

Microsoft Dynamics 365 and Power Platform Community Conference

MAY 26 – 28, 2025
Portorož, Slovenia, Europe



FINANCE &
OPERATIONS



BUSINESS
CENTRAL



CUSTOMER
ENGAGEMENT



POWER
PLATFORM



5th
TRACK



PARTNER
BizTRACK



Microsoft Dynamics 365 and Power Platform Community Conference

MAY 26 – 28, 2025
Portorož, Slovenia, Europe

**FERNANDO TUDELA
DESANTES**

*Senior Technical Consultant
@Illumina Technology Solutions*



Demystifying Electronic Reporting: Concepts and Functions You May Not Yet Know, But Should Know

ALBIN LOTRIČ

*Senior Technical Consultant
@Docentric*



Some Tips & Tricks for Working With ER Configurations



I want to
move it,
move it!



Configurations > User Parameters

- Run settings = Yes
- Use destinations for draft status = Yes
- Enable data debugging at format run = Yes



User parameters

Parameters

Disable relative paths in expressi...	<input type="radio"/> No	Suppress baseline warnings	<input type="radio"/> No
Continue configurations consiste...	<input type="radio"/> No	Use destinations for draft status	<input checked="" type="radio"/> Yes
Disable advanced formula editor	<input type="radio"/> No	Validate the configuration after i...	<input type="radio"/> No
Generate performance trace	<input type="radio"/> No	Validate the configuration after r...	<input type="radio"/> No
Model refactoring mode	<input type="radio"/> No	Validate and repair delta	<input type="radio"/> No
Skip validation at configuration's...	<input type="radio"/> No	CONFIGURATION DEBUGGING	
Automatically run the procedure ...	<input type="radio"/> No	Enable data debugging at format...	<input checked="" type="radio"/> Yes
Run in debug mode	<input type="radio"/> No	EXECUTION TRACING	
Run model mapping instead of f...	<input type="radio"/> No	Collect query statistics	<input type="radio"/> No
Select used fields only (Experi...	<input type="radio"/> No	Trace access of caching	<input type="radio"/> No
Show file generation time	<input type="radio"/> No	Trace data access	<input type="radio"/> No
Show partial import/export	<input type="radio"/> No	Trace list enumeration	<input type="radio"/> No
Run settings	<input checked="" type="radio"/> Yes	Execution trace format	None
Skip copying prerequisites from ...	<input type="radio"/> No	Trace query	<input type="radio"/> No
		Query trace text mode	None

OK Cancel

Avoid Spaces in Data Model Node Names

- I don't recommend spaces in ER data model node names
- They can have side effects in some scenarios
- Just use Pascal casing, as in the screenshot below:



SPACES HURT

Data model | Invoice model : 277 | Standard view

+ New ▾ | Delete

SEARCH

Account type(LedgerJournalACType)

▸ Additional document reference(AdditionalDocumentReference)

AddressType_1

AdjustmentType_1

▸ AdvanceInvoice

▸ **Attached notes(AttachedNotes)**

▸ Backorder invoice lines(CustInvoiceBackorderLine)

▸ Bank account(BankAccount)

▸ Charges transaction(MarkupTransaction)

▸ Commercial invoice(TMSCCommercialInvoice)

▸ Company Info(CompanyInfo)

Company type(CompanyType_MX)

▸ Contact person(Contact)

▸ Contract line items(ProjectContractLineItem)

▸ Corrected invoice(CorrectedInvoice)

Correction method(CustVendCorrectiveReasonMethod)

Country/region type(CountryRegionType)

CustAdvancedInvoice

Invoice

Data model

Root reference

GENERAL

Name

Invoice

Description

Invoice

Node

GENERAL	TYPE
Name	Natural key
AttachedNotes	
Label (*Recommended to use lab...	
Attached notes	
Description	

A Word or Two on Labels

- Put custom labels in the data model or format
 - You can use system labels
 - If you use GER custom labels, provide translations
 - You can export/import label files
- Several report language options for the format configuration
 - User preference/company preference/explicitly defined/run-time



DM_WorkerName_Value ▼ : ✕ ✓ f_x ▼ Contoso Sales Team

Project invoice

[Your Company Name]

Company Address

[Street Address,	Phone:	[Phone number]	Bank Giro:	[Bank Giro]
City,	Fax:	[Fax number]	Tax registration number:	[Tax registration number]
State	Email:	[Email Address]		
ZIP Code]				

Customer information

[Name]
[Street Address, City, State ZIP Code]

Phone: [Phone number]
Fax: [Fax number]
VAT Number: [VAT number]

Invoice Id:

Invoice Date:	
Project contract:	[Project contract]
Invoice account:	[Invoice account]
Our account number:	[Our account number]
Paym terms:	[Payment terms]
Invoice currency:	[Invoice currency]
Payment reference:	[Payment reference]
Corrected invoice:	[Corrected invoice]
Reason for correction:	[Reason for correction]

[Invoice ID]

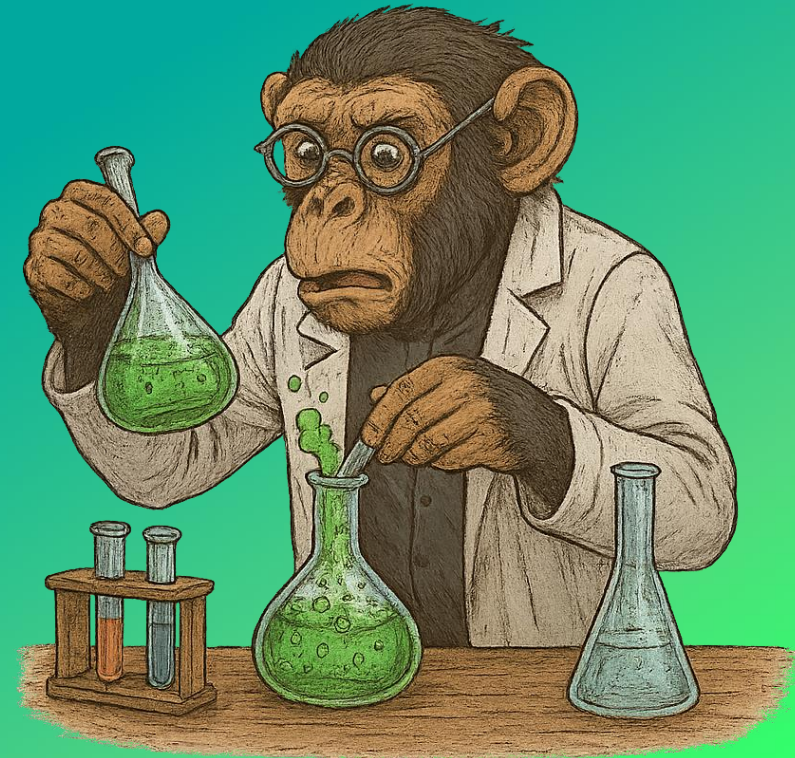
[Project contract]
[Invoice account]
[Our account number]
[Payment terms]
[Invoice currency]
[Payment reference]
[Corrected invoice]
[Reason for correction]

Project	Type	Category	Description	Quantity	Extended price	Amount	Print code	Base price	Index
Project	Empl			0.00	0.00	0.00			

Demo of Practical ~~Example~~ Experiment

Query the worker's title of the currently logged in user

This example shows how you can use some of the ER functions and different data source types.



And Now Over to Fernando's Magic...



I want to
move it,
move it!



Once upon a time...

Back in 2018, I received a requirement to tweak an Electronic Reporting file generation... And then all started.



Why using ER at all?

- **As replacement for SSRS**

- Prettier and faster designs
- No development needed (almost always)
- Functional people can work on them (at least in the format designs)

- **As an option for integrations**

- Create complex structured files (Unlike DMF or Odata)
- No development needed
- Re-use and adapt already existing MS provided common formats
 - ✓ ISO20022
 - ✓ Taxes
 - ✓ E-Invoice
 - ✓ Etc.



Real requirement:

- + “We need to add the charges to the Sales Order Confirmation report” - Random customer
- “Ok, It shouldn’t be hard, let me handle it, I might have it done by EOD” - Ferni

Later the same day,
me:



Complexity breakdown

- Charges (*MarkupTrans*) are at header and Line level.
(*Customer wanted them listed all together*).
- They are related to Confirmation tables (*CustConfirmJour* and *Trans*) In a normal scenario
- In Proforma scenario, charges are not related to confirmation tables, since they don't exist yet, we need to go to Sales order tables direct relation to find them



Should I give up?

- I can probably cover all this many cases with development
- But... ER and Devs are not the best friends from an ALM standpoint.
- Also any changes in the future will need dev iterations.
- And in my case... I was a big mouth



Let's go step by step:

- **Format**
- **Model**
- **Model Mapping**



Let's go step by step: "AllItemsQuery()"

```
▲ $SalesConfirmHeader: Calculated field = FIRSTORNUL(ReportDataProvider.getSalesConfirmHeaderTmp): Record
  ▷ #AnticipatedMonetaryTotal: Container
  ▷ $FTDSalesTable: Calculated field = IF(@.IsProForma=Enums.NoYes.Yes, FIRSTORNUL(FILTER(Tables.'$SalesTable',
  ▷ $Notes: Calculated field = SPLIT(@.MainNotes, CHAR(10)): Record list
  { ▷ <Relations: Record    <Relations: from 1 to N relations (EX: Header-lines)
    ▷ >Relations: Record    >Relations: from N to 1 relations (EX: Line-Header)
```

First let's get Confirm Jour and Trans related Charges:

```
ALLITEMSQUERY(@.'$SalesConfirmHeader'.>Relations'.CustConfirmJour'.<Relations'.MarkupTrans)
```

```
ALLITEMSQUERY(@.'$SalesConfirmHeader'.>Relations'.CustConfirmJour'.
<Relations'.CustConfirmTrans'.<Relations'.MarkupTrans)
```



Let's go step by step: "AllItemsQuery()"

First let's get Confirm Jour and Trans related Charges:

And we store them in 2 Calculated fields (variables).

Variables: Container

- ▶ \$Customer: Calculated field = @.'\$SalesConfirmHeader'.>Relations'.SalesId'.custTable_CustAccount(): Record
- ▶ \$FTDAllSalesLines: Calculated field = FILTER(Tables.'\$SalesLine', Tables.'\$SalesLine'.SalesId=@.'\$SalesConfirmHeader'.SalesId): Record list
- ▶ \$FTDMarkupTransCustConfirmJour: Calculated field = ALLITEMSQUERY(@.'\$SalesConfirmHeader'.>Relations'.CustConfirmJour'.<Relations'.MarkupTrans): Record list
- ▶ \$FTDMarkupTransCustConfirmTrans: Calculated field = ALLITEMSQUERY(@.'\$SalesConfirmHeader'.>Relations'.CustConfirmJour'.<Relations'.CustConfirmTrans'.<Relations'.MarkupTrans): Record list



Let's go step by step: “ FirstOrNull() & Filter() & If()”

Sales Order related charges (needed in proforma).

We first need to get the Sales order and Lines.

Sales Order:

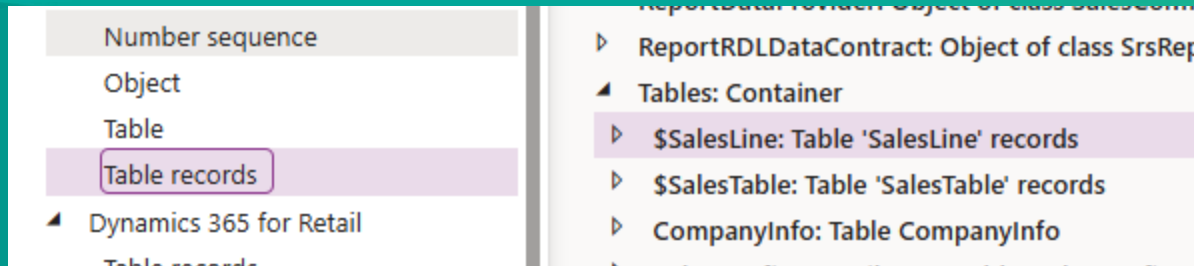
```
IF(@.IsProForma=Enums.NoYes.Yes,  
FIRSTORNULL(FILTER(Tables.'$SalesTable', Tables.'$SalesTable'.SalesId=@.SalesId)),  
@.'>Relations'.CustConfirmJour.'salesTable()')
```



Let's go step by step: “ FirstOrNull() & Filter() & If()”

Sales Lines, let's go simple:

`FILTER(Tables.'$SalesLine', Tables.'$SalesLine'.SalesId=@.'$SalesConfirmHeader'.SalesId)`



Let's go step by step: " FirstOrNull() & Filter() & If()"

Once we have the SalesTable and Lines saved in Calculated fields:

```
> $Customer: Calculated field = @.$SalesConfirmHeader.>Relations.SalesId.CustTable_CustAccount().Record  
▷ $FTDAllSalesLines: Calculated field = FILTER(Tables.'$SalesLine', Tables.'$SalesLine'.SalesId=@.$SalesConfirmHeader.SalesId): Record list
```

```
■ $SalesConfirmHeader: Calculated field = FIRSTORNUL(ReportDataProvider.getSalesConfirmHeaderTmp): Record  
▷ #AnticipatedMonetaryTotal: Container  
▷ $FTDSalesTable: Calculated field = IF(@.IsProForma=Enums.NoYes.Yes, FIRSTORNUL(FILTER(Tables.'$SalesTable', Tables.'$SalesTable'.SalesId=@.SalesId)), @.>Relations'.CustConfirmJour'.salesTable()):
```

Then, we create the CF for The related Charges:

```
> $FTDMarkupTransSalesTable: Calculated field = @.$SalesConfirmHeader.'$FTDSalesTable'.<Relations'.MarkupTrans: Record list
```

```
> FTDMarkupTransSalesLines: Calculated field = ALLITEMSQUERY(@.$FTDAllSalesLines'.<Relations'.MarkupTrans): Record list
```

Recap

We have gotten the following Record Lists in Variables:

- *\$FTDMarkupTransCustConfirmJour* – With the charges related to confirmation journal
- *\$FTDMarkupTransCustConfirmTrans* – With the charges related to all the Confirmation Trans
- *\$FTDMarkupTransSalesTable* – With the charges related to the SalesTable
- *\$FTDMarkupTransSalesLine* – With the charges related to all the Sales Lines

What do we do now?



2 Options:

Contact your trusted expert

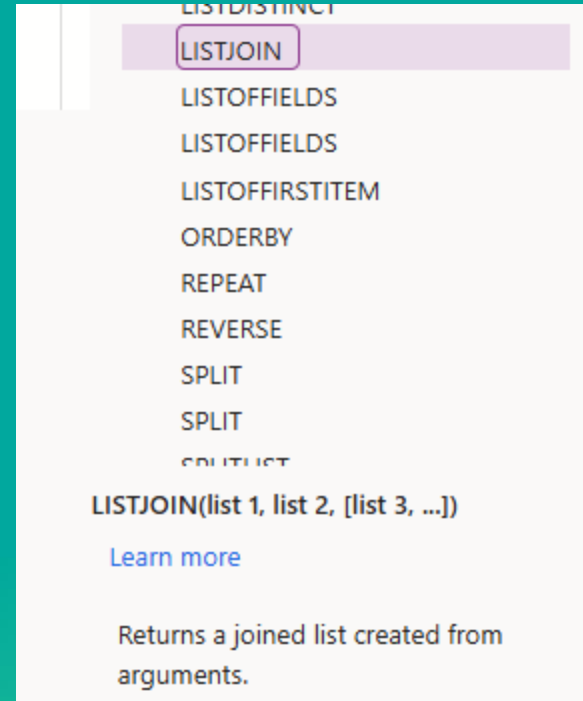


Albin Lotrič ✓ · 1er
Senior Technical Consultant - Microsoft Dynamics 365 F&O
Eslovenia · [Información de contacto](#)
Más de 500 contactos
Gustav Sundblad, Naresh G y 164 contactos en común más
[Enviar mensaje](#) [Más](#)

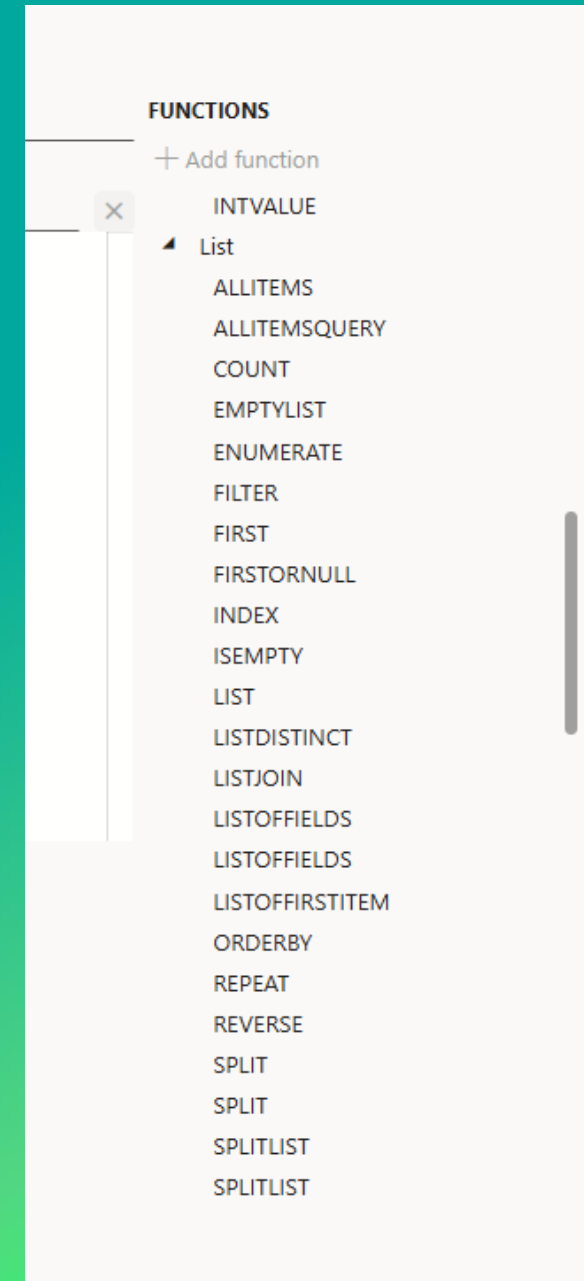


Fernando Tudela Desantes ✓
Microsoft MVP | Dynamics 365 F.O./Commerce Tech architect in Illumina Technology Solutions
Valencia y alrededores · [Información de contacto](#)
3612 seguidores · Más de 500 contactos
[Tengo interés en...](#) [Añadir sección](#) [Ver mi blog](#) [Re](#)

Check the functions and try:



LISTDISTINCT
LISTJOIN
LISTOFFIELDS
LISTOFFIELDS
LISTOFFIRSTITEM
ORDERBY
REPEAT
REVERSE
SPLIT
SPLIT
SPLITLIST
LISTJOIN(list 1, list 2, [list 3, ...])
[Learn more](#)
Returns a joined list created from arguments.



FUNCTIONS
+ Add function
x
INTVALUE
List
ALLITEMS
ALLITEMSQUERY
COUNT
EMPTYLIST
ENUMERATE
FILTER
FIRST
FIRSTORNUL
INDEX
ISEMPTY
LIST
LISTDISTINCT
LISTJOIN
LISTOFFIELDS
LISTOFFIELDS
LISTOFFIRSTITEM
ORDERBY
REPEAT
REVERSE
SPLIT
SPLIT
SPLITLIST
SPLITLIST

All Charges:

```
IF(@.'$SalesConfirmHeader'.IsProForma=Enums.NoYes.No,  
    LISTJOIN(@.'$FTDMarkupTransCustConfirmJour', @.'$FTDMarkupTransCustConfirmTrans'),  
    LISTJOIN(@.'$FTDMarkupTransSalesTable', @.FTDMarkupTransSalesLines))
```



FINALLY!



YOU REALLY
THINK YOU'RE
DONE?

THANK YOU!

