



COSMO CONSULT
Business-Software für Menschen

Microsoft AL

GUI & In-Client Designer IV

Prerequisite

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



Visual Studio Code

<https://code.visualstudio.com>



Competence Team DevTech

[Microsoft AI](#)

VS Code

Useful Extensions

> AL Formatter

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

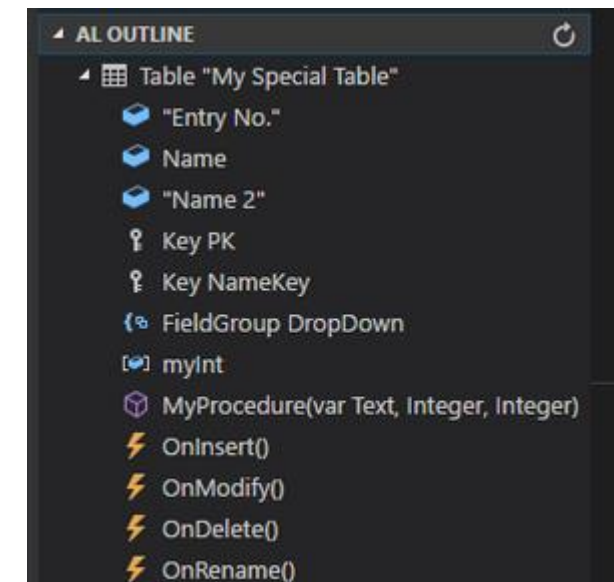
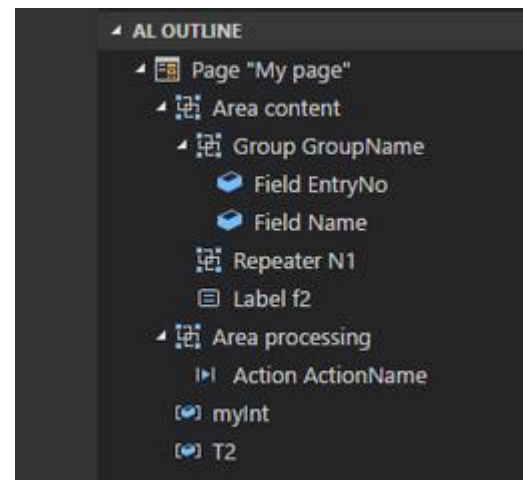
> AL Code Outline

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

> al-util

- > “Rename & Move” Objects
- > Templates e.g. “Readme.md”

```
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 (Re-Cap)
- 2 Wizard (Re-Cap)
- 3 Json, List, Dictionary
- 4 Assisted Setup
- 5 Hands On

Microsoft AL

Language & Types

➤ Re-Cap

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>

Improve the UX

Features

- Wizard (Re-Cap)
- New Datatypes
- Assisted Setups



Pages in Microsoft AL

Wizard (Re-Cap)

> 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*


BEER CATEGORIES SETUP



Please add your Beer Categories to the list

Beer Category List

SE...	CODE	NAME	DESCRIPTION
<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 WI...	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



Back

Next

Pages in Microsoft AL

Json

➤ JsonObject

- .Get(<Name>, jToken)
- .ReadFrom(...) / .WriteTo(...)
- .SelectToken(...)

➤ JsonArray

- *ForEach* Item in Array do begin /* Loop */ end;
- Item *JsonToken*

➤ JsonToken & JsonValue

- .IsObject() .AsObject()
- .IsArray() .AsArray()
- .IsValue() .AsValue().*IsNull()* / *IsUndefined()*
 .AsValue().AsText()
 .AsValue().AsDate()

```
1  {
2    "Beer Categories":
3  Array [
4    { Element
5      "Name":      "Altbier",
6      "Description": "German-Style Altbier"
7    }, Key      Value
8    {
9      "Name":      "Amber ale",
10     "Description": "American-Style Amber/Red Ale"
11   },
12   {
13     "Name":      "Barley wine",
14     "Description": "British-Style Barley Wine Ale"
15   },
16 ]
17 }
```

```
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;
```

Pages in Microsoft AL

List & Dictionary & TextBuilder

➤ Datatype **List** *represents a strongly typed list of value*

- var <Name>: List of [<Datatype>];
- .Contains(<item>)
- .Add(<item>)
- .Remove(<item>)

➤ Datatype **Dictionary** *represents a collection of keys and values*

- var <Name>: Dictionary of [<Key Datatype>, <Value Datatype>]
- .ContainsKey(<key>)
- .Add(<key>, <value>)
- .Remove(<key>)

➤ Datatype **TextBuilder** *can performantly concatenate multiple bigger strings together*

- .Append(<text>)
- .ToText()

Pages in Microsoft AL

Assisted Setup

> 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

➤ Wizard

Pages in Microsoft AL

Hands On

➤ Material

- Project Skeleton „02-GUI-&-In-Client-Designer\Starter\04 My Favorite Beer\”
- Tables: „Beer Category”
- Tables.Extensions: „Customer”, „Item”
- Pages: „Beer Wizard [CC]”

➤ Assets

- “BeerCategories.json”

➤ Prerequisites

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

➤ Hand On - Tasks

- Complete the **Setup Wizard** to initialize Beer Categories and use a **List Part** for Table “Beer Category”
- Register the **Wizard** as **Assisted Setup**

Pages in Microsoft AL

Hands On – Wizard

- Call the Wizard by changing the “*launch.json*” StartupObject information
- The Wizard skeleton looks like this:

```
"startupObjectId": 50200,  
"startupObjectType": "Page"
```

BEER CATEGORIES SETUP

WELCOME
Setup of Beer Categories\you may import the file "...\\Assets\\BeerCategories.json")

Next

BEER CATEGORIES SETUP

BEER CATEGORIES
Please add your Beer Categories to the list

Back Next

BEER CATEGORIES SETUP

FINISH SETUP
Import of selected Beer Categories

Back Assign

Pages in Microsoft AL

Hands On – Wizard

➤ Improve 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 **BeerJson : Text**

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

BEER CATEGORIES SETUP

Setup of Beer Categories\you may import the file "...\\Assets\\BeerCategories.json")

BEER CATEGORIES - JSON

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

Next

Pages in Microsoft AL

Hands On – Wizard

➤ Improve 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 jsonObject: JsonObject;
  jToken: JsonToken;
  jItem: JsonToken;
begin
    if not jsonObject.ReadFrom(BeerJson) then exit(false);
    if not jsonObject.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;
```

BEER CATEGORIES SETUP

Please add your Beer Categories to the list

Beer Category List

SE...	CODE	NAME	DESCRIPTION
<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 WI...	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

Back Next

Pages in Microsoft AL

Hands On – Wizard

➤ Improve 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;
```

BEER CATEGORIES SETUP

Import of selected Beer Categories

Count
11

Kind of Import: Replace All

Back Assign

Pages in Microsoft AL

Hands On – Notification & Assisted Setup

➤ Use Code Unit „*Beer Mgt. [CC]*“ ID: 50201

➤ Register the Wizard as Assisted Setup

- Use the Correct Event Subscriber
- Add the temporary Record

HOME ACTIONS	
VIEW - ASSISTED SETUP	
NAME	STATUS
Beer Category Setup	Completed
Migrate business data	Not Completed
	Not Completed
	Not Completed

```
// ##### SETUP NOTIFICATION #####

[EventSubscriber(ObjectType::Table, Database::"Aggregated Assisted Setup", 'OnRegisterAssistedSetup', '', true, true)]
0 references
local procedure OnRegisterAssistedSetup(var TempAggregatedAssistedSetup: Record "Aggregated Assisted Setup")
var cName: TextConst ENU = 'Beer Category Setup', DEU = 'Bier Kategorie Setup';
    bc: Record "Beer Category [CC]";
begin
    TempAggregatedAssistedSetup.Init();
    TempAggregatedAssistedSetup."Page ID" := Page::"Beer Category List [CC]";
    TempAggregatedAssistedSetup."Assisted Setup Page ID" := Page::"Beer Wizard [CC]";
    TempAggregatedAssistedSetup.Name := cName;
    TempAggregatedAssistedSetup."Item Type" := TempAggregatedAssistedSetup."Item Type"::"Setup and Help";
    if bc.IsEmpty() then
        TempAggregatedAssistedSetup.Status := TempAggregatedAssistedSetup.Status::"Not Completed"
    else
        TempAggregatedAssistedSetup.Status := TempAggregatedAssistedSetup.Status::Completed;
    TempAggregatedAssistedSetup.Insert();
end;
```

Q & A



Visual Studio Code

<https://code.visualstudio.com>



Competence Team DevTech

[Microsoft AL](#)