



COSMO CONSULT

Business-Software für Menschen

Microsoft AL

GUI & In-Client Designer

Prerequisite

Use Workshop-VM / Workshop-Computer
follow the instructions of the Boarding Pass



Visual Studio Code

<https://code.visualstudio.com>



Competence Team DevTech

[Microsoft AL](#)

VS Code

Useful Extensions

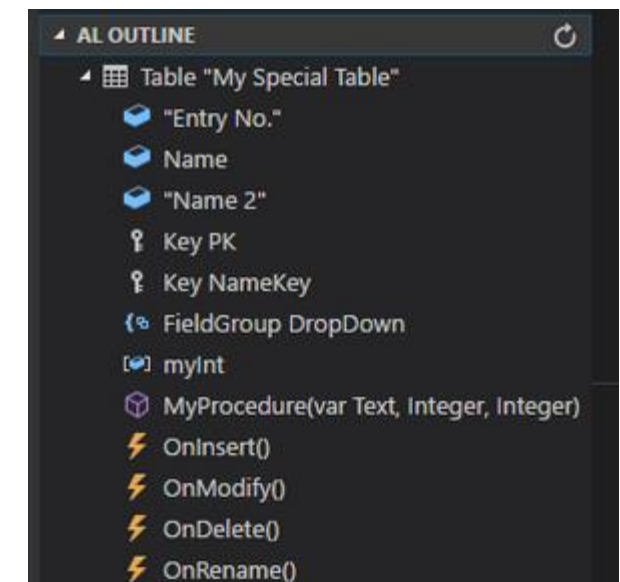
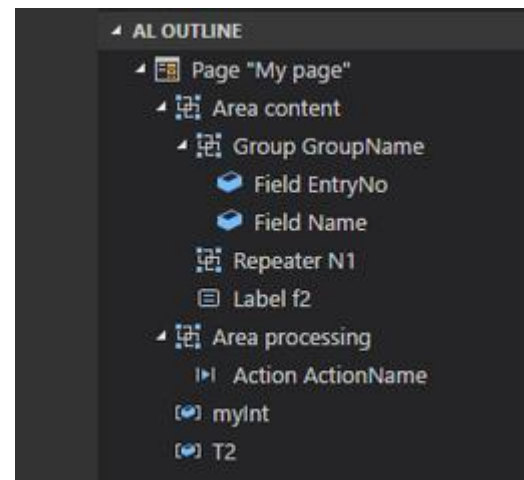
> AL Formatter

- > Indentation
- > Keyword case style
- > Sort variable definitions
- > Readability Guidelines - Spacing and newlines (experimental)

> AL Code Outline

- > Outlining – AL Object Structure
- > “Create List Page” from Table
- > “Create Card Page” from Table

```
Codeunit 70000000 MyCodeunit
{
    var
        Customer : Record Customer;
        Counter : Integer;
        InvalidCustomerErr : TextConst ENU='Invalid Customer.
    trigger OnRun();
    begin
        if Customer.FindSet then
            repeat
                if not Customer.HasAddress then
                    Error(InvalidCustomerErr);
            until Customer.Next = 0;
    end;
}
```



GUI & In-Client Designer

Agenda

- 1 Language & Types
- 2 In-Client Designer
- 3 Pages in Microsoft AL
- 4 Notification
- 5 Wizard & Assisted Setup



Microsoft AL

Language & Types

- Page
- Page Extension
- Page Customization

Microsoft AL

Language & Types

- Recommendation for VS Code
 - 1 File per Object
 - Organize Object Files in Folders (Pages, PageExtension, ...)
- Use Snippets in AL
 - ttable
 - tpageext
 - tcodeunit
 - ...
- Documentation - Development in AL
 - <https://docs.microsoft.com/.../developer/devenv-dev-overview>

C/Side	Microsoft AL	
	<i>Extending</i>	<i>New</i>
Codeunit	-	<u>codeunit</u>
Table	<u>tableextension</u>	<u>table</u>
Page	<u>pageextension</u> <u>pagecustomization</u>	<u>page</u>
XMLport	-	<u>xmlport</u>
Query	-	<u>query</u>
Report	-	<u>report</u>
Menu Suite	-	-
		<u>profile</u>
		<u>controladdin</u>

Microsoft AL

Page

page <ID> „<Unique Name>“

➤ Content Structure

- Page Properties
- Layout Section
- Actions Section
- Global Variables
- Procedures & Events & Page-Triggers

➤ Sections

- Layout → Areas → Groups → *Group-Properties* & Fields → *Field-Properties* & Triggers
- Action → Areas → *Area-Properties* & Actions → *Action-Properties* & Trigger

```
page 58215 "Test Page ccAI"
{
    PageType = Card;
    UsageCategory = Administration;
    ApplicationArea = All;
    Caption = 'cc TEST';

    layout
    {
        0 references
        area(content)
        {
            0 references
            group(GroupName)
            {
            }
        }
    }

    actions
    {
        0 references
        area(processing)
        {
            0 references
            action(ActionName)
            {
                Caption = 'Test';
                Image = TestDatabase;
                ApplicationArea = All;

                trigger OnAction();
                var CU: Codeunit "Intent Handler Test ccAI";
                begin
                    CU.TestStopWorkcenter;
                end;
            }
        }
    }
}
```

Microsoft AL

Page Extension

pageextension <ID> "<Unique Name>" extends "< Target Page Name>" / or <ID>

➤ Extensions / Modifications

- Add *addfrist(<ID>), addbefore(<ID>), addafter(<ID>), addlast(<ID>)*
- Move *movefrist(<ID>), movebefore(<ID>), moveafter(<ID>), mocelast(<ID>)*
- Modify *modify(<ID>)*

➤ Changes in Sections

- Layout → <Extension/Modification> → new / changed page content
- Action → <Extension/Modification> → new / changed page actions

➤ Additional Content

- Global Variables
- Procedures & Events & Page-Triggers

```
pageextension 58205 "Work Center List ccAI" extends "Work Center List"
{
    layout
    {
        addafter(Name)
        {
            0 references
            field(Status; Status)
            { StyleExpr = StatusStyle; }

            0 references
            field("Fault Information"; "Status Message")
            { StyleExpr = StatusStyle; }
        }

        addfirst(FactBoxes)
        {
            2 references
            part("Bot"; "Bot ccAI")
            { }
        }
    }

    actions
    {
        addlast(Processing)
        {
            0 references
            action("Stop Work Center")
            {
                CaptionML = DEU = 'Stop', ENU = 'Stop';
                Image = Stop;
                Promoted = true;
                PromotedIsBig = true;
                ApplicationArea = All;

                trigger OnAction();
                begin
                    Rec.Validate(Status, Status::Down);
                    Rec.Modify;
                end;
            }
        }
    }

    5 references
    var StatusStyle: Text;

    trigger OnAfterGetRecord();
    begin
```


Microsoft AL

Page Customization

pagecustomization "<Unique Name>" customizes "<Target Page Name>" / or <ID>

➤ Modifications

- Move *movefirst(<ID>), movebefore(<ID>), moveafter(<ID>), moveafter(<ID>)*
- Modify *modify(<ID>)*

➤ Changes in Sections

- Layout → <Modification> → changed page content
- Action → <Modification> → changed page actions

➤ No Additional Content!!!

```
pagecustomization "Item List Cust" customizes "Item List"
{
    layout
    {
        // Modify fields / groups
        modify("No.")
        {
            Visible = false;
        }

        // Move Fields / Groups relative to an Anchor
        movefirst(Content; Description, "Search Description")
    }

    actions
    {
        // Add changes to page actions here
        modify(Action29)
        {
            Visible = false;
        }
    }

    // Variables, procedures and triggers are
    // not allowed on Page Customizations
}
```

In-Client Designer

WSIWYG Designer (Web Client)

- Page Customization
- Page Extension



In-Client Designer

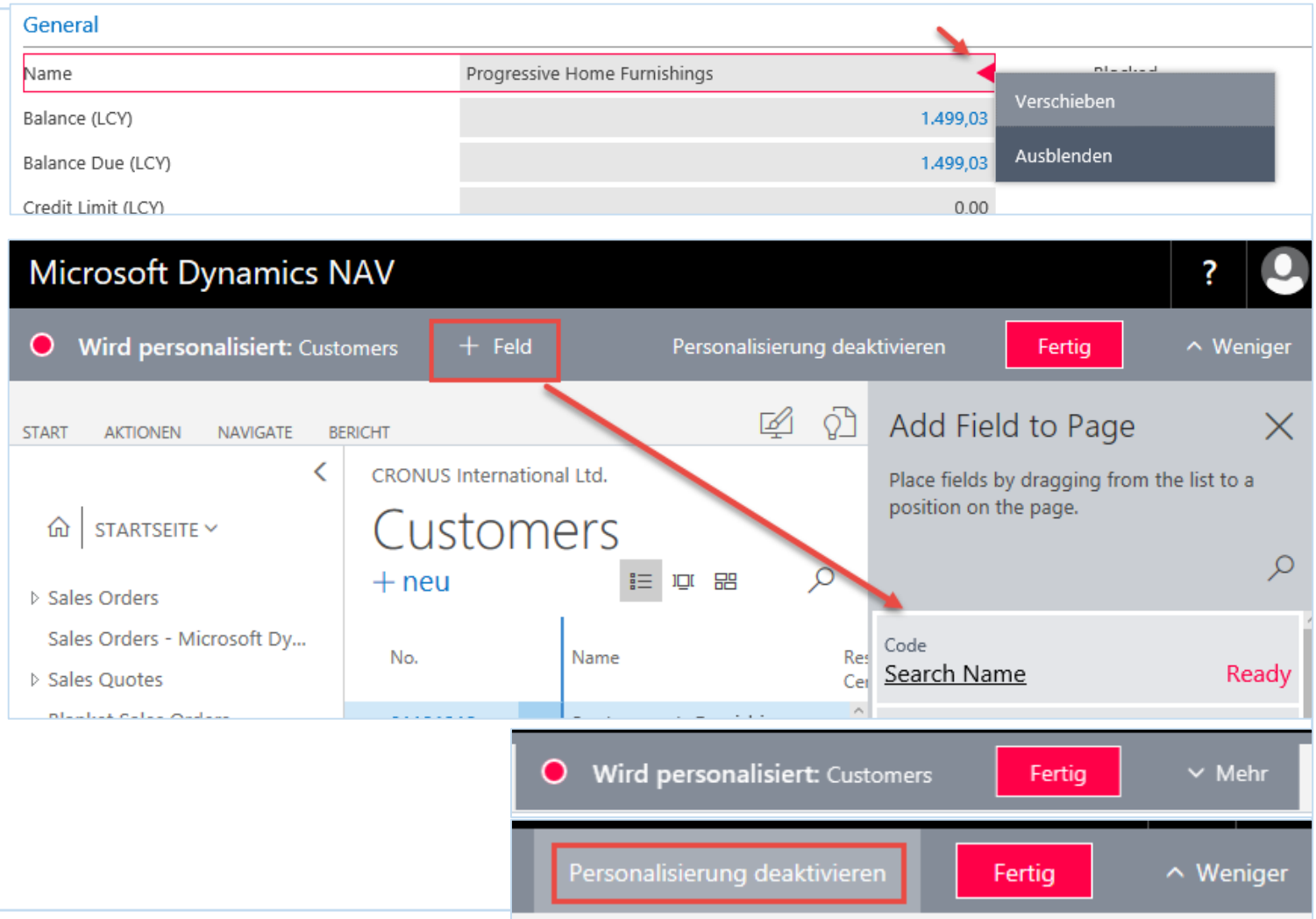
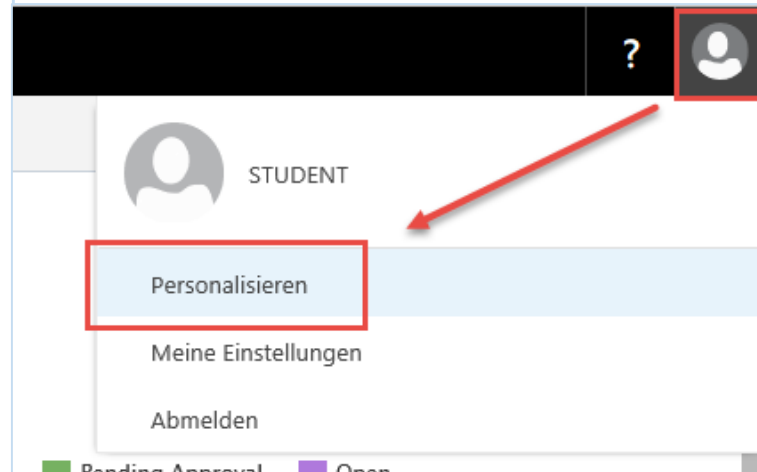
Page Customization

Customization

- Individual per User
- On-Top of all modifications
- User-Menu → Personalization

Actions

- Show / Hide Fields
- Move Fields



In-Client Designer

Page Extension

➤ Extension

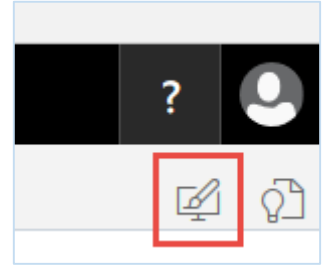
- For all Users
- On-Top of all **existing** modifications *except Customizations*
- Designer-Menu

➤ Designer Actions

- Rename Groups (e.g. Card Pages)
- Add, Delete, Move Fields

➤ Finish Up

- Download Extension package (zip)



Gestalten: Customer Card **Gestalten einstellen** **Mehr**

GeneralXXXX **Rename Groups**

Name	Progressive Home Furnish...	Placed
Balance (LCY)	1.499,03	Löschen

Add Field to Page **X**

Place fields by dragging from the list to a position on the page.

Code	No.	Placed
Text	Name	Placed
Code	Search Name	Ready

Finish Up Design **X**

Gestalten einstellen **Weniger**

Extension Name

Publisher

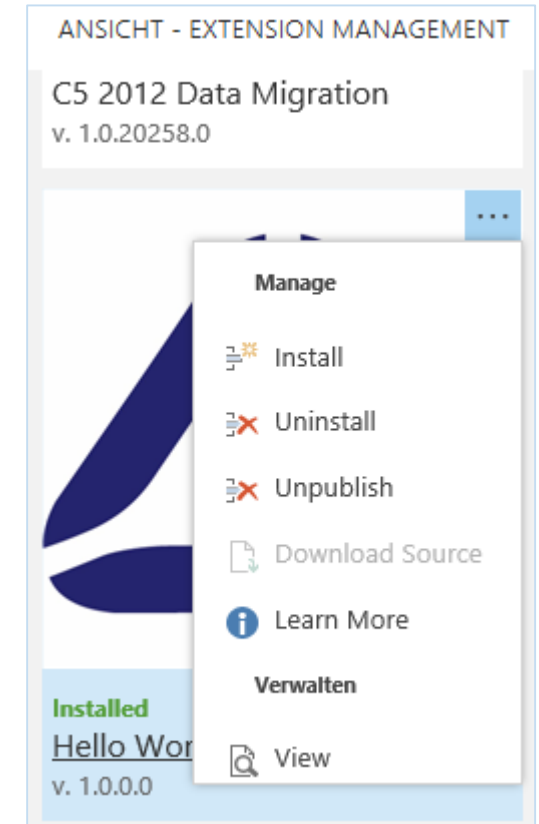
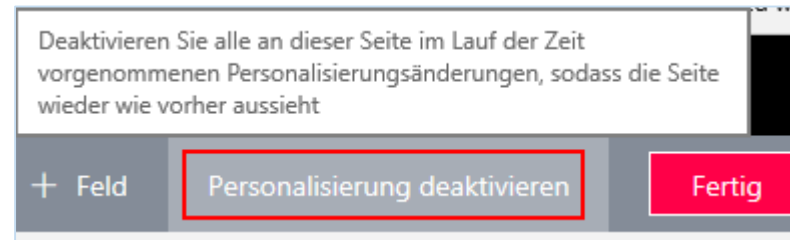
Download Code

☐

In-Client Designer

Extension / Customization Management

- Remove an Extension
 - Page “Extension Management”
 - Uninstall → “Deactivate”
 - Unpublish → Remove from System
- Remove a Customization
 - Open Personalization → Deactivate the Personalization



In-Client Designer

Hands On

- Open the Web Client
- Personalize the „Item Card“
 - Hide some Fields
 - Rearrange some Fields
 - Save the Customization
- Create an Extension for the Item Card
 - Unlock Designer when necessary
 - Rename the Item Group
 - Rearrange some Fields / Factoboxes
 - Add Fields if you want
 - Finish the Extension and Download the Package

1000 · Bicycle

Item

No. 1000 > Description

Inventory

Shelf No. F4 Inventory

Costs & Posting > 350

1000 · Bicycle

My unlimited Item Information

Description Bicycle Item Category...

Base Unit of Me... PCS Blocked ☐

My Inventory

Shelf No. F4 Otv. on Compo...

Gestalten: Item Card

Sie haben die Seite „Item Card“ zuvor personalisiert. Wenn Sie diese Seite erneut ohne die Personalisierungsänderungen laden möchten, wählen Sie die Seite „Entsperren“ aus.

Seite „Entsperren“

Finish Up Design

Extension Name

My Extension

Publisher

COSMO COSNULT

Download Code

☒

Wie möchten Sie mit My Extension_COSMO COSNULT_1.0.0.0.zip verfahren?

Von: dedrs-vscode1.westeurope.cloudapp.azure.com

Öffnen | Speichern | ^

Pages in Microsoft AL

Microsoft AL Page Design

- Page Properties
- Layout, Action, Triggers



Microsoft AL

Page Properties

➤ PageType

➤ *Card, List, RoleCenter, Card Part, List Part, **API**, ...*

➤ SourceTable / SourceTableView

➤ *SourceTableView = order(ascending) where (Code = const('PROD'));*

➤ ApplicationArea *Application Area for the Page WebClient SearchPage*

➤ *All, Basic, Suite, Advanced*

➤ UsageCategory *Department for the Page WebClient SearchPage*

➤ *None, Lists, Tasks, ReportsAndAnalysis, Documents, History, Administration*

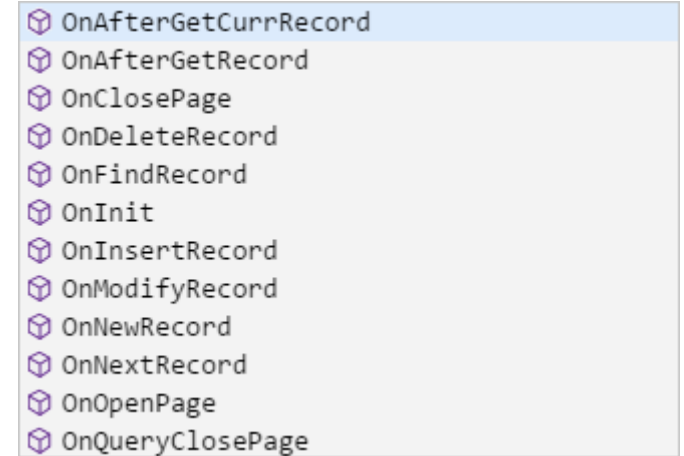
➤ CardPageId

Microsoft AL

Page / Page Extension - Triggers

➤ Page - Triggers *(Picture 1)*

- Same Triggers as in C/Side

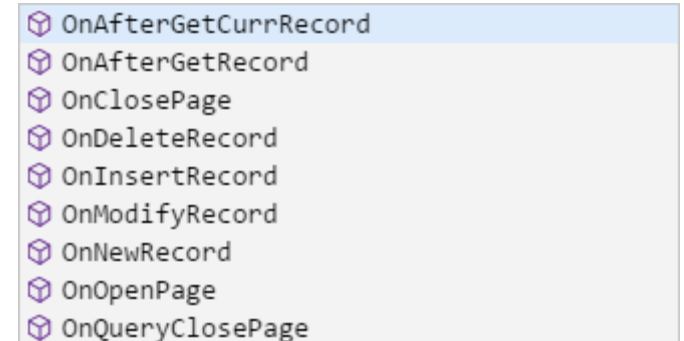


A screenshot of a list of triggers for a Page object in Microsoft AL. The list is displayed in a light gray box with a blue header bar. The triggers are: OnAfterGetCurrRecord, OnAfterGetRecord, OnClosePage, OnDeleteRecord, OnFindRecord, OnInit, OnInsertRecord, OnModifyRecord, OnNewRecord, OnNextRecord, OnOpenPage, and OnQueryClosePage. The first trigger, OnAfterGetCurrRecord, is highlighted in blue.

- OnAfterGetCurrRecord
- OnAfterGetRecord
- OnClosePage
- OnDeleteRecord
- OnFindRecord
- OnInit
- OnInsertRecord
- OnModifyRecord
- OnNewRecord
- OnNextRecord
- OnOpenPage
- OnQueryClosePage

➤ Page Extension - Triggers *(Picture 2)*

- less Triggers as Pages
- Microsoft: "...triggers are called "closer" to the original trigger than the events..."



A screenshot of a list of triggers for a Page Extension object in Microsoft AL. The list is displayed in a light gray box with a blue header bar. The triggers are: OnAfterGetCurrRecord, OnAfterGetRecord, OnClosePage, OnDeleteRecord, OnInsertRecord, OnModifyRecord, OnNewRecord, OnOpenPage, and OnQueryClosePage. The first trigger, OnAfterGetCurrRecord, is highlighted in blue.

- OnAfterGetCurrRecord
- OnAfterGetRecord
- OnClosePage
- OnDeleteRecord
- OnInsertRecord
- OnModifyRecord
- OnNewRecord
- OnOpenPage
- OnQueryClosePage

Microsoft AL

Page Actions

➤ Properties *equal to C/Side*

- Image
- Promoted / PromotedOnly / PromotedIsBig / PromotedCategory
- Ellipsis
- ApplicationArea

All, Basic, Suite, Advanced

- InFooterBar *(Wizards)*
- RunObject & RunPageView

RunObject = page "Target Page"

RunPageView = order(ascending) where("Target Rec Field" = field("Source Field"));

➤ Trigger

- OnAction();

```
actions
{
    area(Navigation)
    {
        action(NewAction)
        {
            RunObject = codeunit "Document Totals";
        }
    }
}
```

```
group(MyActionGroup)
{
    Action(MyAction1)
    {
        CaptionML = ENU='Hello!';

        trigger OnAction();
        begin
            Message('My message');
        end;
    }

    Action(MyAction2)
    {
        RunObject = codeunit "Activities Mgt.";
    }
}
```


It's Lab Time

Hands On

- New Table & Table Extension
- New Page & Page Extension



Pages in Microsoft AL

Hands On

➤ Material

- Project Skeleton „02-GUI-&-In-Client-Designer**Starter**\02 My Favorit Beer\”
- Tables: “Beer Category”
- Tables.Extensions: “Cutomer”, “Item”

➤ Prerequisites

- Connect to your Workshop VM → use Instructions on your Workshop Boarding Pass

➤ Hand On - Tasks

- Create a **Card Page** and a **List Page** for Table “Beer Category”
- Create a **Page Extension** (Card & List) for Table Extension “Customer”
- Create a **Page Extension** (Card & List) for Table Extension “Item”

Pages in Microsoft AL

Hands On – Card Page

➤ Create a Card Page „*Beer Category Card [CC]*“ ID: 50201

➤ Setup the Properties for the Page

- PageType *Card*
- UsageCategory *Documents*
- ApplicationArea *All*
- SourceTable *„Beer Category [CC]“*
- SourceTableView *sorting: „Code“ order: ascending*
- CaptionML *ENU + DEU*

The screenshot displays the Microsoft Dynamics 365 AL development environment. At the top, there is a 'START' menu with icons for 'Neu' (New), 'Bearbeiten' (Edit), 'Löschen' (Delete), 'Notizen' (Notes), 'Links', 'Vorheriger' (Previous), 'Nächster' (Next), 'Verwalten' (Manage), and 'Dateianhang anzeigen' (Show Attachments). Below the menu, the page title is 'ANSICHT - BIER KATEGORIE KARTE'. The main content area shows the title 'AMBER ALE' and a 'General' tab. Under the 'General' tab, there is a table with two columns: 'Code' and 'Beschreibung'. The 'Code' column contains the value 'AMBER ALE' and the 'Name' column contains the value 'Amber ale'. The 'Beschreibung' column contains the value 'American-Style Amber/Red Ale'.

General	
Code	AMBER ALE
Name	Amber ale
Beschreibung	American-Style Amber/Red Ale

➤ Layout

- Add all fields of the table to the layout *area „content“* in a *group „General“*
- Setup field properties (e.g. ApplicationArea)

Pages in Microsoft AL

Hands On – List Page

➤ Create a List Page „*Beer Category List [CC]*“ ID: 50202


➤ Setup the Properties for the Page

- PageType *List*
- UsageCategory *Lists*
- ApplicationArea *All*
- SourceTable *„Beer Category [CC]“*
- SourceTableView *sorting: „Code“ order: ascending*
- CaptionML *ENU + DEU*
- CardPageId *"Beer Category Card [CC]"*


➤ Layout

- Add all fields of the table to the layout *area „content“* in a *repeater „General“*
- Setup field properties (e.g. ApplicationArea)


START




Neu




Bearbeiten




Ansicht




Löschen




Liste bearbeiten



Notizen



Links



In Excel öffnen

Neu

Verwalten

Dateianhang anzeigen

Seite

ANSICHT - BIER KATEGORIE LISTE + Neu

	CODE	NAME	BESCHREIBUNG
	ALTBIER	...	Altbeer
	AMBER ALE	Amber ale	German-Style Altbeer
	BARLEY WINE	Barley wine	American-Style Amber/Red Ale
	BERLINER WEISSE	Berliner Weisse	British-Style Barley Wine Ale
	BITTER	Bitter	Berliner-Style Weisse
	BLONDE ALE	Blonde Ale	Ordinary Bitter
	BROWN ALE	Brown Ale	Golden or Blonde Ale
	DOPPELBOCK	Doppelbock	English-Style Brown Ale
	DUNKELWEIZEN	Dunkelweizen	German-Style Doppelbock
	EISBOCK	Eisbock	South German-Style Dunkel Weizen
	HEFEWEIZEN	Hefeweizen	German-Style Eisbock
			South German-Style Hefeweizen

Pages in Microsoft AL

Hands On – Page Extension

➤ Create a Page Extension „*Item Card [CC]*“ extends „*Item Card*“

ID: 50200

➤ Layout

- Add a group “Beer” behind the “Item” group
- CaptionML ENU + DEU
- Add the “Beer” fields
- Hide “Beer Category Group” & „Beer Category Group Name“, when the item is not marked as „Beer Item“

➤ Actions

- Add an action to show the “Beer Category Card”
- Use Properties:
CaptionML, ApplicationArea, Visible, RunObject, RunObjectView, Promoted, Image

70061 · Freiberger Pils

Item Mehr anzeigen

Description	Freiberger Pils	Base Unit of Meas...	PCS	...
Blocked	<input type="checkbox"/>	Item Category Code		...
Type	Inventory			

Bier Informationen

Beer Item	<input checked="" type="checkbox"/>	Bier Kategorie Na...	Blonde Ale
Bier Kategorie Code	BLONDE ALE		...



BLONDE ALE

General

Code	BLONDE ALE
Name	Blonde Ale

Pages in Microsoft AL

Hands On – Page Extension

➤ Create a Page Extension „*Customer Card [CC]*“ extends “*Customer Card*” ID: 50201

➤ Layout

- Add a group “Favorite Beer” behind the “General” group
- CaptionML ENU + DEU
- Add the “Beer” fields from record “Customer”

➤ Actions

- Add an action to show the “Beer Category Card”
- Add an action to show the possible “Beer Items”

BEARBEITEN - CUSTOMER CARD

10000 · The Cannon Group PLC


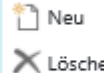




General Mehr anzeigen

Name	The Cannon Group PLC	Blocked	<input type="checkbox"/>
Balance (LCY)	168.364,41	Total Sales	17.100,96
Balance Due (LCY)	168.364,41	Costs (LCY)	11.762,70
Credit Limit (LCY)	0,00		

Lieblingsbier

Bier Kategorie Code	BLONDE ALE	...	Lieblings-Bier Code	70061	...
Bier Kategorie Na...	BLONDE ALE		Lieblings-Bier Na...	Freiberger Pils	

START AKTIONEN NAVIGATE BERICHT

 Bearbeiten	 Neu	 Löschen	 Bier Kategorie	 Biere	 Sales Quot
Verwalten			Process		

Improve the UX

Features

- Notification
- Assisted Setups (Wizard)



Pages in Microsoft AL

Improve the UX

> Notifictation

“... programmatic way to send non-intrusive information to the User Interface (UI)...”

- > ID *Guid*
- > Message
- > Scope *Local / Global*
- > Actions
- > GetData(...) / SetData(...)

BEARBEITEN - CUSTOMER CARD

X Bier-Kategorien sind nicht initialisiert Einrichten

10000 · The Cannon Group PLC

General Mehr anzeigen

Name	The Cannon Group PLC	Blocked	
Balance (LCY)	168.364,41	Total Sales	17.100,96
Balance Due (LCY)	168.364,41	Costs (LCY)	11.762,70
Credit Limit (LCY)	0,00		

> Configurable

- > Record "My Notifications"
 - IsEnabled(...) / IsEnabledForRecrod(...)

BEARBEITEN - MY NOTIFICATIONS

NOTIFICATION	ENABL...	CONDITIONS
Confirm marking booking as invoiced.	...	<input checked="" type="checkbox"/>
Update Sell-to Customer Address	...	<input checked="" type="checkbox"/>
Show a suggestion to create contacts for newly create...	...	<input checked="" type="checkbox"/>
Update Pay-to Vendor Address	...	<input checked="" type="checkbox"/>
Item availability is low.	...	<input checked="" type="checkbox"/> (View filter details)
Customer exceeds credit limit.	...	<input checked="" type="checkbox"/> (View filter details)
Data migration notification	...	<input checked="" type="checkbox"/>

Pages in Microsoft AL

Improve the UX

> Wizards

- > Object Type *Page (Navigation)*
- > Groups *InstructionalText*

> Actions

- > Back, Next, Finish *Visibility depends on Page*
- > Properties *InFooterBar*

> Groups = Wizard Pages *Show / Hide for each Page*

- > Start Page / Welcome *Introduktion*
- > Page XYZ
- > Finish Page *Summary*

DYNAMICS AI EINRICHTUNGS ASSISTENT

Konfigurieren Sie die Spracheinstellungen für Dynamics AI

Speech-API Schlüssel

●●●●●●●●

Sprach Code

DE-DE-MALE

...

Stimme Id

Microsoft Server Speech Text to Speech Voi...

Stimme

Achim

Zurück

Weiter

Pages in Microsoft AL

Improve the UX

➤ Assisted Setup

- Registration of individual Setups

➤ Register a Page / Wizard

- Record *"Aggregated Assisted Setup"*
- Event *"OnRegisterAssistedSetup"*

```
[EventSubscriber(ObjectType::Table, Database::"Aggregated Assisted Setup", 'OnRegisterAssistedSetup', '', true)
0 references
local procedure RegisterAssistedSetup(var TempAggregatedAssistedSetup: Record "Aggregated Assisted Setup")
var tCaption: TextConst DEU = 'Dynamics AI Einrichtung', ENU = 'Dynamics AI Setup';
    Setup: Record "Setup ccAI";
begin
    TempAggregatedAssistedSetup.Init();
    TempAggregatedAssistedSetup."Record ID" := Setup.RecordId();
    TempAggregatedAssistedSetup."Page ID" := Page::"Setup ccAI";
    TempAggregatedAssistedSetup."Assisted Setup Page ID" := Page::"Setup Wizard ccAI";
    TempAggregatedAssistedSetup."External Assisted Setup" := true;
    TempAggregatedAssistedSetup."Item Type" := TempAggregatedAssistedSetup."Item Type"::"Setup and Help";
    TempAggregatedAssistedSetup.Visible := true;
    TempAggregatedAssistedSetup.Status := TempAggregatedAssistedSetup.Status::"Not Completed";
    TempAggregatedAssistedSetup.Name := tCaption;

    TempAggregatedAssistedSetup.Insert();
end;
```

ANSICHT - UNTERSTÜTZTES SETUP	
NAME	STATUS
Cashflowplanung einrichten	Abgeschlossen
Genehmigungsworkflows einrichten	Nicht abgeschlossen
Debitorengenehmigungsworkflow einrichten	Nicht abgeschlossen
E-Mail einrichten	Nicht abgeschlossen
E-Mail-Protokollierung einrichten	Nicht abgeschlossen
Ihren Unternehmensposteingang in Outlook einrichten	Nicht abgeschlossen
Berichtsdaten einrichten	Nicht abgeschlossen
Artikelgenehmigungsworkflow einrichten	Nicht abgeschlossen
Zahlungsgenehmigungsworkflow einrichten	Nicht abgeschlossen
Dynamics 365 for Sales-Verbindung einrichten	Nicht abgeschlossen
MwSt. einrichten	Nicht abgeschlossen
Externen Buchhalter einladen	Nicht abgeschlossen
Konsolidierungsberichte einrichten	Nicht abgeschlossen
Dynamics AI Einrichtung	...

It's Lab Time

Hands On

- Notification
- Wizard

Pages in Microsoft AL

Hands On

➤ Material

- Project Skeleton *„02-GUI-&-In-Client-Designer\Starter\02 My Favorit Beer\”*
- Tables: *“Beer Category”*
- Tables.Extensions: *“Cutomer”, “Item”*

➤ Assets

- *“BeerCategories.json”*

➤ Prerequisites

- Connect to your Workshop VM → use Instructions on your Workshop Boarding Pass

➤ Hand On - Tasks

- Invoke a **Notification**, when the selected Beer does not match the Customers Favorite Beer Category
- Create a **Setup Wizard** to initialize Beer Categories and use a **List Part** for Table *“Beer Category”*
- Invoke a **Notification**, when Beer Categories are empty and start the Wizard by a Notification Action

Pages in Microsoft AL

Hands On – Notification

- Create a Code Unit „*Beer Mgt. [CC]*“ ID: 50201
- Add 2 Methods
 - IsBeerMatch ... check the correctness of the beer
 - CheckBeer ... use the check and show a notification
- Use the codeunit from table extension “Customer [CC]” to check and invoke a notification

```
2 references
field(50200; "Beer Category Code [CC]"; Code[20])
{
    CaptionML      = ENU = 'Beer Category Code', DEU = 'Bier Kategorie Code';
    Description    = '[Dev] GUI & In-Client Designer';
    TableRelation  = "Beer Category [CC]".Code;
    DataClassification = CustomerContent;

    trigger OnValidate();
    var beerMgt: Codeunit "Beer Mgt. [CC]";
    begin
        CalcFields("Beer Category Name [CC]");

        beerMgt.CheckBeer(Rec);
    end;
}
```

```
procedure IsBeerMatch(customer: Record Customer) : Boolean;
var beerItems: Record Item;
begin
    beerItems.SetRange("Beer Item [CC]", true);
    beerItems.SetRange("Beer Category Code [CC]", customer."Beer Category Code [CC]");

    exit(not beerItems.IsEmpty());
end;

// Create a GUID with PowerShell > [System.Guid]::NewGuid()
1 reference
var cBeerMatchID : TextConst ENU = '30546dde-20ae-458e-9c12-c25e4491c3dd';
1 reference
procedure CheckBeer(customer: Record Customer): Boolean;
var notification: Notification;
    beerMatch: Boolean;
    id: GUID;
    msg: TextConst ENU = 'The selected beer does not belong to the preferred category',
        DEU = 'Das Bier gehört nicht zur bevorzugten Kategorie';
    myNotifications: Record "My Notifications";
begin
    id := FORMAT(cBeerMatchID);
    beerMatch := IsBeerMatch(customer);

    if not myNotifications.IsEnabledForRecord(id, customer) then exit(beerMatch);

    notification.Scope := Notification.Scope::LocalScope;
    notification.Id := id;

    if beerMatch then
    begin
        notification.Recall();
    end else begin
        notification.Message := msg;
        notification.Send();
    end;

    exit(beerMatch);
end;
```

Pages in Microsoft AL

Hands On – Wizard

- Create a Page „*Beer Wizard [CC]*“ *ID: 50200*
- Add 1 Variable (Option) *CurrentPage "Welcome", "Beer Categories", "Finish"*
- Add 3 Variables (Boolean) *WelcomeVisible, BeerCategoriesVisible, FinishVisible*
- Add 3 Pages (Groups) *"Welcome", "Beer Categories", "Finish"* *Caption, InstructionalText, Visible ...*
- Add 3 Actions *Back, Next, Finish*
- Manage the visible Page by this method and check the result

```
3 references
local procedure UpdateView(newPage: Integer)
begin
    if newPage > CurrentPage::Finish then newPage := CurrentPage::Finish;
    if newPage < CurrentPage::Welcome then newPage := CurrentPage::Welcome;

    // Change the Page & Update the visibility
    CurrentPage          := newPage;
    WelcomeVisible       := (CurrentPage = CurrentPage::Welcome);
    BeerCategoriesVisible := (CurrentPage = CurrentPage::"Beer Categories");
    FinishVisible        := (CurrentPage = CurrentPage::Finish) and (ImportRecordCount > 0);
end;
```

Pages in Microsoft AL

Hands On – Wizard

➤ Page "Welcome"

- Add and Initialize a Multiline Textfield with a Json content
- Add an Assisted Edit trigger to the field, which **import** (read) the **Json content** from File into the variable

```
local procedure ImportJsonFile()
var windowTitle: TextConst ENU = 'Import Beer Categories',
    DEU = 'Import Bier Kategorien';
fileFilter: TextConst ENU = 'All Files (*.*)|*.*|Json Files (*.json)|*.json',
    DEU = 'Alle Dateien (*.*)|*.*|Json Dateien (*.json)|*.json';

name: Text;
data: TextBuilder;
line: Text;
stm: InStream;
begin
    // Import the local file content
    if not UploadIntoStream(windowTitle, '', fileFilter, name, stm) then exit;

    // Read all data from file
    while stm.ReadText(line) > 0 do
        begin
            data.Append(line);
        end;

    // Assign the imported text to the Field
    BeerJson := data.ToText();
    CurrPage.Update();
end;
```

BIER KATEGORIEN EINRICHTUNG

Einrichtung der Bier Kategorien\ (Sie können die Datei "...\\Assets\\BeerCategories.json" importieren)

BIER KATEGORIEN - JSON

{"Beer Categories":[{"Name":"Altbier","Description": "German-Style Altbier"}, {"Name":"Amber ale","Description": "American-Style Amber/Red Ale"}]}

Weiter

Pages in Microsoft AL

Hands On – Wizard

➤ Page “Beer Categories”

- Add a List Part “*Beer Category Sub [CC]*” ID 50203 which allow the selection of categories
- Add a temp. Record “*Beer Category[CC]*” to the wizard
- Read the Json Content into the record during page change
“Welcome” → “Beer Categories”
- Assign the Record to the *List Part*

```
local procedure ImportBeerCategoriesFromJson(): Boolean
var jObject: JsonObject;
    jToken: JsonToken;
    jItem: JsonToken;
begin
    if not jObject.ReadFrom(BeerJson) then exit(false);
    if not jObject.Get('Beer Categories', jToken) then exit(false);
    if not jToken.IsArray() then exit(false);

    // Clear the temp-record
    tmpBeerCategories.DeleteAll();

    // Import all items in the array 'Beer Categories' into the tmp.-record
    foreach jItem in jToken.AsArray() do
        if jItem.IsObject() then
            begin
                ImportBeerCategoryFromJson(tmpBeerCategories, jItem.AsObject());
            end;
        end;

    // assign the temp-record to the list part
    CurrPage.BeerCategories.Page.AssignRecord(tmpBeerCategories);
    exit(true);
end;
```

```
local procedure ImportBeerCategoryFromJson(var beerCategory: Record "Beer Category [CC]"; jData: JsonObject)
var jToken: JsonToken;
begin
    beerCategory.Init();

    // read the Json information {"Name": "...", "Description": "..."} from jData
    if jData.Get('Name', jToken) then
        if jToken.IsValue() then
            beerCategory.Name := jToken.AsValue().AsText();
    if jData.Get('Description', jToken) then
        if jToken.IsValue() then
            beerCategory.Description := jToken.AsValue().AsText();

    // Avoid an overflow of the Code-Field
    beerCategory.Code := DelStr(beerCategory.Name, MaxStrLen(beerCategory.Code) + 1);
    beerCategory.Insert();
end;
```

BIER KATEGORIEN EINRICHTUNG

Bitte fügen Sie Bier Kategorien der Liste hinzu

Bier Kategorie Liste

AU...	CODE	NAME	BESCHREIBUNG
<input checked="" type="checkbox"/>	ALTBIER	Altbier	German-Style Altbier
<input checked="" type="checkbox"/>	AMBER ALE	Amber ale	American-Style Amber/Red
<input checked="" type="checkbox"/>	BARLEY WL...	Barley wine	British-Style Barley Wine Ale
<input checked="" type="checkbox"/>	BERLINER...	Berliner Weisse	Berliner-Style Weisse
<input checked="" type="checkbox"/>	BITTER	Bitter	Ordinary Bitter
<input checked="" type="checkbox"/>	BLONDE ALE	Blonde Ale	Golden or Blonde Ale
<input checked="" type="checkbox"/>	BROWN ALE	Brown Ale	English-Style Brown Ale
<input checked="" type="checkbox"/>	DOPPELBO...	Doppelbock	German-Style Doppelbock

Pages in Microsoft AL

Hands On – Wizard

➤ Page “Finish”

- Add an Information Field *“Count of Selected Records”*
- Add an *Option* Field with values *“Replace All”* and *“Merge”*
- Update the value of the Information Field, when the page changes *“Beer Categories”* → *“Finish”*
- Use the Option Information, when the user press *“Finish”* and apply the content of the temp. record *“Beer Categories”* to the database and close the wizard

```
// Prepare New Page: Beer Category List --> Finish
if (CurrentPage = CurrentPage::"Beer Categories") and (newPage = CurrentPage::Finish) then
begin
    tmpBeerCategories.DeleteAll();
    CurrPage.BeerCategories.Page.GetSelectedRecords(tmpBeerCategories);
    ImportRecordCount := tmpBeerCategories.Count();
end;
```

```
local procedure Finish()
begin
    // Apply with import option (replace / merge)
    case KindOfImport of
        KindOfImport::"Replace All": DeleteAll();
        KindOfImport::Merge:         ;
    end;

    // Import the new values
    if tmpBeerCategories.FindSet() then
    repeat
        TransferFields(tmpBeerCategories);
        if not Insert() then
            Modify();
    until tmpBeerCategories.Next() = 0;

    // Close the page
    CurrPage.Close();
end;
```

BIER KATEGORIEN EINRICHTUNG

Import der ausgewählten Bier Kategorien

Ausgewählte Bier Kategorien 11

Art des Imports Alle ersetzen

Zurück Übernehmen

Pages in Microsoft AL

Hands On – Notification

- Use Code Unit „*Beer Mgt. [CC]*“ *ID: 50201*
- Add some Methods to achieve this task:
“Show a notification, when the record “Beer Categories” is empty and call the “Beer Wizard” from the notification by using an action”
- Methods can be
 - IsBeerSetupComplete ... check if record “Beer Categories” is empty
 - CheckBeerSetup ... use the check and show a notification
 - StartBeerSetup ... callback method for the notification action
- Use the codeunit from “Customer Card [CC]” and invoke a notification

```
5 references
var mBeerSetup: Boolean;
    2 references
    mBeerMgt: Codeunit "Beer Mgt. [CC]";

trigger OnOpenPage()
begin
    mBeerSetup := mBeerMgt.CheckSetup();
end;

trigger OnAfterGetRecord()
begin
    mBeerSetup := mBeerMgt.CheckSetup();
end;
```

Q & A



Visual Studio Code

<https://code.visualstudio.com>



Competence Team DevTech

[Microsoft AL](#)