



2026 DYNAMICS CON Vegas



2026 DYNAMICS CON



Cracking the Code: Event Subscriptions for AI Development Success



Who is this session for?



Who is this session for?

- AL Developers
- Anyone looking to
 - Deepen their understanding of event subscriptions
 - Understand how to customize and extend Business Central effectively
 - Bridge the gap between business process and technical implementations

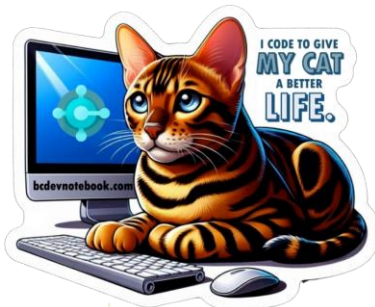


Tonya Bricco-Meske

- Manager, BC Development at RSM
- Work remotely from Central Wisconsin
- Started NAV development in 2016
- Reporting and Query Queen



BCDEVNOTEBOOK.COM



Tonya
Bricco-Meske



2

0

2

6



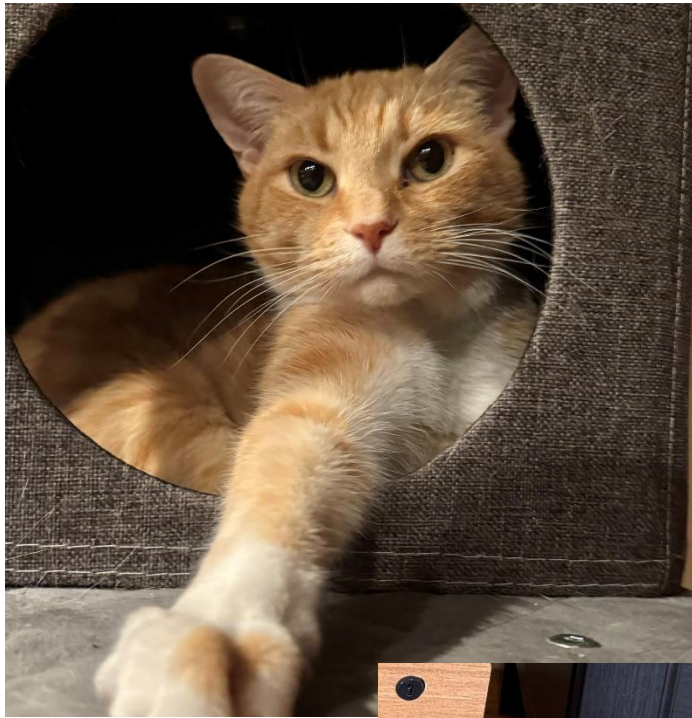
Henrik Helgesen

- President, Helgesen Consulting, Inc.
- Born and raised in Denmark.
- Currently lives in the Los Angeles, CA area.
- Began with NAV in 1995 at a Furniture Manufacturer
- Has done projects in all aspects of human life from before you are born till the day you die.
- Hockey Fan – Go Kings!
- Magic Fan – Magic Castle Member
- Whisky Fan - Slàinte mhath



**Henrik
Helgesen**





Who are you?



- Partner
- ISV
- End User

- 🧐 Developer?
- 😼 non-developer?

Agenda

- Implementing Best Practices in Eventing
- Understand and utilize Event Handling Tools
- Apply Practical Examples and Manage Data Flow



What is an Event

Significant occurrence or change in state within an application that can trigger a response or action



How do you subscribe to an event?

```
[EventSubscriber(ObjectType:<Event Publisher Object Type>,  
<Event Publisher Object>, '<Published Event Name>', '<Published Event Element Name>', <SkipOnMissingLicense>, <SkipOnMissingPermission>)]
```

Object Type

Object Name

Event Name

Event Element Name

Skip on Missing License

Skip on Missing Permission

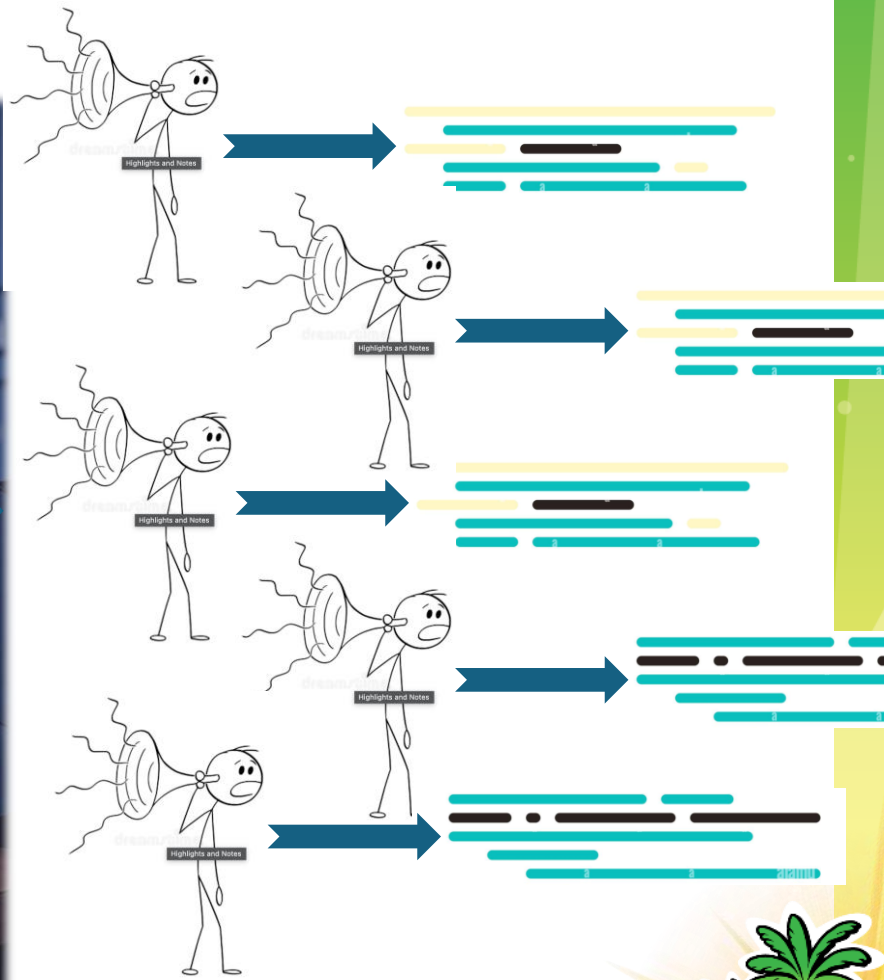
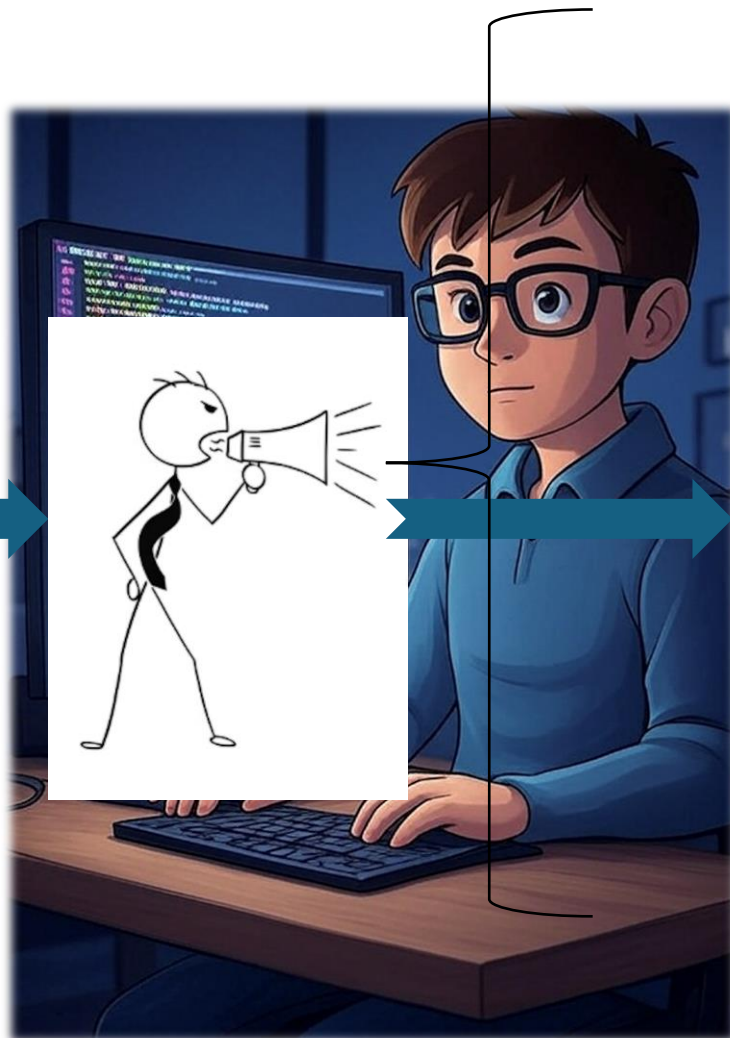
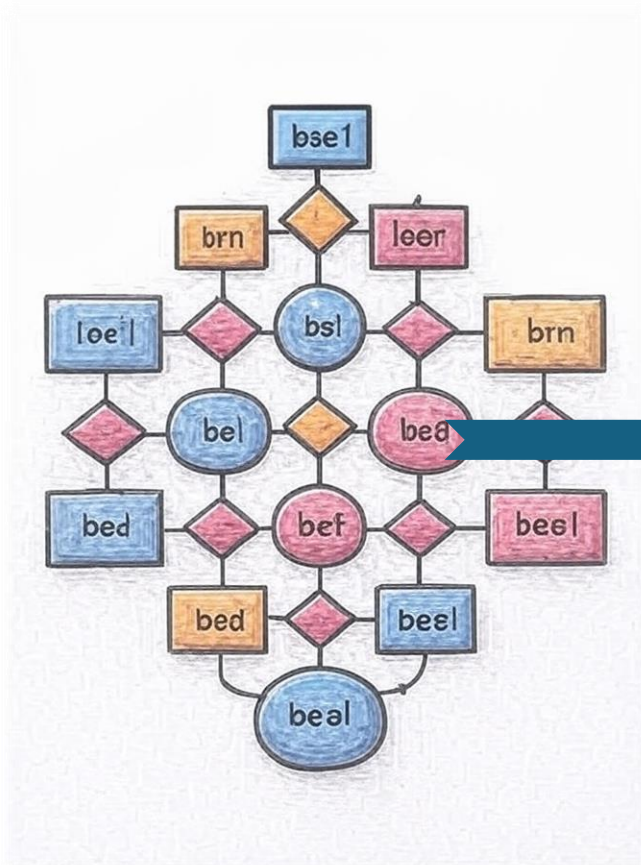


Why Use Events?

- Flexible Customizations
- Real-Time Automation
- Simplified Upgrades and Maintenance



How Do Events Work



Types of Events



Types of Events

Business Event

- Defined that structure is not going to change
- Facilitate communication between BC and external systems
- Do not have access to global variables

```
[BusinessEvent(IncludeSender: Boolean [, Isolated: Boolean])]
```

```
[ExternalBusinessEvent(Name: Text, DisplayName: Text, Description: Text, Category: enum [, Version: Text])]
```



Types of Events

Business Event

```
[ExternalBusinessEvent('PurchaseReceiptPosted', 'Purchase receipt posted',  
    'This business event is triggered when a purchase order is received.', EventCategory::"My Purchase Events")]  
1 reference  
local procedure MyBusinessEventPurchaseReceivedPosted(PurchaseInvoiceId: Guid; Url: Text[250]; Base64Text: Text[250])  
begin  
end;
```



```
[EventSubscriber(ObjectType::Codeunit, Codeunit::"Purch.-Post", OnAfterFinalizePosting, '', true, true)]
```

```
0 references
```

```
local procedure OnAfterFinalizePosting(  
var PurchHeader: Record "Purchase Header"; var PurchRcptHeader: Record "Purch. Rcpt. Header";  
var PurchInvHeader: Record "Purch. Inv. Header"; var PurchCrMemoHdr: Record "Purch. Cr. Memo Hdr.";  
var ReturnShptHeader: Record "Return Shipment Header"; var GenJnlPostLine: Codeunit "Gen. Jnl.-Post Line";  
PreviewMode: Boolean; CommitIsSupressed: Boolean)  
var  
    Url: Text[250];  
    PurchaseReceiptsApiUrlTok: Label 'v2.0/companies(%1)/purchaseReceipts(%2)', Locked = true;  
    Outstream: OutStream;  
    Instream: InStream;  
    RecRef: RecordRef;  
    Variant: Variant;  
    Base64: Codeunit "Base64 Convert";  
    TempBlob: Codeunit "Temp Blob";  
    Base64Text: Text;  
begin  
    if PurchRcptHeader."No." <> '' then begin  
        Url := GetBaseUrl() + StrSubstNo(PurchaseReceiptsApiUrlTok, GetCompanyId(), TrimId(PurchRcptHeader.SystemId));  
        //Print purchase receipt  
        PurchRcptHeader.SetRecFilter();  
        RecRef.GetTable(PurchRcptHeader);  
        Outstream := TempBlob.CreateOutStream();  
        if Report.SaveAs(Report::"Purchase - Receipt", '', ReportFormat::Pdf, Outstream, RecRef) then begin  
            Instream := TempBlob.CreateInStream();  
            Base64Text := Base64.ToBase64(Instream);  
            MyBusinessEventPurchaseReceivedPosted(PurchRcptHeader.SystemId, Url, Base64Text);  
        end;  
    end;  
end;  
end;
```



Types of Events

Business Event

```
[ExternalBusinessEvent('PurchaseReceiptPosted', 'Purchase receipt posted',  
    'This business event is triggered when a purchase order is received.', EventCategory::"My Purchase Events")]  
1 reference  
local procedure MyBusinessEventPurchaseReceivedPosted(PurchaseInvoiceId: Guid; Url: Text[250]; Base64Text: Text[250])  
begin  
end;
```



When a business event occurs (V3)

Parameters Settings Code view About

Environment *

TONYA

Event *

The business event in Dynamics 365 Business Central.

'Event' is required.

Advanced parameters

Showing 1 of 1

Show all

Clear all

Company (optional)

The Dynamics 365 Business Central company. Leave this empty to subscribe to busi...

Connected to tonya.bricco-meske@ onmicrosoft.com. Change connection

Customer blocked - 1.0 (Microsoft)

Customer unblocked - 1.0 (Microsoft)

...

Job queue task failed - 1.0 (Microsoft)

Opportunity activated - 1.0 (Microsoft)

Opportunity closed as lost - 1.0 (Microsoft)

Purchase credit memo posted - 1.0 (Microsoft)

Purchase invoice posted - 1.0 (Microsoft)

Purchase order released - 1.0 (Microsoft)

Purchase payment posted - 1.0 (Microsoft)

Purchase receipt posted - 1.0 (Microsoft)

Purchase receipt posted (Tonya Bricco-Meske)

...

...

...

...

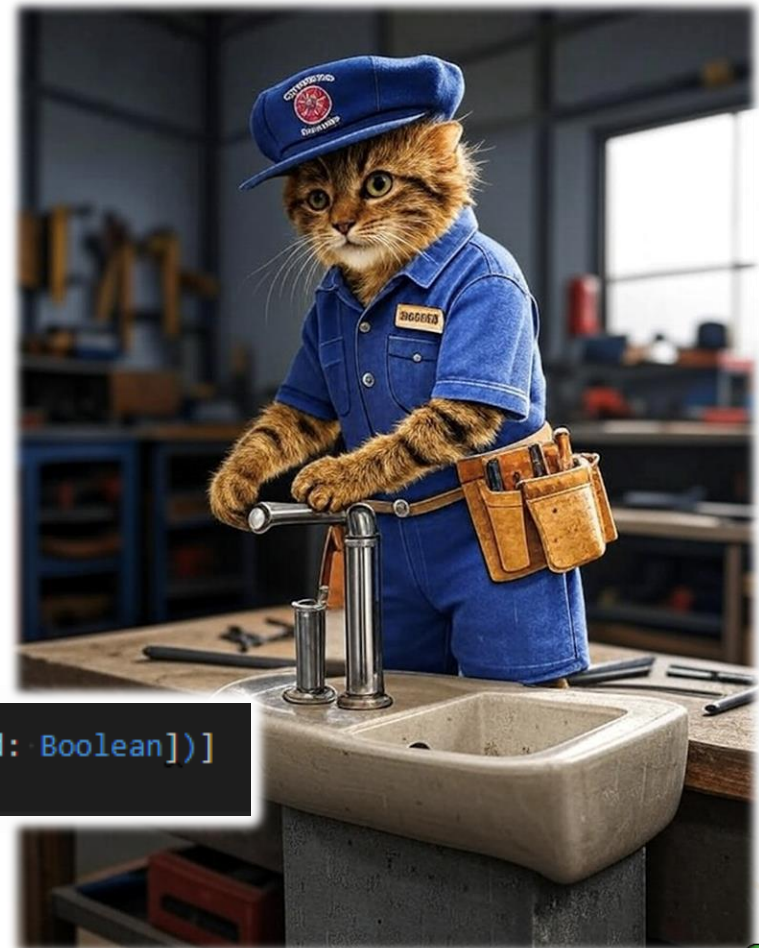
Quote created for opportunity - 1.0 (Microsoft)



Types of Events

Integration Events

- Extend Business Central at specific points in application logic
- Base Application
- ISV Solutions
- Your own solutions



```
[IntegrationEvent(IncludeSender: Boolean, GlobalVarAccess: Boolean [, Isolated: Boolean]]
```

Types of Events

Integration Events

- Code

```
SalesHeader := SalesHeader2;  
OnCodeOnBeforeFillTempLines(SalesHeader, CalledBy);  
FillTempLines(SalesHeader, TempSalesLineGlobal);  
  
// Check that the invoice amount is zero or greater  
OnRunOnBeforeCheckTotalInvoiceAmount(SalesHeader);
```

```
[IntegrationEvent(false, false)]  
local procedure OnRunOnBeforeCheckTotalInvoiceAmount(var SalesHeader: Record "Sales Header")  
begin
```

```
[IntegrationEvent(false, false)]  
local procedure OnRunOnBeforeCheckAndUpdate(var SalesHeader: Record "Sales Header")  
begin  
end;
```

```
OnRunOnBeforeFinalizePosting(  

```

```
[IntegrationEvent(false, false)]  
local procedure OnRunOnBeforeFinalizePosting(var SalesHeader: Record "Sales Header"; var SalesShipmentHeader: Record "Sales Shipment")  
begin  
end;
```

```
FinalizePosting(SalesHeader, EverythingInvoiced, TempDropShipPostBuffer);
```



Types of Events

Internal Event

- Most restrictive in scope
- Can only be subscribed within same module



Types of Events

Global Events

Global events are predefined system events that are automatically raised by various base application codeunits.

| Codeunit ID | Codeunit Name | Event |
|-------------|----------------------------|-------------------------------------|
| 9170 | Conf./Personalization Mgt. | OnRoleCenterOpen |
| | | OnAfterLogInEnd |
| | | OnBeforeLogInStart |
| | | OnBeforeCompanyOpen |
| | | OnAfterCompanyOpen |
| | | OnBeforeCompanyClose |
| 42 | TextManagement | OnAfterCompanyClose |
| | | OnBeforeMakeTextFilter |
| | | OnAfterMakeDateTimeFilter |
| | | OnAfterMakeDateFilter |
| 49 | GlobalTriggerManagement | OnAfterMakeTextFilter |
| | | OnAfterMakeTimeFilter |
| | | OnAfterGetGlobalTableTriggerMask |
| | | OnAfterOnGlobalInsert |
| | | OnAfterOnGlobalModify |
| | | OnAfterOnGlobalDelete |
| | | OnAfterOnGlobalRename |
| | | OnAfterGetDatabaseTableTriggerSetup |
| | | OnAfterOnDatabaseInsert |
| | | OnAfterOnDatabaseModify |
| | | OnAfterOnDatabaseDelete |
| | | OnAfterOnDatabaseRename |
| | | OnBeforeOnDatabaseInsert |
| | | OnBeforeOnDatabaseModify |
| | | OnBeforeOnDatabaseDelete |
| | | OnBeforeOnDatabaseRename |



Object Type Events (trigger events)

Table Events

- Insert
- Rename
- Validate
- Modify
- Delete



**Trigger
Event -
Before**

OnBeforeDelete

**Table
Trigger**

onDelete

**Global table
trigger in
codeunit**

OnDatabaseDelete

**Database
Operations**

Delete the record

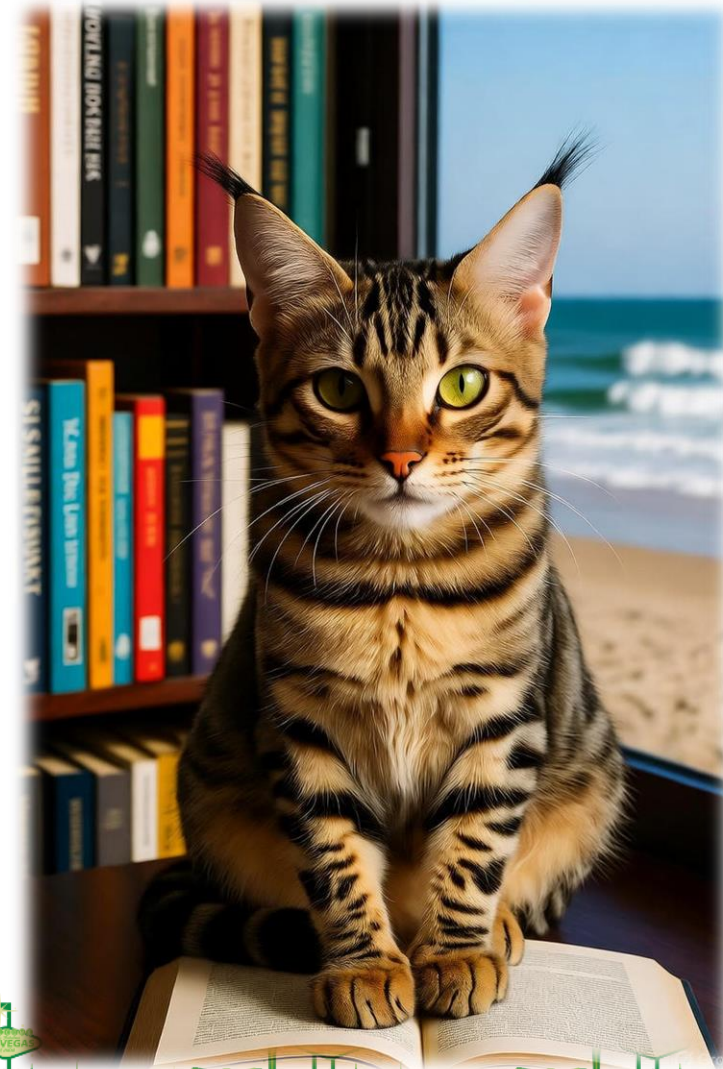
**Trigger
Event -
After**

OnAfterDelete

Object Type Events (trigger events)

Page Events

- OnAfterGetRecord
- OnAfterGetCurrRecord
- OnNewRecord



Object Type Events (trigger events)

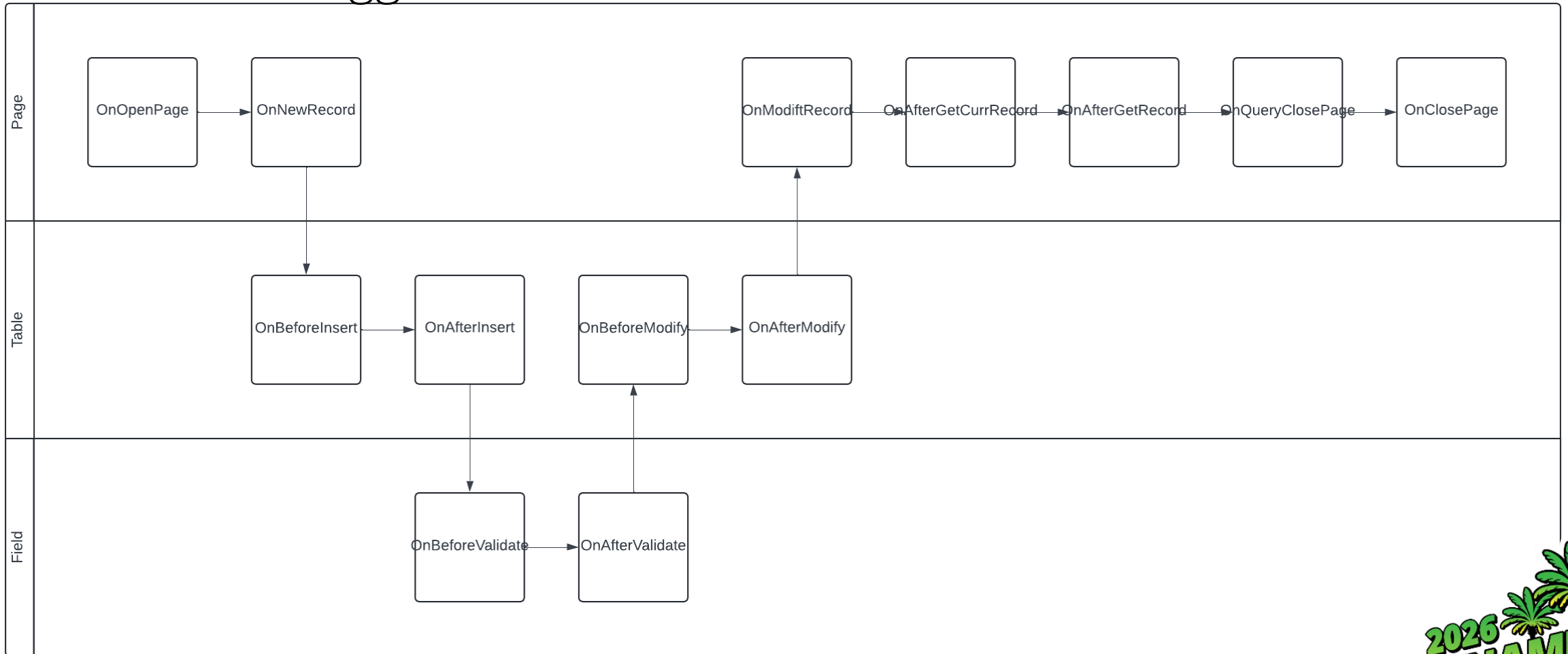
Report Events

- OnInitReport
- Request Page --> page triggers
- OnPreReport
- Data Item Events
 - OnPreDataItem
 - OnAfterGetRecord
 - OnPostDataItem
- OnPostReport

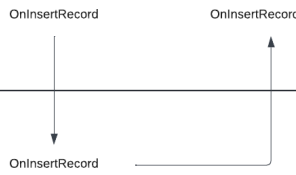


Execution Order of Events

New Record – Trigger Events



Execution Order of Events

| | |
|----------|---|
| Page |  |
| PageExt | |
| Table | |
| TableExt | |
| Global | |

Trigger Sequence in AL

Jan 1, 2025

Here's a small *"let me bookmark this"* post.

I often did not completely understand trigger and event sequences in AL. I **sort of** knew them, but I doubted myself just enough to keep going back to the documentation. It's not like the documentation is bad, but every time it took me just a few additional minutes to figure out the answer, that I decided I wanted to **visualize** this.

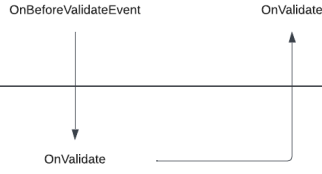
It came in handy for me, maybe it will for you too. I'd love to hear if you'd add any other sequence!

Few things to remember that I didn't want to jam into the visualizations

- Apps with **lower IDs** execute their triggers and subscribers **before those with higher IDs**.
For example, if *App A* has an ID like *"1bs3.."* and *App B* has *"c23f.."*, *App A*'s triggers and subscribers will always run first.
- Within the same App, objects with **lower Object IDs** execute their triggers and subscribers **before those with higher Object IDs**.
A subscriber in *Codeunit 50000* will execute before one in *Codeunit 50001*. Similarly, triggers in *Table Extension 50000* will run before those in *Table Extension 50001*.
- Subscribers in the same codeunit are executed in **alphabetical order** by procedure name.
For instance, a procedure named *MySubscriber1* will run before *MySubscriber2*, regardless of their order in the code.

When looking at the Lookup flow, remember that **only one OnLookup trigger can be executed** at a time. Once the first one is executed, others are ignored.

I made a small trigger logger for the purpose of this post, you can find it [here](#). It's a simple extension that logs the trigger and event sequence to a table.

| | |
|--|---|
| |  |
| | |
| | |
| | |
| | |



<https://tine.staric.net/blog/2025/trigger-sequence/>



Event Handling Tools



Event Recorder

- Built in tool designed to help discover events raised during a process
- Includes name, type, and where it is published
- AL Snippet Generation



Event Recorder

Record Events
🔍

| Call Order ↑ | Object Type ↑ | Object Name | Event Name ↑ | Element Name ↑ | Calling Object Type ↑ | Calling Object Name | Calling Method ↑ | Get AL Snippet |
|--------------|---------------|-------------|---------------------------------|---------------------|-----------------------|---------------------|-------------------------|-----------------|
| 342 | Table | Customer | OnAfterSetLastModifiedDateTi... | | Table | Customer | SetLastModifiedDateTime | Get AL Snipp... |
| 399 | Table | Customer | On... | | Table | Customer | SetLastModifiedDateTime | Get AL Snipp... |
| 624 | Table | Customer | On... | | Table | Customer | SetLastModifiedDateTime | Get AL Snipp... |
| 867 | Table | Customer | On... | | Table | Customer | SetLastModifiedDateTime | Get AL Snipp... |
| 1106 | Table | Customer | On... | | Table | Customer | SetLastModifiedDateTime | Get AL Snipp... |
| 1349 | Table | Customer | On... | | Table | Customer | SetLastModifiedDateTime | Get AL Snipp... |
| → 1547 | Table | Customer | On... | | Table | Customer | SetLastModifiedDateTime | Get AL Snipp... |
| 1098 | Table | Customer | On... | | Table | Customer | City - OnValidate | Get AL Snipp... |
| 620 | Table | Customer | OnAfterValidateEvent | CPD Customer Name | | | | Get AL Snipp... |
| 1099 | Table | Customer | OnAfterValidateEvent | City | | | | Get AL Snipp... |
| 617 | Table | Customer | OnAfterValidateEvent | Name | | | | Get AL Snipp... |
| 338 | Table | Customer | OnAfterValidateEvent | Payment Method Code | | | | Get AL Snipp... |
| 1540 | Table | Customer | OnAfterValidateEvent | Post Code | | | | Get AL Snipp... |
| 1539 | Table | Customer | OnAfterValidatePostCode | | Table | Customer | Post Code - OnValidate | Get AL Snipp... |
| 232 | Table | Customer | OnBeforeInsert | | Table | Customer | OnInsert | Get AL Snipp... |

i

[EventSubscriber(ObjectType::Table, Database::"Customer", 'OnAfterSetLastModifiedDateTime', "", true, true)]
 local procedure MyProcedure()
 begin
 end;

OK

Shipments

22

Posted Sales
Credit Memos

Invoices

0

Customer
Subscription
Contracts

Return Receipts

0

Subscriptions



Event Subscriptions

AL Explorer Project: ALProject1 (Default Publisher)

OBJECTS EVENTS APIS EXTENSIBLE ENUMS

Search: Type to filter Group by: None Module: All

| Signature | Type | Object Name ↑ | Module | Namespace |
|--|--------|----------------------------|------------------|--------------------------|
| OnBeforeValidateRegistrationNumber(var Customer: Rec... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnBeforeValidateShortcutDimCode(var Customer: Recor... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnBeforeValidateVATRegistrationNo(var Customer: Reco... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnBeforeVATRegistrationValidation(var Customer: Recor... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnGetBalanceAsVendorOnBeforeCalcBalance(var Vendor... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnGetCustNoOpenCardOnAfterMarkCustomersWithSim... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnGetCustNoOpenCardOnAfterOnAfterCustomerFilterFr... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnGetCustNoOpenCardOnAfterSetCustomerFilters(var C... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnGetCustNoOpenCardOnBeforeCustomerFindSet(var C... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnGetCustNoOpenCardOnBeforeFilterCustomer(var Cus... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnGetTotalAmountLCYOnAfterCalcFields(var Customer: ... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnGetTotalAmountLCYUIOnAfterSetAutoCalcFields(var C... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnMarkCustomersWithSimilarNameOnBeforeCustomerF... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnShowContactOnBeforeOpenContactCard(var Contact:... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnShowContactOnBeforeOpenContactList(var Contact: ... | Table | Customer | Base Application | Microsoft.Sales.Customer |
| OnAfterCalcRemainingAmount(var TempCustLedgerEntr... | Report | Customer - Balance to Date | Base Application | Microsoft.Sales.Reports |

Source Subscribe



AI

The image shows a screenshot of the Visual Studio Code interface. The command palette is open, displaying a search for 'AL: Find Event'. The search results are as follows:

- AL: Find Event (Shift + Alt + E, recently used)
- Preferences: Open User Settings (JSON)
- AL: Explorer (Ctrl + Shift + F12)
- AL: Download symbols
- AL: Go! (Alt + A, Alt + L)
- AL: Debug without publishing (Ctrl + Shift + F5)
- Git: Clone

Below these results, there is a section titled 'other commands' which includes:

- Accounts: Manage Extension Account Preferences
- Accounts: Manage Trusted Extensions For Account
- Accounts: Manage Trusted MCP Servers For Account
- Add Cursor Above (Ctrl + Alt + UpArrow)
- Add Cursor Below (Ctrl + Alt + DownArrow)
- Add Cursors to Bottom
- Add Cursors to Line Ends (Shift + Alt + I)
- Add Cursors to Top

The background shows a code editor with a file named 'codeunit 70103 "TBM_Event"' and a taskbar at the bottom with various application icons and system tray icons.



Event Subscriptions

Dynamics 365 Business Central

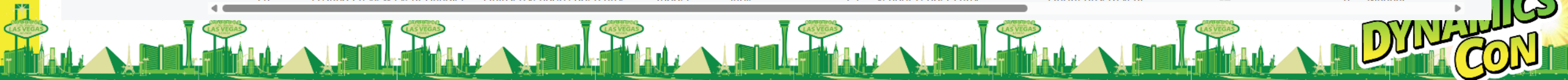
Environment: Sandbox

CRONUS USA, Inc. | RoyaltyCentral | Sales | Purchasing | All Reports

PA SO

Event Subscriptions:

| Subscriber Codeunit ID ↑ | Subscriber Codeunit Name | Subscriber Function ↑ | Event Type | Publisher Object Type | Publisher Object ID | Publisher Object Name | Published Function | Acti... | Number of Calls | Subscriber Instance |
|--------------------------|-------------------------------|---------------------------------|------------|-----------------------|---------------------|--------------------------------|--------------------|-------------------------------------|-----------------|---------------------|
| 20 | Posting Preview Event Handler | OnInsertBankAccountLedgerEn... | Trigger | Table | 271 | Bank Account Ledger Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertCustLedgerEntry | Trigger | Table | 21 | Cust. Ledger Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertDetailedCustLedgEntry | Trigger | Table | 379 | Detailed Cust. Ledg. Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertDetailedEmployeeLed... | Trigger | Table | 5223 | Detailed Employee Ledger Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertDetailedVendorLedgE... | Trigger | Table | 380 | Detailed Vendor Ledg. Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertEmployeeLedgerEntry | Trigger | Table | 5222 | Employee Ledger Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertExchRateAdjmtLedgEn... | Trigger | Table | 186 | Exch. Rate Adjmt. Ledg. Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertFALedgEntry | Trigger | Table | 5601 | FA Ledger Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertGLEntry | Trigger | Table | 17 | G/L Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertItemLedgerEntry | Trigger | Table | 32 | Item Ledger Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertJobLedgEntry | Trigger | Table | 169 | Job Ledger Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertMaintenanceLedgerEn... | Trigger | Table | 5625 | Maintenance Ledger Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertPhysInventoryLedgerE... | Trigger | Table | 281 | Phys. Inventory Ledger Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertResourceLedgerEntry | Trigger | Table | 203 | Res. Ledger Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertValueEntry | Trigger | Table | 5802 | Value Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertVATEntry | Trigger | Table | 254 | VAT Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |
| 20 | Posting Preview Event Handler | OnInsertVendorLedgerEntry | Trigger | Table | 25 | Vendor Ledger Entry | OnAfterInsertEvent | <input checked="" type="checkbox"/> | 0 | Manual |



Master Best Practices in Eventing



How to find the best event

What is the difference between an event, and a great event?

Technical Tip:

To make the field flow from the Item to the Sales Line, you'll typically:

1. Add the field to the Item table.
2. Extend the Sales Line table to include the field.
3. Modify the logic in the OnValidate trigger of the Sales Line.ItemNo field to copy the value from the Item record.

- Use Case: Enhances clarity in fulfillment and delivery processes.

1 codeunit 70101 "TBM_CoolCatCodeunit"

2 {

3 [EventSubscriber(ObjectType::Table, Database::"Sales Line", OnAfterValidateEvent, "No.", false, false)]

0 references

4 local procedure OnAfterValidateItemNo(var Rec: Record "Sales Line"; xRec: Record "Sales Line")

5 begin

6 if Rec.IsTemporary() then exit;

7 UpdateSalesLine(Rec);

8 end;

AL Sales

9

70

1 reference

78 10

procedure UpdateSalesLine(var Rec: Record "Sales Line")

186 11

var

212 12

Item: Record Item;

287 13

Resource: Record Resource;

288 14

begin

289 15

case Rec.Type of

290 16

Rec.Type::Item:

291 17

if Item.Get(Rec."No.") then

292 18

GetItemDetails(Rec, Item);

293 19

Rec.Type::Resource:

294 20

if Resource.Get(Rec."No.") then

295 21

GetResourceDetail(Rec, Resource);

296 22

end;

297 23

end;

298 24

1 reference

300 25

local procedure GetItemDetails(var Rec: Record "Sales Line"; var Item: Record Item)

301 26

begin

302 27

Rec."TBM_PackagingType" := Item."TBM_PackagingType";

303 28

end;

304 29

1 reference

306 30

local procedure GetResourceDetail(var Rec: Record "Sales Line"; var Resource: Record Resource)

307 31

begin

308 32

Rec.TBM_Department := Resource.TBM_Department;

309 33

end;

4210

4217

4218



DELIVERABLESITEMQUANTITY();

OnAfterAssignItemValues(Rec, Item, SalesHeader, xRec, CurrFieldNo);

end;



How to find the best event

```
4210 CheckPurchasingCodeForInvoice();
4211 if "Document Type" in ["Document Type"::Quote, "Document Type"::Order, "Document Type"::Invoice, "Document Type"::"Blanket Order"] then
4212     Validate("Purchasing Code", Item "Purchasing Code");
4213     OnAfterCopyFromItem(Rec, Item, CurrFieldNo, xRec);
```

onaftercopyfromitem

```
[EventSubscriber(ObjectType::Table, Database::"Sales Line", OnAfterCopyFromItem, '', false, false)]
```

0 references

```
local procedure "Sales Line_OnAfterCopyFromItem"(var SalesLine: Record "Sales Line"; Item: Record Item; CurrentFieldNo: Integer; xSalesLine: Record "Sales Line")
```

```
begin
```

```
    SalesLine.TBM_PackagingType := Item.TBM_PackagingType;
```

```
end;
```



How to find the best event

- What is the difference between an event, and a great event?
- Look at how many times an event gets triggered during an operation





Analysis 1 ▾



Filters ✕

Filter list by:

✕ Event Name

OnAfterValidateEvent

+ Filter...

Reset filters

| Object Name / Event Name / Element Name ▾ | Call Order | Event Type | ↓ Sum(Hi.. | Object Type |
|---|------------|------------|------------|-------------|
| ▾ Sales Line (52) | | | | |
| ▾ OnAfterValidateEvent (52) | | | | |
| > Outstanding Amount (7) | | | 7 | |
| > Unit Cost (LCY) (4) | | | 4 | |
| > Shipped Not Invoiced (4) | | | 4 | |
| > Inv. Disc. Amount to Invoice (4) | | | 4 | |
| > Document Type (4) | | | 4 | |
| > Document No. (4) | | | 4 | |
| > Unit Price (3) | | | 3 | |
| > Shipment Date (3) | | | 3 | |
| > Quantity (3) | | | 3 | |
| > Qty. to Assemble to Order (3) | | | 3 | |
| > Line Discount % (3) | | | 3 | |
| > Unit of Measure Code (2) | | | 2 | |
| > Qty. to Ship (2) | | | 2 | |
| > VAT Prod. Posting Group (1) | | | 1 | |
| > Purchasing Code (1) | | | 1 | |
| > No. (1) | | | 1 | |
| Sub Total OnAfterValidateEvent | | | 52 | |



Top 10 Events Chart (Most Subscribers) as of Fall, 2024

| Event | Object | Apps subscribing |
|---|-----------------------------------|------------------|
| OnAfterPostSalesDoc | Codeunit 80 "Sales-Post" | 5.6K |
| OnBeforePostSalesDoc | Codeunit 80 "Sales-Post" | 5.5K |
| OnAfterValidateEvent-Sell-to Customer No. | Table 36 "Sales Header" | 4K |
| OnAfterInitItemLedgEntry | Codeunit 22 "Item Jnl.-Post Line" | 3.9K |
| OnAfterSubstituteReport | Codeunit 44 ReportManagement | 3.8K |
| OnBeforePostPurchaseDoc | codeunit 90 "Purch.-Post" | 3.7K |
| OnAfterCopyGLEntryFromGenJnlLine | Table 17 "G/L Entry" | 3.2K |
| OnAfterPostPurchaseDoc | Codeunit 90 "Purch.-Post" | 3.11K |
| OnAfterValidateEvent-No. | Table 37 "Sales Line" | 3.1K |
| OnAfterCopyGenJnlLineFromSalesHeader | Table 81 "Gen. Journal Line" | 2.7K |



Control of Event Subscriptions

- EventSubscriberInstance Property
- Session.BindSubscription(Codeunit) Method
- Session.UnbindSubscription(Codeunit) Method



Control of Event Subscriptions

```
AL CustomerCardExt.PageExt.al > ...
5  pageextension 50100 CustomerCardExt extends "Customer Card"
7  actions











16  ..... 0 references
16  ..... action(OpenCustomerListAction)
17  ..... {
18  .....     ApplicationArea = All;
19  .....     Caption = 'Open Customer List';
20  .....     Image = Customer;
21  .....     0 references
21  .....     trigger OnAction()
22  .....     var
23  .....         HelloWorldExampleCodeunit: Codeunit HelloWorldExample;
24  .....     begin
25  .....         HelloWorldExampleCodeunit.OpenCustomerList();
26  .....     end;
27  ..... }
28  ..... 0 references
28  ..... action(OpenCustomerListBindingAction)
29  ..... {
30  .....     ApplicationArea = All;
31  .....     Caption = 'Open Customer List with Binding';
32  .....     Image = Customer;
33  .....     0 references
33  .....     trigger OnAction()
34  .....     var
35  .....         HelloWorldExampleCodeunit: Codeunit HelloWorldExample;
36  .....     begin
37  .....         HelloWorldExampleCodeunit.OpenCustomerListBinding();
38  .....     end;
39  ..... }
```



Control of Event Subscriptions

Customers











+ New Manage Home New Document Customer Prices & Discounts Report ...


| | | | |
|---|------------------------------|---|----------------------|
|  10000 Adatum Corporation Robert Townes | 4,677,926.34 4,670,047.78 |  20000 Trey Research Helen Ray | 4,918.08 4,918.08 |
|  30000 School of Fine Art Meagan Bond | 55,449.65 55,449.65 |  40000 Alpine Ski House Ian Deberry | 5,715.95 5,715.95 |
|  50000 Relecloud Jesse Homer | 8,836.80 8,836.80 |  C00010 | 0.00 0.00 |
|  C00020 | 0.00 0.00 |  C00030 | 0.00 0.00 |
|  C00040 Test | 0.00 0.00 |  C00050 Orlando Smith Team | 0.00 0.00 |

Without Binding

Customers

+ New Manage Home New Document Customer Prices & Discounts Report ...

| | | | |
|---|------------------------------|---|----------------------|
|  10000 Adatum Corporation Robert Townes | 4,677,926.34 4,670,047.78 |  20000 Trey Research Helen Ray | 4,918.08 4,918.08 |
|  30000 School of Fine Art Meagan Bond | 55,449.65 55,449.65 |  40000 Alpine Ski House Ian Deberry | 5,715.95 5,715.95 |
|  50000 Relecloud Jesse Homer | |  C00010 | 0.00 0.00 |
|  C00020 | |  C00030 | 0.00 0.00 |
|  C00040 Test | 0.00 0.00 |  C00050 Orlando Smith Team | 0.00 0.00 |

 Hello world

OK

WITH Binding

Practical Application and Data Flow Management



Event Subscription Best Practices

- Check if IsHandled
- Check for temporary records
- Distinguishing between Rec and xRec

```
[EventSubscriber(ObjectType::Table, Database::"Sales Header", 'OnAfterValidateEvent', 'Ship-to Code', false, false)]
```

```
0 references | 0% Coverage
```

```
local procedure SalesHeader_OnAfterValidateEvent_ShipToCode(var Rec: Record "Sales Header"; var xRec: Record "Sales Header"; CurrFieldNo: Integer)  
var
```

```
    IsHandled: Boolean;
```

```
begin
```

```
    if IsHandled then exit;
```

```
    if Rec.IsTemporary() then exit;
```

```
    if Rec."Ship-to Code" = xRec."Ship-to Code" then exit;
```

```
    if Rec."Ship-to Code" = '' then exit;
```



Event Subscription Best Practices

- Check if IsHandled
- Check for temporary records
- Distinguishing between Rec and xRec
- Running triggers
- Is your feature even enabled?

```
[EventSubscriber(ObjectType::Table, Database::"Sales Header", OnAfterInsertEvent, '', false, false)]
```

0 references | 0% Coverage

```
local procedure OnAfterInsertEvent_SalesHeader(var Rec: Record "Sales Header"; RunTrigger: Boolean)
```

```
var
```

```
    SalesLine: Record "Sales Line";
```

```
begin
```

```
    if not RunTrigger then exit;
```

```
    if Rec.IsTemporary() then exit;
```

```
    if not EssentialsCPGMgt.CheckFillandCancelSalesEnabled() then exit;
```

Event Subscription Best Practices

- Is your feature even enabled?

```
467 [EventSubscriber(ObjectType::Codeunit, Codeunit::"Sales-Post", 'OnAfterPostSalesDoc', '', false, false)]
468 local procedure SalesPostOnAfterPostSalesDoc(var SalesHeader: Record "Sales Header"; var GenJnlPostLine: Codeunit "Gen. Jnl.-Post Line";
469 > var ...
494 begin
495     if (SalesInvHdrNo <> '') and (not PreviewMode) then begin
496         SalesInvHeader.Get(SalesInvHdrNo);
497         PaymentMethod.Get(SalesInvHeader."Payment Method Code");
498         CustLedgerEntry2.Get(CustLedgerEntry."Entry No.");
499         CustLedgerEntry2."Payment Method Code" := PaymentMethod."Payment Method Code";
500         CustLedgerEntry2."Payment Method Code" := PaymentMethod."Payment Method Code";
501         CustLedgerEntry2.Modify(false);
502     end;
503
```

Examples

- Recreate Sales Lines
 - Header value changes, and lines need to get recreated

```
[EventSubscriber(ObjectType::Table, Database::"Sales Header", OnCreateSalesLineOnBeforeTransferFieldsFromTempSalesLine, '', false, false)]
0 references
local procedure Table_SalesHeader_OnBeforeTransferFieldsFromTempSalesLine(var SalesLine: Record "Sales Line";
    var TempSalesLine: Record "Sales Line" temporary; var SalesHeader: Record "Sales Header")
begin
    // Transfer Custom Fields
    SalesLine.TBM_PackagingType := TempSalesLine.TBM_PackagingType;
    SalesLine.TBM_Department := TempSalesLine.TBM_Department;
end;
```

Examples

- Data flow from
 - Item
 - Requisition Worksheet
 - Production Order



Examples

- Data flow from
 - Item

1.

```
[EventSubscriber(ObjectType::Table, Database::"Requisition Line", OnAfterCopyFromItem, '', false, false)]
```

0 references

```
local procedure Table_RequisitionLine_OnAfterCopyFromItem(var RequisitionLine: Record "Requisition Line"; Item: Record Item; CurrentFieldNo: Integer)
begin
    RequisitionLine.TBM_PackagingType := Item.TBM_PackagingType;
end;
```



Examples

- Data flow from
 - Item
 - Requisition Worksheet



```
[EventSubscriber(ObjectType::Codeunit, Codeunit::"Mfg. Carry Out Action", OnInsertProdOrderOnBeforeProdOrderInsert, '', false, false)]
```

0 references

```
local procedure Codeunit_MfgCarryOutAction_OnInsertProdOrderOnBeforeProdOrderInsert(var ProdOrder: Record "Production Order"; ReqLine: Record "Requisition Line")
```

```
begin
```

```
    ProdOrder.TBM_PackagingType := ReqLine.TBM_PackagingType;
```

```
end;
```



Examples

- Data flow from
 - Item
 - Requisition Worksheet
 - Production Order

3.

```
[EventSubscriber(ObjectType::Table, Database::"Production Order", OnBeforeAssignItemNo, '', false, false)]
```

0 references

```
local procedure ProductionOrderOnBeforeAssignItemNo(var ProdOrder: Record "Production Order"; xProdOrder: Record "Production Order"; var Item: Record Item; CallingFieldNo: Integer)
begin
    If ProdOrder.IsTemporary() then
        exit;
    if ProdOrder.TBM_PackagingType = xProdOrder.TBM_PackagingType then
        exit;
    ProdOrder.TBM_PackagingType := Item.TBM_PackagingType;
end;
```

Call to Action



Questions?



Thank you!

Please provide
feedback by taking
the session survey

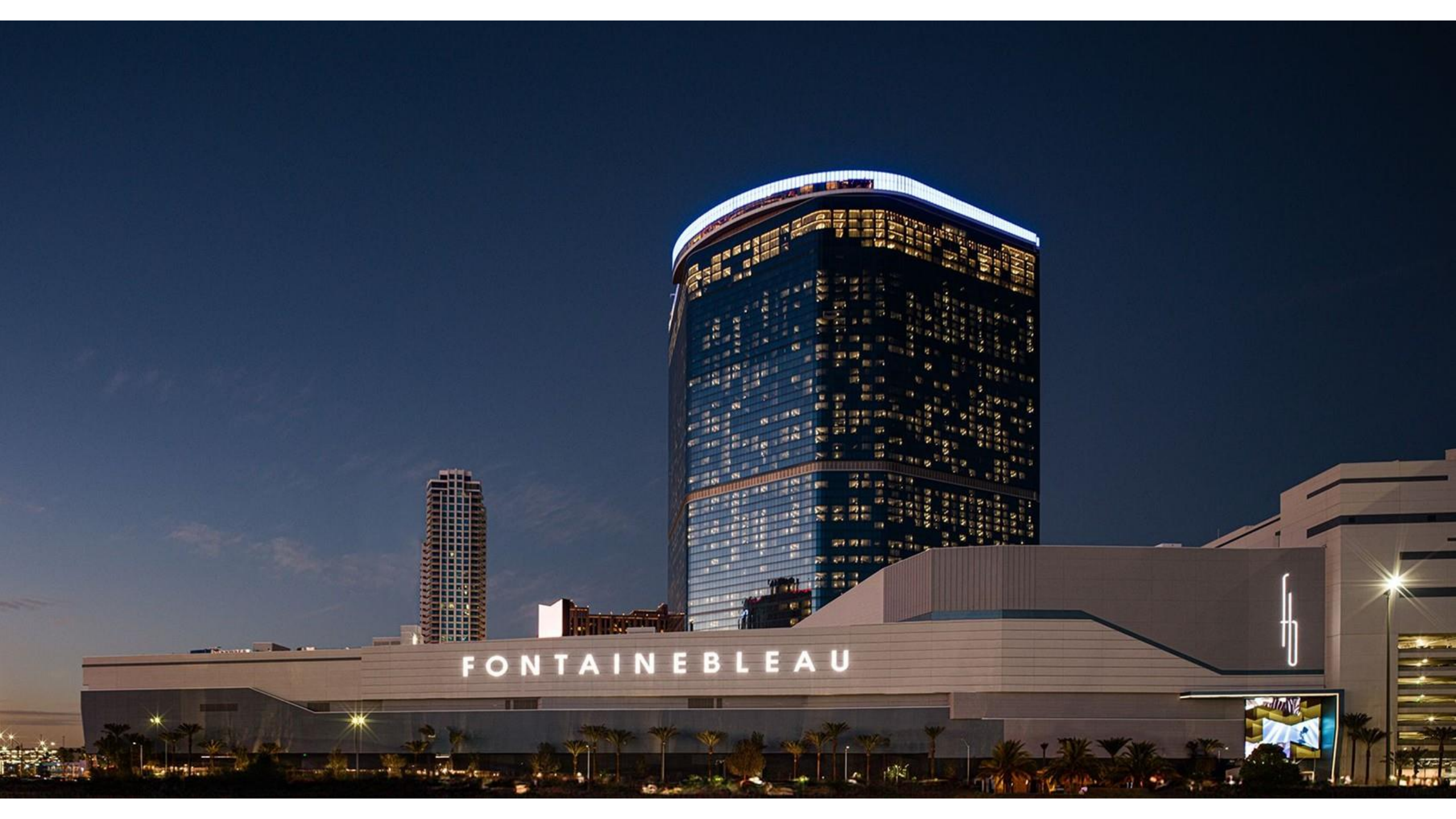




FONTAINEBLEAU



THANK YOU



FONTAINEBLEAU

fb

Tonya Bricco-Meske

Manager,
RSM US LLP
Central Wisconsin



Time zone: Central

e: tonya.bricco-meske@rsmus.com

GitHub: <https://github.com/tonyabriccomeske>

Blog: <https://bcdevnotebook.com/>

Bluesky: @tonyabm



Henrik Helgesen

President,
Helgesen Consulting, Inc.
Los Angeles, CA



Time zone: Pacific

e: henrik@helgesenconsulting.com

p: +1 818 932-1240

GitHub: <https://github.com/TheDoubleH>

Blog: <https://thedoubleh.dev>

Bluesky: @thedoubleh.dev

<https://go.thedoubleh.dev>



2026 DYNAMICS CON

