompany
- ItemJournal.Page.al Base Application.Source • InventoryMgt
- RevaluationJournal.Page.al Base Application.Source • InventoryMgt
- IntrastatJournal.Page.al Base Application.Source • InventoryMgt\Intrastat 2
- JobGLJournal.Page.al Base Application.Source • ProjectMgt\Jobs
- JobJournal.Page.al Base Application.Source • ProjectMgt\Jobs
- ResourceJournal.Page.al Base Application.Source • ProjectMgt\Resources

I didn't find references to "EditWorksheetInExcel"



EditWorksheetInExcel

Replace

files to include

files to exclude

1 result in 1 file - Open in editor

- ODataUtility.Codeunit.al Base Application.Source 9

```
procedure EditWorksheetInExcel(PageCaption: Text[240]; PageId: Text; "Filter": Text)
```

Limits when using Excel for the web

When Edit in Excel is used on list pages for tables with many columns, the resulting workbook may have too many columns for the file to be viewed in Excel for the web. Business Central automatically limits the exported workbook to 100 columns when OneDrive is configured for system features.

Conclusion

"Excel Add-in" is a very powerful tool for changing data. But it must be used with extreme caution so as not to break the business logic and not lead to performance degradation if we want to change large amounts of data.