



# COMMUNITY SUMMIT BC/NAV

2023



The Largest Independent Gathering  
of the Microsoft User Ecosystem



# | Developing an Extension in Business Central

Day 1

Decrease Complexities,  
Deliver Results.

# Your speakers



**Brad Prendergast**  
Director of Development  
Tigunia, LLC



**"KrisGPT" Ruyeras**  
VP of Operations  
Tigunia, LLC

The Largest Independent Gathering  
of the Microsoft User Ecosystem



## About Me

- Microsoft Dynamics NAV / D365 Business Central Enthusiast
- Development, System Architecture, and more
- 1998
- brad@tigunia.com

Decrease Complexities,  
Deliver Results.



Brad Prendergast

The Largest Independent Gathering  
of the Microsoft User Ecosystem



## About Me

- Microsoft Dynamics NAV / D365 Business Central Enthusiast
- Functional, Development, System Architecture (2007)
- Podcast Co-host to Dynamics Corner
- kristoffer@tigunia.com
- Play an Ukulele and prefer MacOS than Windows

Decrease Complexities,  
Deliver Results.



"KrisGPT" Ruyeras

# Know Before You Go

- Restrooms are where?
- Nearest Emergency Exit is where?





# Session Objectives

- Setup a Development Environment
- Understand Business Central Architecture
- Create an extension in Business Central
- Extend Business Central



# Agenda

- Day 1
- Introduction
- Getting Started
- Setting up your Development Environment
- AL Language
- Develop Your First App
- Creating an App
- Extending Business Central
- Day 2

# About You



- Name
- Company
- Role
- What are you hoping to learn?



# Getting Started

# Business Central Architecture

Single-Tenant or On-premises



Clients

BC  
Services

SQL  
Server



Server

Application &  
System Data

Company Data

- Web Server
- Web Services (OData)
- Admin Tools

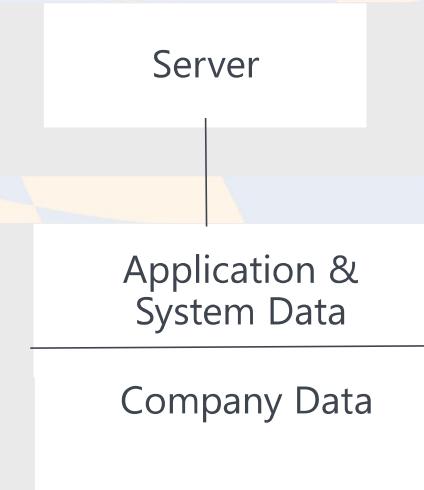
# Business Central Architecture

## Recommended Architecture



Web and BC  
Server

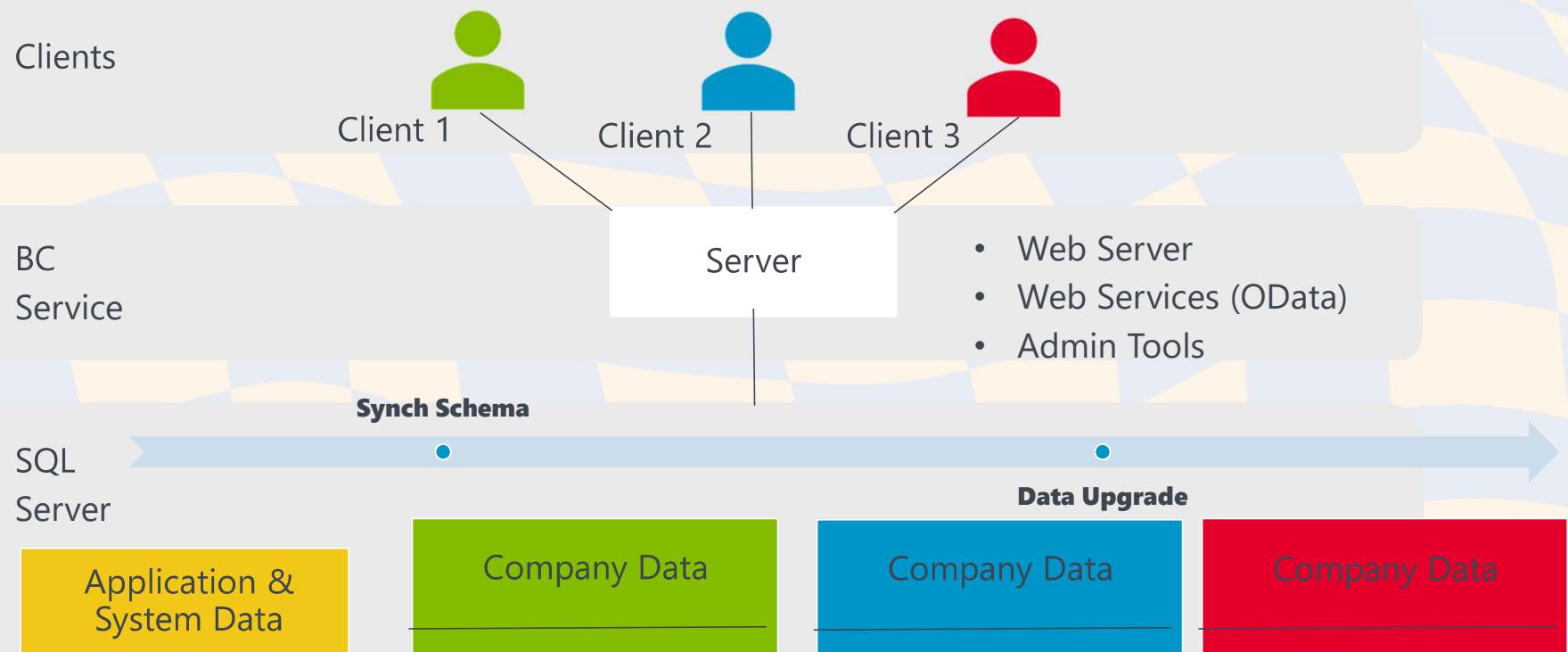
SQL  
Server



# Business Central Architecture



## Multitenant Architecture



# Business Central Architecture



The screenshot shows the Dynamics 365 Business Central homepage for CRONUS USA, Inc. with various UI components highlighted by red arrows:

- Navigation Menu:** Located at the top, showing Finance, Cash Management, Sales, Purchasing, Shopify, Customers, Vendors, Items, Bank Accounts, and Chart of Accounts.
- List Area:** A horizontal menu bar below the navigation menu.
- Action Area:** A dropdown menu titled "Actions" containing options like Sales Quote, Purchase Quote, Find entries..., Payments, Sales Order, Purchase Order, Search in data..., Reports, Sales Invoice, Purchase Invoice, New, and Excel Reports.
- Headline Area:** A large callout box containing a headline and a "Get started" message.
- Cues and Action Tiles:** A large rectangular area containing activity statistics and ongoing sales/purchases/payments.

**Headline:** Want to learn more about Business Central?

**Get started:** Here are a few things you can try out [Show demo tours](#)

**Activities**

Sales This Month	Overdue Sales Invoice Amount	Overdue Purch. Invoice Amount	Sales Invoices Predicted Overdue
\$1,906	\$63,890	\$49,422	0

**Ongoing Sales**

Sales Quotes: 2	Sales Orders: 8	Sales Invoices: 7
-----------------	-----------------	-------------------

**Ongoing Purchases**

Purchase Orders: 8	Ongoing Purch. Invoices: 3	Purch. Invoices... Next Week: 0
--------------------	----------------------------	---------------------------------

**Payments**

Unprocessed Payments: 1	Average Collect... Days: 5.8	Outstanding V... Invoices: 13
-------------------------	------------------------------	-------------------------------

# What is the AL Language?



- AL is the program language used for Dynamics 365 Business Central; Extension Based Approach
- Origin from C/AL (Client/Server Application Language) for extending Dynamics NAV (along side C/SIDE)
- VS Code Integration
- Transpiler – source-to-source compiler – "translator" and "compiler."
- <https://www.dynamicscorner.com/2023/06/episode-220-in-the-dynamics-corner-chair-the-al-compiler-in-depth/>



# Personalize vs. Design vs. Develop

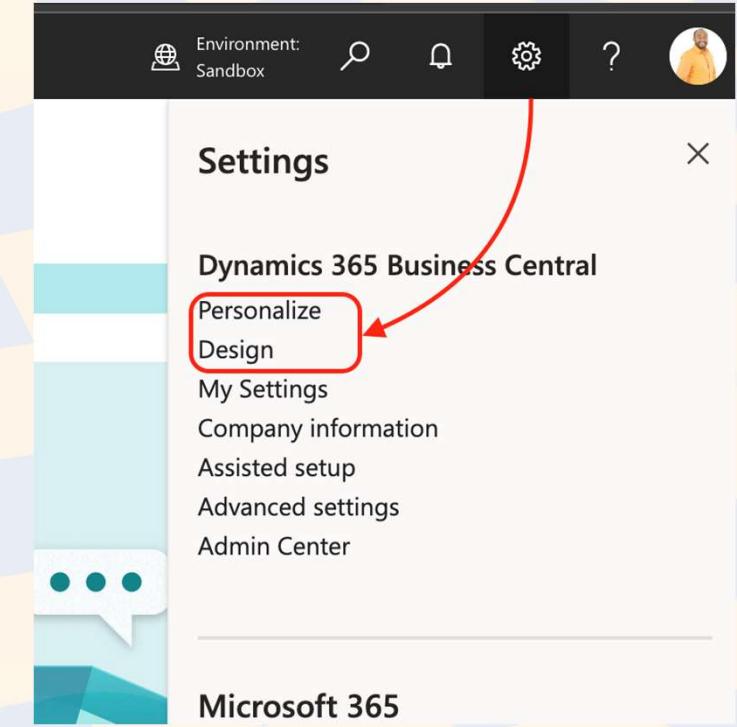


## Personalize

- User Specific
- Limited
- Only Fields Available to Page

## Design

- Global
- Creates new Extension as .zip
- Only from Sandbox



# Personalize vs. Design vs. Develop



**Existing Objects**

- Tables, Pages, Reports, Enums

**Create new objects of any kind**

- Page Type: API

**BC Online**

- Cannot change base objects

**BC On-Premises**

- Base Objects CAN be modified

<https://bit.ly/d365bcuci>

# Extension Types



- **Global**
  - Installed from the AppSource
  - Production and Sandbox
  - Preserved on upgrade for both Production and Sandbox
- **Per-tenant (PTE)**
  - Installed through Extension Management
  - Specific to environment
  - Production and Sandbox
  - Preserved on upgrade in Production (unless there is a problem)
  - Uninstalled from Sandbox when relocated if dependent on DEV



# Extension Types

- DEV
  - Published from VS Code
  - Used for development purposes
  - Only exist in Sandbox
  - uninstalled when the sandbox environment is upgraded or relocated (data is not removed)

Version	Is Inst...	Published As
v. 20.17.34.0	<input checked="" type="checkbox"/>	Dev
v. 22.4.18.0	<input checked="" type="checkbox"/>	PTE
v. 22.5.59966.60187	<input checked="" type="checkbox"/>	Global
v. 22.5.59966.60187	<input checked="" type="checkbox"/>	Global
v. 22.5.59966.60187	<input checked="" type="checkbox"/>	Global

# Extension Management

A screenshot of the Microsoft Extension Management interface. The main window shows a list of installed extensions, including Microsoft, Imaginings 3, and Tigunia Royalties Management. A context menu is open over the Tigunia Royalties Management extension, with options like Manage, Install, Uninstall, Unpublish, Set up, Download Source, Learn More, and Select More. A modal dialog titled "Upload And Deploy Extension" is displayed, containing a list of extensions with deployment status (Global or Dev) and checkboxes. The Tigunia Royalties Management extension is highlighted with a teal background in both the list and the modal. At the bottom right of the modal are "Deploy" and "Cancel" buttons.

Extension	Version	Status
Microsoft	v. 22.5.59966.60187	Global
Imaginings 3	v. 22.4.18.0	PTE
Microsoft	v. 22.5.59966.60187	Global
Microsoft	v. 22.5.59966.60187	Global
Microsoft	v. 22.5.59966.60187	Global
Microsoft	v. 22.5.59966.60565	Global
Microsoft	v. 22.5.59966.60187	Global
Microsoft	v. 22.5.59966.60187	Global
Tigunia Royalties Management	v. 20.17.34.0	Dev
Data Archive	Microsoft	Global
Data Search	Microsoft	Global





# Setup Development Environment

# Setup Development Environment



- Local Environment
- Dynamics 365 Business Central Tenant
- Docker
- Extension Settings

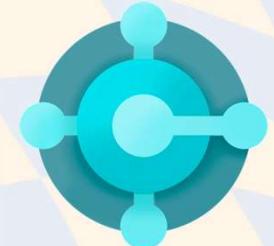
# Local Environment



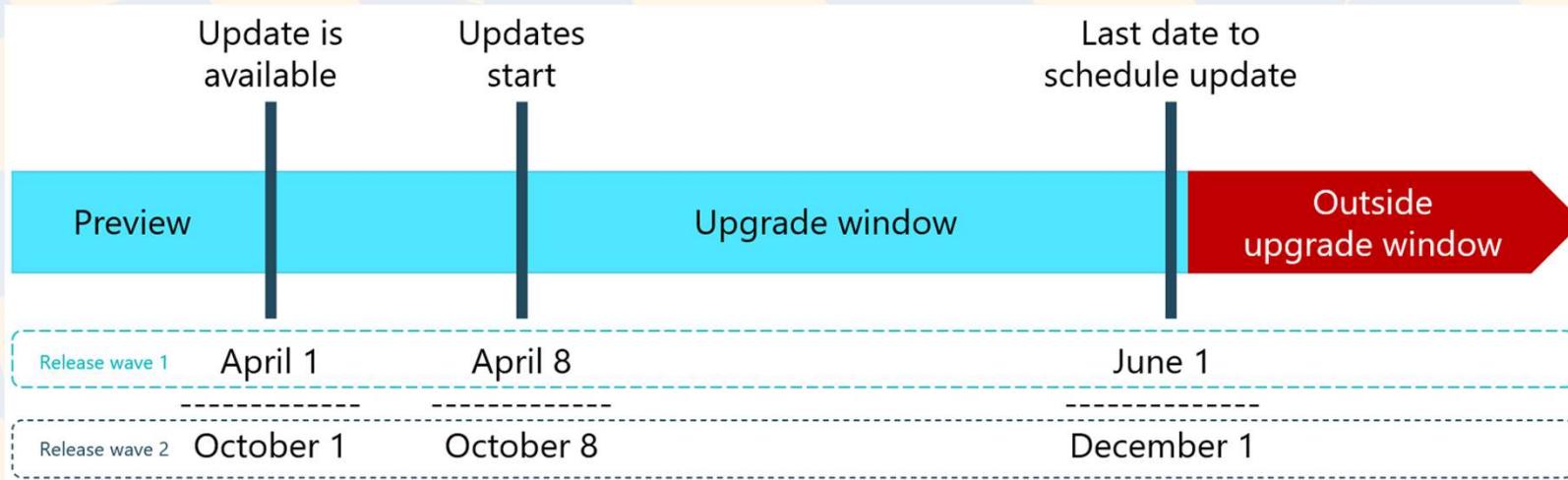
- Visual Studio Code
  - <https://code.visualstudio.com/Download/>
- AL Language Extension
- Other Extensions
  - AZ AL Dev Tools/AL Code Outline
  - Waldo's CRS AL Language Extension
  - Vscode-icons

## Report Builder

<https://www.microsoft.com/en-us/download/details.aspx?id=53613>



# Business Central Tenant



**Update Window is still  
currently at 60 Days**

# Business Central Tenant



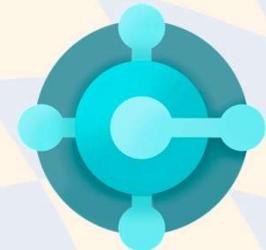
- Admin Center
- Create an Environment
  - Environment Types
  - Copy an environment
- Managing Updates
  - Set Update Window
  - Schedule an Update date
  - Get notified of Updates



# Local Environment



- Docker
  - <https://www.docker.com/get-started/>
- BCContainer Helper
  - Install-Module BcContainerHelper (-force) (-allowPrerelease)
  - Uninstall-Module BCContainerHelper
  - New-BCContainer



# Extension Settings



Environment  
↳ Workspace  
↳ Project

```
{  
  "al.incognito": true,  
  "al.browser": "Chrome",  
  "al.enableCodeAnalysis": true,  
  "al.backgroundCodeAnalysis": false,  
  "al.incrementalBuild": false,  
  "al.codeAnalyzers": [  
    "${AppSourceCop}",  
    "${CodeCop}",  
    "${PerTenantExtensionCop}",  
    "${UICop}"  
  ],  
  "al.compilationOptions": {  
    "generateReportLayout": true  
  },  
  "alOutline.completionProviders": [  
    "VariableNamesWithType",  
    "VariableDataTypes"  
  ],  
  "debug.inlineValues": "on",  
  "debug.console.fontFamily": "default",  
  "debug.console.fontSize": 14,  
  "ui": {  
    "fontFamily": "monospace",  
    "fontSize": 14  
  }  
}
```



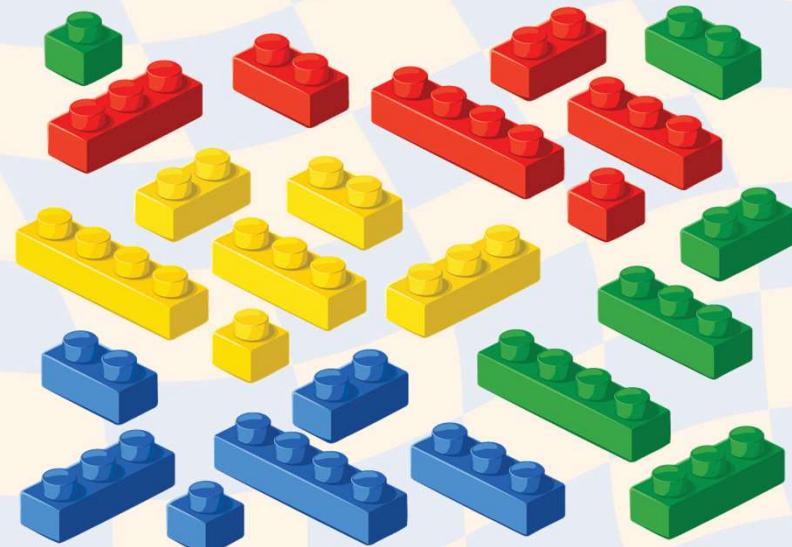


# AL Language

# AL Language



- Objects
  - Table
  - Page
  - Reports
  - Codeunits
  - XMLPorts
  - Query
  - Permissionset



# AL Language – Table



- Tables are the core objects used to store data in Dynamics 365 Business Central
- The structure of a table has four sections:
  - The first block contains metadata for the overall table, such as the table type.
  - The fields section describes the data elements that make up the table, such as their name and the type of data they can store.
  - The keys section contains the definitions of the keys that the table needs to support.
  - The final section details the triggers and code that can run on the table.

```
table S0104 Address
{
    Caption = 'Sample table';
    DataPerCompany = true;

    fields
    {
        field(1; Address; Text[50])
        {
            Description = 'Address retrieved by Service';
        }
        field(2; Locality; Text[30])
        {
            Description = 'Locality retrieved by Service';
        }
        field(3; "Town/City"; Text[30])
        {
            Description = 'Town/City retrieved by Service';
        }
        field(4; County; Text[30])
        {
            Description = 'County retrieved by Service';

            trigger OnValidate();
            begin
                ValidateCounty(County);
            end;
        }
    }
    keys
    {
        key(PrimaryKey; Address)
        {
            Clustered = TRUE;
        }
    }
    var
        Msg: Label 'Hello from my method';

        trigger OnInsert();
        begin
        end;

        procedure MyMethod();
        begin
            Message(Msg);
        end;
    }
}
```

# AL Language – Page



- Pages are the main way to display and organize visual data in Dynamics 365 Business Central.
- The structure of a table has four sections:
  - The first block contains metadata for the overall page; the type of the page and the source table it is showing data from.
  - The next section; the layout, describes the visual parts on the page.
  - The final section details the actions that are published on the page.

```
page 50101 SimpleCustomerCard
{
    PageType = Card;
    SourceTable = Customer;
    ContextSensitiveHelpPage = 'my-feature';

    layout
    {
        area(content)
        {
            group(General)
            {
                field("No.", "No.")
                {
                    ApplicationArea = All;
                    CaptionML = ENU = 'Hello';

                    trigger OnValidate()
                    begin
                        if "No." < '' then
                            Message('Number too small');
                    end;
                }
                field(Name; Name)
                {
                    ApplicationArea = All;
                }
                field(Address; Address)
                {
                    ApplicationArea = All;
                }
            }
        }
        actions
        {
            area(Navigation)
            {
                action(NewAction)
                {
                    ApplicationArea = All;
                    RunObject = codeunit "Document Totals";
                }
            }
        }
    }
}
```

# AL Language – Report



- Reports are used to print or display information\*
- Reports consist of a data model and layout
- A report object consists of four sections
  - The first block contains metadata for the overall table
  - The second block is the dataset that defines the data model
  - The third section is the request page for accepting user input
  - The final section details the triggers and code that can run on the report

```
trigger OnAfterGetRecord();
begin
    CalcFields("Balance (LCY)");
    FormatAddr.FormatAddr(
        CustAddr.Name, "Name 2", "", Address, "Address 2",
        City, "Post Code", County, "Country/Region Code");
end;

}
requestpage
{
    SaveValues = true;
    // These properties come from the report properties
    AboutTitle = 'Awesome';
    AboutText = 'This is a great report!';
    // Use the multi-language support
    // This property defines the language
    // Remember to also set the ContextSensitiveHelpPage
    layout
    {
    }
    actions
    {
    }
    labels
    {
        LabelName = 'Label T';
    }
    trigger OnPreReport();
    var
        CaptionManagement : Codeunit;
        CustFilter := CaptionManagement.GetCustomerFilter();
begin
    CurrReport.PaperNoCaption := Customer_CustFilter;
    Customer_Customer_Post;
    Customer_Customer_Disc;
    Customer_Invoice_Disc;
    Customer_Customer_Price;
    Customer_Payment_Terms;
    Customer_Currency_Code;
    Total_LCY_CaptionLbl := '';
end;
var
    FormatAddr : Codeunit;
    CustFilter : Text;
    CurrReport : Report;
    Customer_ListCaptionLbl;
    CurrReport_PaperNoCaption;
    Customer_Customer_Post;
    Customer_Customer_Disc;
    Customer_Invoice_Disc;
    Customer_Customer_Price;
    Customer_Payment_Terms;
    Customer_Currency_Code;
    Total_LCY_CaptionLbl : Text;
}

report 50103 "Customer List"
{
    CaptionML=ENU='Customer List';
    DefaultLayout = RDLC; // if Word use WordLayout property
    RDCLayout = 'MyRDLReport.rdl';

    dataset
    {
        dataitem(Customer;Customer)
        {
            RequestFilterFields="No.,""Search Name"";Customer Posting Group";
            column(CompanyName;CompanyName)
            {
            }
            column(CurrReport_PageNo;Customer."no.")
            {
            }
            column(Customer_TableCaption_CustFilter;TableCaption + ': ' + CustFilter)
            {
            }
            column(CustFilter;CustFilter)
            {
            }
            column(Customer_No;"No.")
            {
            }
            column(Customer_Customer_Posting_Group;"Customer Posting Group")
            {
            }
            column(Customer_Customer_Disc_Group;"Customer Disc. Group")
            {
            }
            column(Customer_Invoice_Disc_Code;"Invoice Disc. Code")
            {
            }
            column(Customer_Customer_Price_Group;"Customer Price Group")
            {
            }
            column(Customer_Fin_Charge_Terms_Code;"Fin. Charge Terms Code")
            {
            }
            column(Customer_Payment_Terms_Code;"Payment Terms Code")
            {
            }
        }
    }
}
```

\* processing reports; Excel download; Excel Layouts

# AL Language – Codeunit



- Modularized container for AL Code that contains business logic.
- Procedures, Variables, Event Subscribers
- The Access property defines the scope of the codeunit – public or internal.

```
codeunit 50100 MyCodeunit
{
    Access = Public;
    Subtype = Normal;

    trigger OnRun()
    begin
    end;

    procedure MyFunction(Param1: Integer; Param2: Text[50]) : Boolean
    begin
    end;
}
```

# AL Language – XMLPort



- Export and Import data between an external source and Dynamics 365 Business Central
- XML Document
- Variable or Fixed Text Format
- Direction – Inbound, Outbound



# AL Language - Query



- Retrieve records from one or more tables and then combine the data into rows and columns in a single dataset
- Two Types – Normal and API
- dataitems and columns
- SqlJoinType Property

```
query ID Name
{
    elements
    {
        dataitem(DataItem1; Table1)
        {
            column(Column1; Field1)
            {
            }
            column(Column2; Field2)
            {
            }
        dataitem(DataItem2; Table2)
        {
            // Sets a link between FieldY of Table2 and FieldX of Table1.
            DataItemLink = FieldY = DataItem1.FieldX;
            //The dataset contains records from Table1 and Table2 where a match is found between
            SqlJoinType = InnerJoin;

            column(Column1; Field1)
            {
            }
        dataitem(DataItem3; Table3)
        {
            DataItemLink = FieldZ = DataItem2.FieldY;
            SqlJoinType = InnerJoin;
            column(Column1; Field1)
            {
            }
        }
    }
}
```

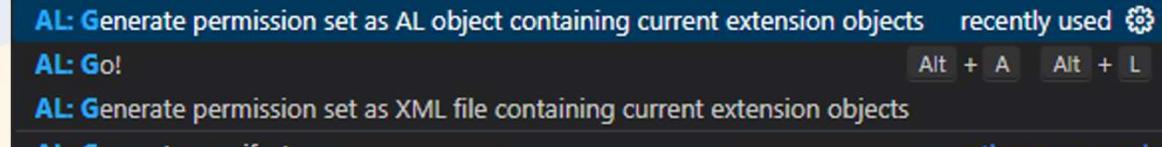
# AL Language – PermissionSet



- Describes permissions on objects
- Assignable to users in Business Central
- Extended permission sets are additive

```
permissionset 50134 "Sales Person"
{
    Assignable = true;
    Caption = 'Sales Person';

    Permissions =
        tabledata Customer = RIMD,
        tabledata "Payment Terms" = RMD,
        tabledata Currency = RM,
        tabledata "Sales Header" = RIM,
        tabledata "Sales Line" = RIMD;
}
```



# Page Inspector

- insight into the page design
- different page elements
- source behind the data

Ctrl + Alt + F1.



> Page Inspection ×

See information about the page, its different elements, and the source behind the data it displays.

Page  
Customer List (22, List)

Explore page in Visual Studio Code

Table  
Customer (18)

View table

Table Fields    Extensions    Page Filters

No. (1, Code[20], PK) 10000 Base Application
Name (2, Text[100]) Adatum Corporation Base Application
Search Name (3, Code[100]) ADATUM CORPORATION Base Application
Name 2 (4, Text[50]) (Blank) Base Application
Address (5, Text[100])



# All Objects



All Objects with Caption

Search

Object Type	Object ID	Object Name	Object Caption	Object Subtype	App Name
TableData	3	Payment Terms	Payment Terms	Normal	Base Application
TableData	4	Currency	Currency	Normal	Base Application
TableData	5	Finance Charge ...	Finance Charge Terms	Normal	Base Application
TableData	6	Customer Price ...	Customer Price Group	Normal	Base Application
TableData	7	Standard Text	Standard Text	Normal	Base Application
TableData	8	Language	Language	Normal	System Application
TableData	9	Country/Region	Country/Region	Normal	Base Application
TableData	10	Shipment Method	Shipment Method	Normal	Base Application
TableData	11	Country/Region ...	Country/Region Translation	Normal	Base Application
TableData	13	Salesperson/Pur...	Salesperson/Purchaser	Normal	Base Application
TableData	14	Location	Location	Normal	Base Application
TableData	15	G/L Account	G/L Account	Normal	Base Application
TableData	17	G/L Entry	G/L Entry	Normal	Base Application
TableData	18	Customer	Customer	Normal	Base Application
TableData	19	Cust. Invoice Disc.	Cust. Invoice Disc.	Normal	Base Application
TableData	21	Cust. Ledger Entry	Cust. Ledger Entry	Normal	Base Application
TableData	23	Vendor	Vendor	Normal	Base Application
TableData	24	Vendor Invoice ...	Vendor Invoice Disc.	Normal	Base Application
TableData	25	Vendor Ledger E...	Vendor Ledger Entry	Normal	Base Application
TableData	27	Item	Item	Normal	Base Application



# Develop your first App

# Develop your First App



- AL Go!
- Download Symbols
- Publish
- Project Files
- AL Home
- AL Explorer
- Source Code Analyzers



# Lunch ?!?





# Development

# Development – Tables



```
0 references
table id MyTable
{
    <property> = <value>

    fields
    {
        0 references
        field(1; MyField; Integer)
        {
            <property> = <value>
        }
    }

    keys
    {
        key(Key1; MyField)
        {
            Clustered = true;
        }
    }

    fieldgroups
    {
        fieldgroup(Name; Fields)
        {
        }
    }

    var
    myInt: Integer;

    trigger <OnInsert()> <OnModify()> <OnDelete()> <OnRename()>
    begin
    end;

    <local> procedure <method_name>(parameter list) <return_value_name> : <data_type>[<length>]
    begin
    end;
}
```

# Development

- Assignment Statements
- Control Statements
- Repetitive Statements



```
procedure assignments()
var
  item: Record Item;
  itemNo: Code[20];
  price: Decimal;
  counter: Integer;
  i: Integer;
  color: Text;

begin
  itemNo := '1000';
  Item.Get(itemNo);
  Counter := 1;

  if counter > 1 then begin
    price := Item."Unit Price" * Counter;
  end else begin
    price := Item."Unit Price";
  end;

  for i := 1 to 10 do begin
    counter += 1;
  end;

  while counter > 0 do begin
    counter -= 1;
  end;

  repeat
    counter += 1;
  until counter >= 10;

  case color of
    'blue':
      Message('blue');
    'black':
      Message('black');
    'green':
      Message('green');
    else
      Message('no color');
  end;
end;
```

# Development – Pages



```
page Id MyPage
{
    PageType = Card;
    ApplicationArea = All;
    UsageCategory = Administration;
    SourceTable = TableName;

    layout
    {
        0 references
        area(Content)
        {
            0 references
            group(GroupName)
            {
                0 references
                fieldName; NameSource
                {
                    ApplicationArea = All;
                }
            }
        }
    }

    actions
    {
        0 references
        area(Processing)
        {
            0 references
            action(ActionName)
            {
                ApplicationArea = All;

                0 references
                trigger OnAction()
                begin
                end;
            }
        }
    }

    var
        0 references
        myInt: Integer;
    }
```

A screenshot of a Dynamics 365 Customer Card page for Adatum Corporation. The card displays general information such as number (10000), name (Adatum Corporation), balance (\$), total sales, and costs. It also shows address and contact details like address (192 Market Square), city (Atlanta), state (GA), and ZIP code (31772).

Customer Card

10000 · Adatum Corporation

No. ..... 10000 Credit Limit (\$) ..... 0.00

Name ..... Adatum Corporation Blocked ..... ✓

Balance (\$) ..... 0.00 Total Sales - Fiscal Year ..... 60,672.80

Balance (\$) As Vendor ..... 0.00 Costs (\$) ..... 40,255.70

Balance Due (\$) ..... 0.00

Address & Contact

Address ..... 192 Market Square Phone No. ....

Address 2 ..... Mobile Phone No. ....

Country/Region Code ..... US Email ..... robert.townes@contoso.com

City ..... Atlanta Home Page .....

State ..... GA Contact ..... Robert Townes

ZIP Code ..... 31772

Show on Map

Invoicing >

DOMESTIC DOMESTIC

# Development – Pages



Page type	Examples of use	Main data display
RoleCenter	Overview of business performance and the start page for a specific user profile.	Defined by the embedded parts.
Card	Master, reference, and set up data management. Card page example	Single entity
Document	Transaction and other document management.	Single entity
ListPlus	Statistics, details, and related data management.	Single entity
List	Entity overviews and navigation, and inline editing of simple entities. List page example	Collection of entities/entries
Worksheet	Line-based data entry tasks (such as journals) and inquiries.	Collection of entities
CardPart	A page that is embedded in another page, such as in a FactBox.	Single entity
ListPart	A page that is embedded in another page, such as in a FactBox.	Collection of entities/entries



# Exercise 1

The Contoso Training and Events (CTE) company would like the ability to capture Instructor profile. They need to be able to capture detail information regarding the instructor. CTE would like to capture an instructor ID, Name, Type, and contact details. They would also like to be able to list Training and Events Sessions that contains the session ID, Name or description, capacity, Type, and if possible, start and end date.

Then create a location for where those sessions will be held.





# Development



- Table Relations – Sets up a lookup into another table

```
TableRelation = <TableName>[.<FieldName>] [WHERE(<TableFilters>)] |  
[IF(<Conditions>) <TableName>[.<FieldName>] [WHERE(<TableFilters>)] ELSE <TableRelation>]  
<Conditions> ::= <TableFilters>  
<TableFilters> ::= <TableFilter> {,<TableFilter>}  
<TableFilter> ::= <DestinationFieldName>=CONST(<FieldConst>) | FIELD(<SourceFieldName>)
```

```
    field(5; "Instructor No."; Code[20])  
    {  
        Caption = 'Instructor No.';  
        TableRelation = "DEV Instructor"."No.";  
    }
```

# Development



- Enumerations – set of named constants

```
enum 50101 "DEV-Program-Level"
{
    Extensible = true;

    0 references
    value(0; "Beginner")
    {
        Caption = 'Beginner';
    }
    0 references
    value(1; Intermediate)
    {
        Caption = 'Intermediate';
    }
    0 references
    value(2; Advanced)
    {
        Caption = 'Advanced';
    }
}
```

# Development

- PageParts



```
part(SalesLines; "DEV Registration Subform")
{
    ApplicationArea = Basic, Suite;
    SubPageLink = "Registration No." = field("No.");
    UpdatePropagation = Both;
}
```

- PageActions

```
action(Instructors)
{
    ApplicationArea = Basic, Suite;
    Caption = 'Instructors';
    Image = Resource;
    RunObject = page "DEV Session Instructors";
    RunPageLink = "Session No." = field("No.");
    ToolTip = 'View or edit the instructors for this session.';

    0 references
    trigger OnAction()
        begin
            InstructorList.Run();
        end;
}
```



## Exercise 2



The Contoso Training and Events (CTE) company would like to extend the functionality beyond just managing the instructor profiles into a page that shows what sessions the instructors are assigned to. Contoso would like to be able to capture registrations that contains the session and Contact/Attendees. Create a page and subpage.



# Extending Business Central

# Extending Business Central



- Table Extension Object
  - allows you to add additional fields or to change some properties on a table
  - define keys for fields added in the table extension
  - define keys for fields in base table
- Page Extension Object
  - adds or overrides the functionality
  - Use keywords to place actions and controls (addfirst, addlast, add after)

# Extending Business Central



- FlowFields – display the result of the calculation described in the CalcFormula Property.

- Sum The sum of a specified set in a column in a table.
- Average The average value of a specified set in a column in a table.
- Exist Indicates whether any records exist in a specified set in a table.
- Count The number of records in a specified set in a table.
- Min The minimum value in a column in a specified set in a table.
- Max The maximum value in a column in a specified set in a table.
- Lookup Looks up a value in a column in another table.

```
field(80; "No. Sessions"; Integer)
{
    Caption = 'No. Sessions';
    Editable = false;
    FieldClass = FlowField;
    CalcFormula = count("DEV Session Instructor" where("Instructor No." = field("No.")));
}
```

# Extending Business Central



- FactBoxes – right-most side of a page for displaying content – other pages, charts, system parts

```
page 50100 "Simple Customercard Page"
{
    PageType = Card;

    layout
    {
        area(FactBoxes)
        {
            part(MyPart; "Acc. Sched. KPI Web Srv. Lines")
            {
                ApplicationArea = All;
                SubPageView = SORTING ("Acc. Schedule Name");
            }
            systempart(Links; Links)
            {
                ApplicationArea = All;
            }
            systempart(Notes; Notes)
            {
                ApplicationArea = All;
            }
        }
    }
}
```

The screenshot shows the Sales Order page for Sales Order S-ORD101001 from Adatum Corporation. The page includes fields for Customer Name (Adatum Corporation), Contact (Robert Toynes), Order Date (4/2/2023), Due Date (5/2/2023), Status (Open), and Posting Date (4/2/2023). The main grid displays a single line item for Item 1996-S, ATLANTA Whiteboard, base, with a quantity of 12. The total amount is 16,767.60, and the total tax is 1,005.06, resulting in a total incl. Tax of 17,773.66. To the right of the grid, several FactBoxes are displayed:

- Customer Details:** Shows Customer No. 10000, Name Adatum Corporation, and Email robert.toynes@contoso.com.
- Sell-to Customer Sales History:** A grid showing sales data for the customer. It has four columns: Ongoing Sales Quotas (0), Ongoing Sales Orders (0), Ongoing Sales Invoiced (6), and Ongoing Sales Received (2). It also includes sections for Pending Sales, Pending Sales Invoiced, and Pending Sales Received.
- Sales Line Details:** Shows the item details for the current sales line: Item No. 1996-S, Required Quantity 12, and Unit of Measure PCS.

## Exercise 3

The Contoso Training and Events (CTE) company would like the ability to see the instructor information quickly from the registration list without having to drill into the record itself. Create a factbox that shows additional detail of the instructor. It must list the Instructor ID, Name, Type, and Job Title.





@ 2023 Dynamic Communities

# Day 1 Recap

- Set up Development Environment
- AL Language
- Creating an App
- Extending Business Central



# Thank You





# COMMUNITY SUMMIT BC/NAV



# COMMUNITY SUMMIT BC/NAV

2023



The Largest Independent Gathering  
of the Microsoft User Ecosystem



# | Developing an Extension in Business Central

Day 2

Decrease Complexities,  
Deliver Results.

# Day 1 Recap

- Set up Development Environment
- AL Language
- Creating an App
- Extending Business Central
- Reports and Report Extensions





# Agenda

- Day 1
- Day 2
- Reports and Report Extensions
- Working with multiple Projects
- Codeunits
- Functions
- Events
- Working with APIs
- Debugging in AL



# Working with Multiple Projects

# Working with Multiple Projects



- **Workspaces**
  - A Visual Studio Code "workspace" is the collection of one or more folders that are opened in a VS Code window (instance)
  - `<name>.code-workspace`

# Working with Multiple Projects



- Dependencies
  - Packages that the extension is dependent upon

A screenshot of a code editor interface. At the top, there is a dark rectangular box containing JSON code:

```
"dependencies": [  
  {  
    "id": "eeb5ab52-dbde-4bac-b267-c44446a1b7ef",  
    "name": "SummitNA 2023 Dev Sample",  
    "publisher": "SummitNA 2023",  
    "version": "22.5.0.0"  
  }  
,
```

Below this, the main code editor area shows a portion of a file with the following content:

```
>dep
```

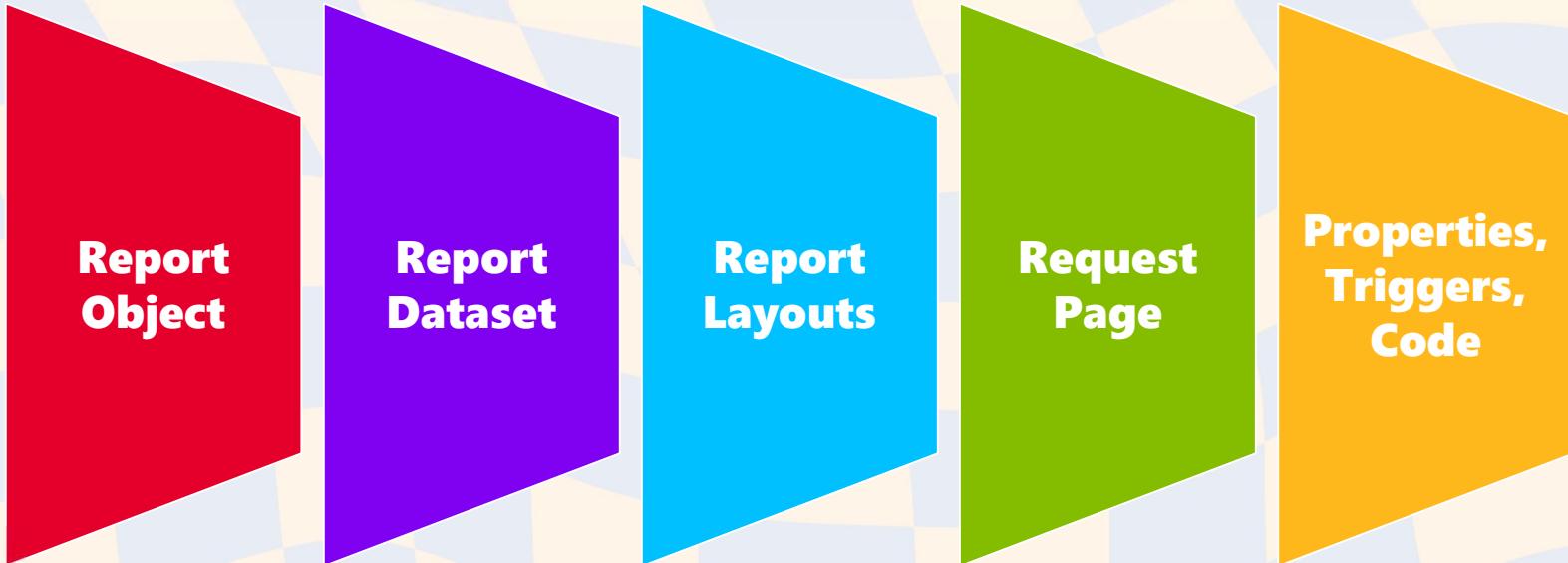
At the bottom of the screen, a dark bar displays the text "AL: Publish full dependency tree for active project" and "Shift + Alt + W recently used".





# Reports

# Report Architecture



# Report Layouts



RDL

Word

Excel

# Extending Base Report



- Request Pages
- New Data Items
- Trigger Implementations
- Columns to existing data items in the report dataset
- New Report layout to reflect the new fields



# Multiple Report Layouts



- treporters
- Rendering
- Layout in sections
- Caption
- Summary

LayoutFile



```
nds "Employee - List"

own in Pivot table in Excel';
ot.xlsx';

rted by last name in Excel';
stName.xlsx';
```

```
}
```

```
layout(LayoutWord)
{
    Type = Word;
    Caption = 'WordList';
    Summary = 'Employee list sorted by last name in Word';
    LayoutFile = 'EmpSortedByLastName.docx';
}
```

# Exercise 4



The Contoso Training and Events (CTE) company would now like to have a few reports built.

- Create a report that list the sessions.
- Extend an existing Report
- Use Report Layout of your choice on one
- Utilize the Multiple Report Layouts with Rendering





# Codeunits

# Codeunits



- Modularized container for AL Code that contains business logic.

```
0 references
codeunit 50101 "DEV Rental Management"
{
    Access = Public;
    Subtype = Normal;
    TableNo = "DEV Book Rental Header";

    0 references
    trigger OnRun()
    begin
        CheckRental(Rec);
    end;
}
```

# Codeunits – Procedures



- group of statements that perform an operation or task
- Scope determines if the procedures can be called, from the same object in which they are declared or from other parts of the application

```
... // [Attributes(arguments · list)]
... // local · procedure <method_name> (parameter · list) · <return_value_name> · : · <data_type> [<length>]
```

# Codeunits - Events



- What are events?
- Business Events
- Integration Events

event Subscriptions





# Lunch ?!?



# Exercise 5

The Contoso Training and Events (CTE) company would now like to

Create a new extension that adds two new fields to the instructor table and displays those fields on the instructor card

Create a codeunit that can accept the Registration Table as a source that will check for specific conditions of your choice on the registration

Create a library codeunit with a set of procedures for performing mathematical functions on sets of numbers.

Create a page that accepts user input and processes the data by using the library procedures





@ 2023 Dynamic Communities



# Working with APIs

# Working with APIs



- What are APIs?
  - Application Programming Interface
  - allows two applications to talk to each other
  - Standardized

# Working with APIs



- API Pages
  - API Page types create versioned, webhook-supported, OData v4 enabled REST web services
  - page cannot be displayed in the user interface
- Naming
  - camelCase for naming attributes, tables, as well as APIPublisher, APIGroup, EntityName, and EntitySetName
  - Alphanumeric characters
  - APIVersion follows the pattern vX.Y

# Working with APIs



```
page 50200 "DEV Widget Entity"
{
    APIGroup = 'Sample';
    APIPublisher = 'SummitNA';
    APIVersion = 'v2.0';
    Caption = 'widgetEntity', Locked = true;
    ChangeTrackingAllowed = true;
    DelayedInsert = true;
    EntityName = 'widget';
    EntitySetName = 'widgets';
    ODataKeyFields = SystemId;
    PageType = API;
    SourceTable = "DEV Widget";

    layout
    {
        0 references
        area(content)
        {
            0 references
            repeater(General)
            {
                0 references
                field(id; Rec.SystemId)
                {
                    ApplicationArea = All;
                    Caption = 'Id', Locked = true;
                    Editable = false;
                }
                0 references
                field(no; Rec."No.")
                {
                    ApplicationArea = All;
                    caption = 'no', Locked = true;
                }
                0 references
                field(description; Rec.Description)
                {
                    ApplicationArea = All;
                    caption = 'description', Locked = true;
                }
            }
        }
    }
}
```

# Working with APIs



# Exercise 6



The Contoso Training and Events (CTE) company would like to provide an endpoint for a 3rd-Party, Vendor, or Website to consume registrations, sessions available, or Instructor information.

- You should create an API Page for information to be consumed externally.



# Debugging in AL



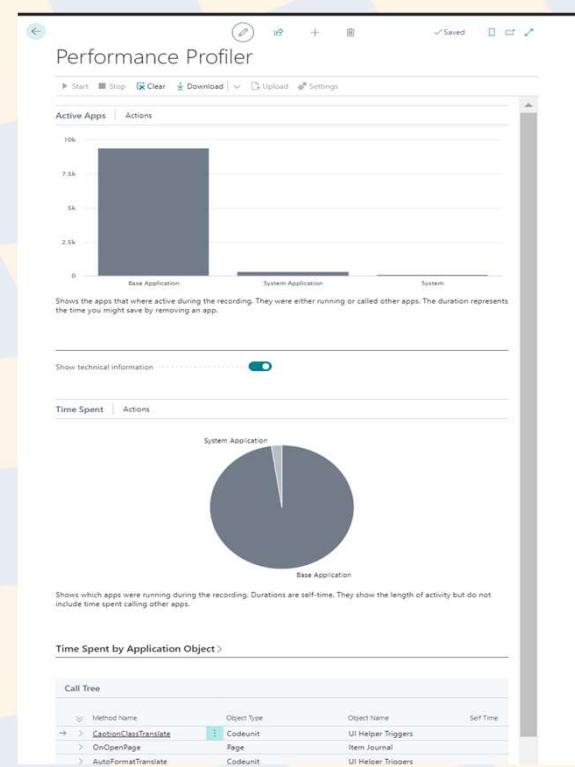
- Breakpoints
- Break on Errors
- Break on Record Change
- Attach and Debug Next
- Watches

```
{
  "name": "Microsoft cloud sandbox",
  "request": "attach",
  "type": "al",
  "environmentType": "Sandbox",
  "environmentName": "sandbox",
  "tenant": "117d2cc0-dbb2-4e85-b82d-96d851551dff",
  "breakOnError": "All",
  "breakOnRecordWrite": "ExcludeTemporary",
  "userId": "brad",
  "enableLongRunningSqlStatements": true,
  "enableSqlInformationDebugger": true
},
```



# Performance Profiler

- In-client Performance Profiler, to record a snapshot of the process
- Profiler monitors all of the apps that are involved in the process
- Identify where there may be a holdup



# Day 2 Recap

- Working with multiple Projects
- Codeunits
- Functions
- Events
- Working with APIs
- Debugging in AL
  
- It doesn't end here!





# Suggested Resources

- Development in AL  
<https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-dev-overview>
- Get started with AL  
<https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-get-started>
- Extensibility overview  
<https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-extensibility-overview>
- Helpful Links  
<https://www.dvlprlife.com/links/helpful-links/>



# Suggested Resources

- **MS Learn**
  - Get started with development in Microsoft Dynamics 365 Business Central - <https://learn.microsoft.com/en-us/training/paths/development-get-started-business-central/>
  - Learn the application foundation with the AL programming language for Microsoft Dynamics 365 Business Central <https://learn.microsoft.com/en-us/training/paths/application-foundation-al-language/>
  - Work with essential development standards for Microsoft Dynamics 365 Business Central <https://learn.microsoft.com/en-us/training/paths/essential-development-standards/>
  - Tailor roles and design the UI for Microsoft Dynamics 365 Business Central <https://learn.microsoft.com/en-us/training/paths/tailor-roles-design-ui/>



# Suggested Resources

- **MS Learn**
  - Discover the foundation of customizing Microsoft Dynamics 365 Business Central  
<https://learn.microsoft.com/en-us/training/paths/foundation-customize-business-central/>
  - Build reports for Microsoft Dynamics 365 Business Central  
<https://learn.microsoft.com/en-us/training/paths/build-reports/>
  - Work with XMLports in Dynamics 365 Business Central  
<https://learn.microsoft.com/en-us/training/modules/work-with-xmlports/>
  - Work with queries in Dynamics 365 Business Central  
<https://learn.microsoft.com/en-us/training/modules/work-with-queries/>



# Thank you for Attending



Brad Prendergast



"KrisGPT" Ruyeras





# COMMUNITY SUMMIT BC/NAV