**A group of people raising their hands

Description automatically generated**

BC Add-in for Excel –   
Overview Services

**Contents**

* [Introduction](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#introduction)
* [How to install and configure for Business Central On-Premises](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#how-to-install-and-configure-for-business-central-on-premises)
  + [Notes](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#notes)
* ["Add-Ins Excel". How it works.](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#%22add-ins-excel%22.-how-it-works.)
* [Troubleshooting](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#troubleshooting)
  + [Issue:](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#**issue%3A**)
  + [Issue:](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#**issue%3A**)
* [Excel](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#excel)
* [Development](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#development)
  + [README.md](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#readme.md)
    - [How to download an Edit in Excel file](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#how-to-download-an-edit-in-excel-file)
    - [How to override Edit in Excel functionality](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#how-to-override-edit-in-excel-functionality)
    - [How to generate your own Excel file](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#how-to-generate-your-own-excel-file)
    - [How to create filters](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#how-to-create-filters)
  + [Some notes](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#some-notes)
* [Limits when using Excel for the web](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#limits-when-using-excel-for-the-web)
* [Conclusion](https://dev.azure.com/ciellos-bc/Wiki/_wiki/wikis/Internal%20Wiki/101/BC-Add-in-for-Excel#conclusion)

# 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)](https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/administration/configuring-excel-addin)

[Get the Business Central Add-in for Excel (Opens in new window or tab)](https://docs.microsoft.com/en-us/dynamics365/business-central/admin-deploy-excel-addin)

[Viewing and Editing in Excel From Business Central (Opens in new window or tab)](https://docs.microsoft.com/en-ca/dynamics365/business-central/across-work-with-excel)

## 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)](https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/administration/update-excel-addin-configuration)

# "Add-Ins Excel". How it works.

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

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

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

**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**.

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"**

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)](https://docs.microsoft.com/en-us/microsoft-365/admin/manage/manage-deployment-of-add-ins?view=o365-worldwide&preserve-view=true) |
| 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**.

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)*](https://learn.microsoft.com/en-us/dynamics365-release-plan/2021wave1/smb/dynamics365-business-central/exclude-flowfields-fields-outside-repeater-etag-odata) So stay tuned for new releases.

## ****Issue:****

Metadata was enable to retrieved for entry 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.

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"

Let's look at options

where we can change environment and company

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

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.

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***

Push **Done** and we get error

No problem, create new tab like this

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

**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"

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

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

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

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"

## 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

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"**

I didn't find references to "EditWorksheetInExcel"

# 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.