Prerequisites

Microsoft Dynamics 365 Business Central

Visual Studio Code

Microsoft Office

- Word

Lab 1.1: Creating and Running an AL Extension

Exercise Scenario

Isaac is a developer for Cronus International Ltd. and is learning how to create objects in the development environment Visual Studio Code. He creates a new AL extension and publishes the extension to his test environment. He should see a Hello World message.

Exercise 1: Installing the AL Extension for Visual Studio Code

Exercise Scenario

Isaac installs the AL Extension.

Task 1: Start Visual Studio Code

High Level Steps

1. Start Visual Studio Code from Windows.

Detailed Steps

1. Start Microsoft Visual Studio Code from Windows.
    a. Click Start > Visual Studio Code.

Task 2: Install the AL Extension

High Level Steps

1. Go to the tab Extensions and install the AL Extension.

Detailed Steps

1. Go to the tab Extensions.
    a. Click View > Extensions.
    b. Type "AL Language" in the "Search Extensions in the Marketplace" search box.
    c. Click the green "Install" button.
    d. After installation click the "Reload" button.

Exercise 2: Create a new AL Extension project

Exercise Scenario

Isaac creates a new AL Extension project with the name "Hello World" from the publisher "Cronus International Ltd".

Task 1: Create a new AL Extension project

High Level Steps

1. Create a new AL Extension project.

2. Correct the settings in the launch.json file.
3. Change the settings in the app.json file

Detailed Steps

1. Create a new AL Extension project
    a. Click View > Command Palette…
    b. Type "AL: Go!" in the search box and press ENTER.
    c. Ignore the authentication request by pressing ESC.
2. Correct the settings in the launch.json file.
    a. Click View > Explorer. The Explorer pane opens.
    b. Click the folder ".vscode".
    c. Click the file "launch.json".
    d. Change the "server" setting with the URL of your Microsoft Dynamics 365 Business Central test environment.
    e. Change the "serverInstance" setting with the instance name of your test environment (if needed).
3. Change the settings in the app.json file
    a. Click the app.json file in the Explorer pane.
    b. Change the "name" setting to "Hello World".
    c. Change the "publisher" setting to "Cronus International Ltd".

Task 2: Download the application symbols.

High Level Steps

1. Download the application symbols.
2. Enter your authentication credentials.
3. Verify the download of the application symbols.
4. Package your AL Extension.

Detailed Steps

1. Download the application symbols.
    a. Click View > Command Palette…
    b. Type "AL: Download symbols" in the search box and press ENTER.
2. Enter your authentication credentials.
    a. Type your username and press ENTER.
    b. Type your password and press ENTER.
3. Verify the download of the application symbols.
    a. Verify that you see a ".alpackages" folder with two ".app" files.
    b. Verify that the "HelloWorld.al" file doesn't show in red.
    c. Open the "HelloWorld.al" file and verify that "Customer List" isn't underlined in red.
4. Package your AL Extension.
    a. Click View > Command Palette…
    b. Type "AL: Package" in the search box and press ENTER.
    c. Verify that you see a ".app" file with the name "Cronus International Ltd_Hello World_1.0.0.0.app".

Exercise 3: Publish the AL Extension to see the "Hello World" message box.

Exercise Scenario

Isaac publishes the AL Extension from Visual Studio Code and opens the "Customer List" to see the "Hello World" message box.

Task 1: Publish the AL Extension project

High Level Steps

1. Publish the AL Extension.

Detailed Steps

1. Publish the AL Extension.
    a. Click View > Command Palette…
    b. Type "AL: Publish"
        i. The Microsoft Dynamics 365 Business Central application launches.
        ii. The Hello World message box appears.

Lab 2.1 Create a Table

Scenario

Simon is a developer working for CRONUS International Ltd. CRONUS

International Ltd. has decided to start selling Microsoft Dynamics 365 Business Central training courses as its business.

Simon must create a table to record course information and set several keys so that his users have the option for a different sorting sequence for the records in the table.

Objectives

The objectives are:

Show how to create a table, set basic field properties, and then create primary and secondary keys for the table.

Exercise 1: Create a Table

Exercise Scenario

Simon creates a table to store the information about courses.

Task 1: Create a Table

High Level Steps

1. Start a new AL Extension project with the name "Course Mgmt" and the publisher "Cronus International Ltd".
2. Remove the HelloWorld.al file and create a table to keep the course information. This includes the course code, course name, course description, course duration in days, course type (instructor led, e-learning or remote training), course prices, and whether it is an active course or not.
3. Set the OptionMembers property for the Type field.
4. Set the TableRelation property for the Instructor Code field.
5. Set the Primary Key field to "Code".
6. Add secondary keys for "Instructor Code" and for "Type"
7. Remove the other code.
8. Set the correct settings in the launch.json file to start the new table.
9. Publish the table.
10. Add records to the new table.

Detailed Steps

1. Start a new AL Extension project with the name "Course Mgmt" and the publisher "Cronus International Ltd".
    a. Click View > Command Palette...
    b. Type "AL: Go!" in the search box and press ENTER.
    c. Ignore the authentication request by pressing ESC.
    d. Click View > Explorer. The Explorer pane opens.

e. Click the folder ".vscode".
f. Click the file "launch.json".
g. Change the "server" setting with the URL of your Microsoft Dynamics 365 Business Central test environment.
h. Change the "serverInstance" setting with the instance name of your test environment (if needed).
i. Click the app.json file in the Explorer pane.
j. Change the "name" setting to "Course Mgmt".
k. Change the "publisher" setting to "Cronus International Ltd".
l. Click View > Command Palette...
m. Type "AL: Download symbols" in the search box and press ENTER.

2. Remove the HelloWorld.al file and create a table to keep the course information. This includes the course code, course name, course description, course duration in days, course type (instructor led, e-learning or remote training), course prices, and whether it is an active course or not.
    a. Remove the HelloWorld.al file.
    b. Create a new file with the name "Course.al".
    c. Create a new table by using code snippets. Type ttable and press the TAB key.
    d. Change the ID in 50100 and the name in "Course".
    e. Remove the field "MyField".
    f. Create the following fields:

Field No.	Field Name	Data Type	Length
10	Code	Code	10
20	Name	Text	30
30	Description	Text	50
40	Type	Option	
50	Duration	Decimal	
60	Price	Decimal	
70	Active	Boolean	
80	Difficulty	Integer	
90	Passing Rate	Integer	
100	Instructor Code	Code	20

3. Set the OptionMember property for the Type field.
    a. OptionMembers: "Instructor Led","e-Learning","Remote Training"
4. Set the TableRelation property for the Instructor Code field.
    a. TableRelation: Resource WHERE (Type=CONST(Person))
5. Set the Primary Key field to "Code".
    a. Change the Primary Key "MyField" to "Code".
6. Add secondary keys for "Instructor Code" and for "Type"
    a. Add a key with the name "Secondary Key 1" for the field "Instructor Code"
    b. Add a key with the name "Secondary Key 2" for the field "Type"
7. Remove the other code.
    a. Remove the variable "myInt".

b. Remove all the trigger code.

8. Set the correct settings in the launch.json file to start the new table.
    a. Open the launch.json file.
    b. Change the "startupObjectId" to 50100.
    c. Change the "startupObjectType" to "Table"
9. Publish the table.
    a. Click View > Command Palette…
    b. Type "AL: Publish"
10. Add records to the new table.
    a. When the Microsoft Dynamics 365 Business Central application launches create the following:

Code	Name	Description	Type	Duration	Price	Active	Difficulty	Passing Rate
80040	Installation & Configuration	Basic knowledge of installation and configuration	Remote Training	2	1,000	Yes	5	75
80041	Finance	Basic knowledge of finance	Instructor Led	3	1,500	Yes	7	80
80042	C/SIDE Introduction	Introduction to programming	Instructor Led	5	2,500	Yes	8	80
80043	Introduction	Introduction to Microsoft Dynamics NAV	Remote Training	2	1,000	Yes	4	60
80049	Application Setup	Basic knowledge of application setup	e-Learning	2	1,000	Yes	5	65
80050	Business Intelligence	Basic knowledge of Business Intelligence	e-Learning	1	500	Yes	5	65
80055	C/SIDE Solution Development	Advanced topics in programming	Instructor Led	5	2,500	Yes	10	75

    b. Close the Table

Exercise 2: Create a FlowField

Exercise Scenario

Simon creates a flowfield Instructor Name.

Task 1: Create a Flowfield

High Level Steps

1. Create an Instructor Name Flowfield.

Detailed Steps

1. Create an Instructor Name Flowfield:
    a. Open the Course.al file.
    b. Add a new field Instructor Name:
        i. Field No: 120
        ii. Field Name: Instructor Name
        iii. Data Type: Text
        iv. Length: 50
    c. Set the following properties:
        i. Editable: No
        ii. FieldClass: FlowField
        iii. CalcFormula: Lookup(Resource.Name WHERE (No.=FIELD(Instructor Code)))
2. Publish the table.
    a. Click View > Command Palette…
    b. Type "AL: Publish".
    c. Close the table.

Lab 3.1 Create a Table Extension

Scenario

Simon is a developer working for CRONUS International Ltd. Simon has to create a solution to keep social media information for his customers. He must create a table extension to be able to safe social media information.

Objectives

The objectives are:

Show how to create a table extension, add extra fields.

Exercise 1: Create a Table Extension

Exercise Scenario

Simon creates a table extension to store the social media information.

Task 1: Create a Table Extension

High Level Steps

1. Start a new AL Extension project with the name "Social Media" and the publisher "Cronus International Ltd".
2. Remove the HelloWorld.al file and create a table extension to keep the social media information. This includes fields for Facebook, Twitter, Instagram, LinkedIn.
3. Remove the other code.

Detailed Steps

1. Start a new AL Extension project with the name "Social" and the publisher "Cronus International Ltd".
    a. Click View > Command Palette...
    b. Type "AL: Go!" in the search box and press ENTER.
    c. Ignore the authentication request by pressing ESC.
    d. Click View > Explorer. The Explorer pane opens.
    e. Click the folder ".vscode".
    f. Click the file "launch.json".
    g. Change the "server" setting with the URL of your Microsoft Dynamics 365 Business Central test environment.
    h. Change the "serverInstance" setting with the instance name of your test environment (if needed).
    i. Click the app.json file in the Explorer pane.
    j. Change the "name" setting to "Social Media".
    k. Change the "publisher" setting to "Cronus International Ltd".
    l. Click View > Command Palette...
    m. Type "AL: Download symbols" in the search box and press ENTER.
2. Remove the HelloWorld.al file and create a table extension to keep the social media information. This includes fields for Facebook, Twitter, Instagram,

LinkedIn.

a. Remove the HelloWorld.al file.
b. Create a new file with the name "SocialMediaTableExt.al".
c. Create a new table extension by using code snippets. Type ttableext and press the TAB key.
d. Change the ID in 50110 and the name in "Social Media".
e. Set the extends to "Customer"
f. Create the following fields:

Field No.	Field Name	Data Type	Length
50110	Facebook	Text	50
50111	Twitter	Text	30
50112	Instagram	Text	50
50113	LinkedIn	Text	50

3. Remove the other code.
    a. Remove the variable "myInt".

Lab 4.1 Create a Card Page

Exercise 1: Create a Card page for the Course table.

Exercise Scenario

Simon wants to follow the standard recommendations for cards and lists. He first creates a card page, names it according to card page naming rules, and saves it.

Task 1: Creating a new page

High Level Steps

1. Create a new Card page that has the source table Course.
2. Set page properties to show the caption according to the naming conventions for card pages.

Detailed Steps

1. Create a new Card page that has the source table Course.
    a. Create a new file with the name "CourseCard.al".
    b. Type "tpage" and press the tab button.
    c. Change the ID to 50100 and the name to "Course Card".
    d. Verify that PageType property is set to "Card" and set the SourceTable property to "Course".
2. Set page properties to show the caption according to the naming conventions for card pages and set the UsageCategory property.
    a. Create the Caption property, enter "Course Card".
    b. Set the UsageCategory, enter "Documents".

Task 2: Adding controls

High Level Steps

1. Add a ContentArea container.
2. Add a FastTab.
3. Add the fields.

Detailed Steps

1. Add a ContentArea container.
    a. Verify that an area of the SubType "Content" is created in the layout section. If not, create an area of the type "Content".
2. Add a FastTab.
    a. Verify that a group exists in the ContentArea, if not create a new group.
    b. Change the name of the group in "General".
    c. Create the Caption property, enter "General".
3. Add the fields.
    a. Add following fields in the General group
        i. Code
        ii. Name
        iii. Description
        iv. Duration
        v. Price
        vi. Type
        vii. Active
        viii. Instructor Code
        ix. Instructor Name

Task 3: Add an action to the page

High Level Steps

1. Add an action to the page.

Detailed Steps

1. Add an action.
    a. Verify that an action is created in the actions section. If not create an action.
    b. Change the name of the action "Resource Card".
2. Create the following action properties:
    a. RunObject: Page "Resource Card"
    b. RunPageLink: No.=FIELD(Instructor Code)
    c. Image: Relationship
    d. Promoted: Yes
    e. PromotedCategory: Process
    f. PromotedIsBig: Yes
3. Remove the trigger "OnAction()"

Lab 4.2 Create a List Page

Exercise 1: Create a List page for the Course table.

Exercise Scenario

Simon now creates the list page. He follows the list page naming standards and guarantees that users cannot edit information in this page. He also makes sure that when a user double-clicks a course, the Course Card page displays the selected course. Simon also makes sure that New, Edit, and View actions work as expected in the client.

Task 1: Creating a new page

High Level Steps

1. Create a new List page that has the source table Course.
2. Set page properties to display the caption according to the naming conventions for list pages.
3. Set page properties to specify that page type is List. Make sure that the list is not editable, and that Course Card page opens when users double-click a line in this page.

Detailed Steps

1. Create a new List page that has the source table Course.
    a. Create a new file with the name "CourseList.al".
    b. Type "tpage" and press the tab button.
    c. Change the ID to 50101 and the name to "Course List".
    d. Set the SourceTable property to "Course".
2. Set page properties to display the caption according to the naming conventions for list pages.
    a. Create the Caption property, enter "Course List".
3. Set page properties to specify that page type is List. Make sure that the list is not editable, and that Course Card page opens when users double-click a line in this page.
    a. Set Editable property, enter No.
    b. Set the PageType property, enter List.
    c. Set CardPageID property, enter "Course Card".
    d. Set the UsageCategory, enter "Documents".

Task 2: Adding controls

High Level Steps

1. Add a ContentArea container.
2. Add a Repeater.
3. Add the fields.

Detailed Steps

1. Add a ContentArea container.
    a. Verify that an area of the SubType "Content" is created in the layout

section. If not, create an area of the type "Content".

2. Add a Repeater.
    a. Remove the group in the ContentArea
    b. Add a Repeater with the name "General"
    c. Create the Caption property, enter "General".
3. Add the fields.
    a. Add following fields in the repeater
        i. Code
        ii. Name
        iii. Description

Task 3: Publish the pages

High Level Steps

1. Set the correct settings in the launch.json to start the Course List.
2. Publish the tables.

Detailed Steps

1. Set the correct settings in the launch.json file to start the Course List.
    a. Open the launch.json file.
    b. Change the "startupObjectId" to 50101.
    c. Change the "startupObjectType" to "Page"
2. Publish the pages.
    a. Click View > Command Palette…
    b. Type "AL: Publish".
    c. Test your extension by adding new records.

Lab 5.1 Create a Page Extension

Exercise 1: Create a page Extension for the Social Media fields.

Exercise Scenario

Simon wants to be able to input social media information via the customer card. He first creates a page extension that will extend the customer card and saves it.

Task 1: Creating a page extension

High Level Steps

1. Create a page extension that extends the Customer Card page.
2. Remove the actions and variable code.

Detailed Steps

1. Create a page extension that extends the Customer Card page.
    a. Create a new file with the name "SocialMediaCard.al".
    b. Type "tpageext" and press the tab button.
    c. Change the ID to 50110 and the name to "Social Media Card".
    d. Set the extends to "Customer Card"
2. Remove the actions and variable code.
    a. Remove the actions section and the variable "myInt".

Task 2: Adding controls

High Level Steps

1. Add the fields.

Detailed Steps

1. Add the fields to the layout section.
    a. Add following fields at the end of the General group
        i. Facebook
        ii. Twitter
        iii. Instagram
        iv. LinkedIn

Task 3: Publish the table and page extension

High Level Steps

1. Set the correct settings in the launch.json to start the Customer List.
2. Publish the tables.

Detailed Steps

1. Set the correct settings in the launch.json file to start the Customer List.
    a. Open the launch.json file.

    b. Change the "startupObjectId" to 22.
    c. Change the "startupObjectType" to "Page"
2. Publish the pages.
    a. Click View > Command Palette…
    b. Type "AL: Publish".
    c. Test your extension by opening a customer card. Your fields should appear in the "General" group.

Lab 6.1 Reading and Writing Files

Exercise Scenario

Isaac is a developer for Cronus International Ltd. and is learning how to work with files in Microsoft Dynamics 365 Business Central. He wants to be able to read and write pictures from an Item.

Exercise 1: Create a codeunit to read and write images

Task 1: Create a new AL Extension project

High Level Steps

4. Create a new AL Extension project.
5. Correct the settings in the launch.json file.
6. Change the settings in the app.json file.

Detailed Steps

3. Create a new AL Extension project
    a. Click View > Command Palette…
    b. Type "AL: Go!" in the search box and press ENTER.
    c. Ignore the authentication request by pressing ESC.
4. Correct the settings in the launch.json file.
    a. Click View > Explorer. The Explorer pane opens.
    b. Click the folder ".vscode".
    c. Click the file "launch.json".
    d. Change the "server" setting with the URL of your Microsoft Dynamics 365 Business Central test environment.
    e. Change the "serverInstance" setting with the instance name of your test environment (if needed).
5. Change the settings in the app.json file
    a. Click the app.json file in the Explorer pane.
    b. Change the "name" setting to "ReadingAndWritingFiles".
    c. Change the "publisher" setting to "Cronus International Ltd".

Task 2: Create a CodeUnit

High Level Steps

1. Add a new codeunit
2. Create a function to import item pictures
3. Create a function to export item pictures

Detailed Steps

1. Add a new codeunit
    a. Create a new file with the name ImageManagement.al
    b. Create a codeunit with ID 50110 with the name ImageManagement
2. Create a function to import item pictures
    a. Create a new procedure with the name "ImportItemPicture"
    b. Create a parameter Item of type Record Item
    c. Create following 3 variables

```
PicInStream: InStream;
FromFileName: Text;
OverrideImageQst: Label 'The existing picture will be replaced. Do
you want to continue?', Locked = false, MaxLength = 250;
```

    d. Add following code to the ImportItemPicture procedure

```
with Item do
begin
  if Picture.Count > 0 then
     if not Confirm(OverrideImageQst) then
        exit;

  if UploadIntoStream('Import', '', 'All Files (*.*)|*.*',
     FromFileName, PicInStream) then
  begin
    Clear(Picture);
    Picture.ImportStream(PicInStream, FromFileName);
    Modify(true);
  end;
end;
```

3. Create a function to export item pictures
    a. Create a new procedure with the name "ExportItemPicture"
    b. Create a parameter Item of type Record Item
    c. Create following 4 variables

```
PicInStream: InStream;
Index: Integer;
TenantMedia: Record "Tenant Media";
FileName: Text;
```

    d. Add following code to the ExportItemPicture procedure

```
with Item do
begin
  if Picture.Count = 0 then
```

```
      exit;

  for Index := 1 to Picture.Count do begin
      if TenantMedia.Get(Picture.Item(Index)) then begin
         TenantMedia.calcfields(Content);
         if TenantMedia.Content.HasValue then begin
            FileName := TableCaption + '_Image' + format(Index) +
            GetTenantMediaFileExtension(TenantMedia);

            TenantMedia.Content.CreateInStream(PicInstream);
            DownloadFromStream(PicInstream, '', '', '', FileName);
         end;
      end;
  end;
end;
```

e. Create a local procedure "GetTenantMediaFileExtension"
f. Add following code

```
local procedure GetTenantMediaFileExtension(var TenantMedia:
Record "Tenant Media"): Text;
begin
   case TenantMedia."Mime Type" of
            'image/jpeg' : exit('.jpg');
            'image/png' : exit('.png');
            'image/bmp' : exit('.bmp');
            'image/gif' : exit('.gif');
            'image/tiff' : exit('.tiff');
            'image/wmf' : exit('.wmf');
   end;
end;
```

Task 3: Create a new page extension that extends "Item Card"

High Level Steps

11. Create a new file, with the name ItemCardExt.al.
12. Create a page extension with ID 50110 and name ItemCardExtension.
13. Create 2 actions in the actions section. Actions should be added after "Functions".
14. Publish your solution and run the Item Card.

Detailed Steps

11. Create a new file, with the name ItemCardExt.al
12. Create a page extension with ID 50110 and name ItemCardExtension
13. Create 2 actions in the actions section. Actions should be added after "F&unctions"
    a. Add an action ImportItemPicture
    b. Set Caption to "Import Item Picture"
    c. Create the OnAction trigger

    d. Create a variable "ImageManagement" of type CodeUnit ImageManagement in the OnAction trigger.
    e. Run the ImportItemPicture procedure from the CodeUnit variable. Pass the current record (Rec) as parameter
    f. Add an action ExportItemPicture
    g. Set Caption to "Export Item Picture"
    h. Create the OnAction trigger
    i. Create a variable "ImageManagement" of type CodeUnit ImageManagement in the OnAction trigger.
    j. Run the ExportItemPicture procedure from the CodeUnit variable. Pass the current record (Rec) as parameter
14. Publish your solution and run the Item Card
    a. Run the Item Card to test your solution

Lab 7.1 Create an XMLPort to Export XML Data

Scenario

Simon is a developer working for CRONUS International Ltd. Now, Simon must create an XMLPort to export all the customer information from the Customer List page in the XML format.

Objectives

The objectives are:

- Create an XMLPort.
- Extend a page to run an XMLPort.

Simon first creates the XMLPort object that imports the document.

Task 1: Create a new AL Extension project

High Level Steps

1. Create a new AL Extension project.
2. Correct the settings in the launch.json file.
3. Change the settings in the app.json file.

Detailed Steps

1. Create a new AL Extension project
    a. Click View > Command Palette…
    b. Type "AL: Go!" in the search box and press ENTER.
    c. Ignore the authentication request by pressing ESC.
2. Correct the settings in the launch.json file.
    a. Click View > Explorer. The Explorer pane opens.
    b. Click the folder ".vscode".
    c. Click the file "launch.json".
    d. Change the "server" setting with the URL of your Microsoft Dynamics 365 Business Central test environment.
    e. Change the "serverInstance" setting with the instance name of your test environment (if needed).
3. Change the settings in the app.json file
    a. Click the app.json file in the Explorer pane.
    b. Change the "name" setting to "Customer XML".
    c. Change the "publisher" setting to "Cronus International Ltd".

Task 2: Create a new XMLPort

High Level Steps

1. Create a new XMLPort objects and set the properties to export in XML format.
2. Create an element of type Text, and name it "Root".
3. Under Root, create a child element of type Table, name it "Customer", and set its source to 18 (Customer).
4. Add attributes and child elements for Id, No., Language Code, Name, Address, Address 2, City, Post Code & Phone No.
5. Remove the requestpage section.

Detailed Steps

1. Create a new XMLport objects and set the properties to export in XML format.
    a. Create a new AL file with the name CustXML.al
    b. Use the code snippet "txmlport" to create a new XMLPort. Assign the ID 50100 and the name CustXML
    c. Set the XMLPort property "Direction" to "Export" (This specifies the XMLPort to be used only for export.)
    d. Set the XMLPort property "Format" to "Xml" (This specifies the external file to be an XML document.)
    e. Set the XMLPort property "FormatEvaluate" to "Xml"
2. Create an element of type Text, and name it "Root".
    a. Change the name of the first textelement (NodeName1) to "Root"
3. Under Root, create a child element of type Table, name it "Customer", and set its source to 18 (Customer).
    a. Change the first tableelement. Change the name (NodeName2) to Customer.
    b. Change the SourceTable (SourceTableName) to 18 (Customer)
4. Add attributes and child elements for Id, No., Language Code, Name, Address, Addres 2, City, Post Code & Phone No.

```
tableelement(Customer; Customer)
{

    fieldattribute(Id; Customer.Id) { }
    fieldattribute(Number; Customer."No.") { }
    fieldattribute(Language; Customer."Language Code") { }
    fieldelement(Name; Customer.Name) { }
    textelement(Address)
    {
        fieldelement(StreetAndNr; Customer.Address)
        {
            fieldattribute(Address2; customer."Address 2") { }
        }
        fieldelement(City; Customer.City)
        {
            fieldattribute(Zipcode; Customer."Post Code") { }
        }
    }
    fieldelement(Phone; Customer."Phone No.") { }}
```

5. Remove the requestpage section.

Task 3: Add action to the Customer List page

High Level Steps

1. Add a Page Extension for the Customer List page
2. Add an action container and an action.
3. Set the properties to run XMLport CustXML.
4. Deploy your solution.
5. Run the "Export to XML" action from the page and export the customer details into a file.

Detailed Steps

1. Add a Page Extension for the Customer List page
    a. Create a new AL file with the name CustXMLExt.al
    b. Use the code snippet "tpageext" to create a new Page Extension. Assign the ID 50100 and the name CustXMLExt
    c. Extend the page "Customer List"
2. Add an action.
    a. Add an action to the "actions" section. Add this after the "&Customer" action
    b. Give the action the name "Export to XML"
3. Set the properties to run XMLport CustXML.
    a. Set the RunObject property to "xmlport 50100".
4. Deploy your solution.
5. Run the "Export to XML" action from the page and export the customer details into a file.
    a. In the Microsoft Dynamics 365 Business Central application, open the customer list page.
    b. On the Actions tab in the ribbon, click Export to XML.
    c. Open or download the file, prompted via your web browser and review its contents.

Lab 7.2 Create an XMLport to Export Variable Text

Scenario

Simon is a developer working for CRONUS International Ltd. Simon has already created an XMLPort to export the customer details in XML format. Some internal systems of CRONUS International are legacy systems that cannot handle XML data. Simon must create an XMLPort to export all the customer details from the Customer List page in the comma-separated text format.

Objectives

The objectives are:

- Create an XMLPort to export data in variable text format.
- Extend a page to run an XMLPort.

Task 1: Create the XMLport

High Level Steps

1. Copy XMLport CustXML as XMLport CustCSV (50101).
2. Modify the properties to export data in variable text format.

Detailed Steps

1. Copy XMLport CustXML as XMLport CustCSV (50101).
    a. Create a new AL file with the name CustCSV.al
    b. Copy the content of the file CustXML.al in the new file
    c. Change the ID to 50101 and the name to CustCSV
2. Modify the properties to export data in variable text format.
    a. Set the Direction property to Export (This specifies the XMLport to be used only for export.)
    b. Set the Format property to Variable Text (This specifies the external file to be a variable text document.)
    c. Remove the FormatEvaluate property

Task 2: Add action to run the XMLport

High Level Steps

1. Add an extra action to the Page Extension "CustXMLExt.al".
2. Set the properties to run XMLport Course Export Variable.
3. Deploy your solution.
4. Run the "Export to CSV" action from the page and export the customer details into a file.

Detailed Steps

1. Add an extra action to the Page Extension "CustXMLExt.al".

    a. Add an action to the "actions" section. Add this after the "fExport To XML" action
    b. Give the action the name "Export to CSV"
2. Set the properties to run XMLport Course Export Variable.
    a. Set the RunObject property to "xmlport 50101".
    b. Close the Properties window.
3. Deploy your solution.
4. Run the "Export to CSV" action from the page and export the customer details into a file.
    a. In the Microsoft Dynamics 365 Business Central application, open the customer list page.
    b. On the Actions tab in the ribbon, click Export to CSV.
    c. Open or download the file, prompted via your web browser and review its contents

Lab 8.1 Using a Query for a Chart

Scenario

Susan, the sales order processor at CRONUS International Ltd., needs a chart that shows the top ten customers by revenue. Isaac and Simon, the consultants at the ISV Company that implements Microsoft Dynamics 365 Business Central for CRONUS International Ltd., will help Susan by creating and customizing the necessary objects, and then configure Microsoft Dynamics 365 Business Central according to Susan's requirements.

Objectives

The objectives of this lab are:

- Learn how to create a new query.
- Understand how to set query and data item properties.
- Understand how to use totaling.
- Understand how to use a query in a chart.

Exercise 1: Creating a query

Exercise Scenario

Isaac, the business software developer, creates a new query object that selects the data that Susan wants to see in a chart part on her role center. The query will join the data from the Customer and Cust. Ledger Entries table's, show only those customers who have ledger entries of type Invoice or Credit Memo, and display the Amount (LCY) column totaled by the Sum method.

Task 1: Create a new AL Extension project

High Level Steps

1. Create a new AL Extension project.
2. Correct the settings in the launch.json file.
3. Change the settings in the app.json file.

Detailed Steps

1. Create a new AL Extension project
    a. Click View > Command Palette…
    b. Type "AL: Go!" in the search box and press ENTER.
    c. Ignore the authentication request by pressing ESC.
2. Correct the settings in the launch.json file.
    a. Click View > Explorer. The Explorer pane opens.
    b. Click the folder ".vscode".
    c. Click the file "launch.json".
    d. Change the "server" setting with the URL of your Microsoft Dynamics 365 Business Central test environment.
    e. Change the "serverInstance" setting with the instance name of your test environment (if needed).

3. Change the settings in the app.json file
    a. Click the app.json file in the Explorer pane.
    b. Change the "name" setting to "Query Cust. Ledger Entries".
    c. Change the "publisher" setting to "Cronus International Ltd".

Task 2: Create a new query object

High Level Steps

1. Create a new query object.
2. Select table 18, Customer as the first data item.

Detailed Steps

1. Create a new query object.
    a. Create a new AL file with the name CustLedgEntrQuery.al
    b. Use the snippet tquery
    c. Change the id in 50100 and the name in CustLedgEntrQuery
2. Select table 18, Customer as the first data item.
    a. Set the SourceTableName of the first dataitem to Customer
    b. Set the DataItemName of the first dataitem to Customer

Task 3: Add another data item

High Level Steps

1. Select table 21, Cust. Ledg. Entry as the second data item, indented under Customer.
2. Set the properties for Cust_Ledger_Entry data item to link to the Customer data item, where the Customer No. field of Cust_Ledger_Entry is equal to the No. field of the Customer table, and to exclude the customers for which there are no ledger entries.
3. Filter the data item to show only entries of type Invoice and Credit Memo.

Detailed Steps

1. Select table 21, Cust. Ledg. Entry as the second data item, indented under Customer.
    a. Create a new dataitem in the Customer dataitem.
    b. Set the SourceTableName of the first dataitem to Cust. Ledg. Entry
    c. Set the DataItemName of the first dataitem to CustLedgEntry
2. Set the properties for Cust_Ledger_Entry dataitem to link to the Customer data item, where the Customer No. field of Cust_Ledger_Entry is equal to the No. field of the Customer table, and to exclude the customers for which there are no ledger entries.
    a. Set the property DataItemLink = "Customer No." = Customer."No."
    b. Set the property SqlJoinType = InnerJoin
3. Filter the data item to show only entries of type Invoice and Credit Memo.
    a. Set the DataItemTableFIlter property = "Document Type" = filter(Invoice|"Credit Memo")

Task 4: Add columns to the query

High Level Steps

1. Add columns for the No., Name and Customer Posting Group fields to the Customer data item.
2. Add a column for field Amount (LCY) to the Cust_Ledger_Entry data item.
3. Specify the Sum totaling for the Amount (LCY) column.

Detailed Steps

1. Add columns for the No., Name and Customer Posting Group fields to the Customer data item.
    a. In the Customer Data Item, add a column for No., Give this the name "No"
    b. In the Customer Data Item, add a column for Name, Give this the name "Name"
    c. In the Customer Data Item, add a column for Customer Posting Group, Give this the name "CustomerPostingGroup"
2. Add a column for field Amount (LCY) to the Cust_Ledger_Entry data item.
    a. In the Customer Ledger Entry Data Item, add a column for Amount LCY., Give this the name "Sum_Amount_LCY"
3. Specify the Sum totaling for the Amount (LCY) column.
    a. Set the property Method to Sum for the Amount (LCY).

Task 5: Set query properties

High Level Steps

1. Set the query properties to order the results by sum of Amount (LCY) in descending order.
2. Set the query properties to return only the top 10 rows.

Detailed Steps

1. Set the query properties to order the results by sum of Amount (LCY) in descending order.
    a. Set the query property OrderBy = descending(Sum_Amount_LCY)
2. Set the query properties to return only the top 10 rows.
    a. Set the TopNumberOfRows property to 10

Exercise 2: Creating a chart

Exercise Scenario

Simon, the consultant, creates a new generic chart in Microsoft Dynamics 365 Business Central, to show the data from the query that was created by Isaac in the previous exercise. The chart will be a column chart and will show Sum_Amount_LCY as a measure that is aggregated by the Sum method over the customer number as the X-axis dimension.

Task 1: Create a new chart

High Level Steps

1. Open the Microsoft Dynamics 365 Business Central Web client.
2. Create a new chart.

Detailed Steps

1. Open the Microsoft Dynamics 365 Business Central Web client.
2. Create a new chart setup.
    a. Go to the search box and type "Generic Charts."
    b. On the Home tab, in the New group, click New.

Task 2: Configure the Chart Setup Card

High Level Steps

1. Set the new chart ID to 50100-01 and name it as Top 10 Cust. By Revenue.
2. Set the data source of the chart to query 50100, Top 10 Cust. By Revenue.
3. Define the Sum_Amount_LCY column as the measure. Set Aggregation to Sum, and Graph Type to Column.
4. Define the No column as the X-Axis dimension.

Detailed Steps

1. Set the new chart ID to 50100-01 and name it as Top 10 Cust. By Revenue.
    a. In the Generic Chart Setup Card page, in the ID field, enter "50100-01".
    b. In the Name field, enter "Top 10 Cust. by Revenue".
2. Set the data source of the chart to query 50100, Top 10 Cust. By Revenue.
    a. In the Chart Setup Card window, in the Source Type field, select Query.
    b. In the Source ID field, enter "50100".
3. Define the Sum_Amount_LCY column as the measure. Set Aggregation to Sum, and Graph Type to Column.
    a. On the Measures (Y-Axis) FastTab, in the Data Column field for Required Measure row, select Sum_Amount_LCY.
    b. In the Aggregation field, select Sum
    c. In the Graph Type field, select Column.
4. Define the No column as the X-Axis dimension. a. On the Dimensions (X- and Z-Axes) FastTab, in the X-Axis field, enter "No".
    a. In the X-Axis Title field, enter "No".
    b. Click OK to close the page.

Results

A new chart setup record that shows data from query 50100, Top 10 Cust. By Revenue.

Lab 8.2 Using Queries in AL

Scenario

Isaac and Simon, the consultants at the ISV Company that implements Microsoft Dynamics 365 Business Central for CRONUS International Ltd., wants to see data from multiple tables into a single page. They want to use queries to carry out the desired task.

Task 1: Create a Query

High Level Steps

1. Create a new Query
2. Save the Query as 50101,name it "ZY Purchase Order Query"

Detailed Steps

a. Create a new Query
b. Create a new AL file with the name ZYPurchaseOrderQuery.al
c. Use the snippet tquery
d. Change the id to 50101 and the name to "ZY Purchase Order Query"
e. Publish it and try to open Query directly with URL or via launch.json file. The data can be displayed.

```
query 50101 "ZY Purchase Order Query"
{
    Caption = 'ZY Purchase Order Query';
    OrderBy = Descending(Buy_from_Vendor_No_);

    elements
    {
        dataitem(Purchase_Header; "Purchase Header")
        {
            column(Buy_from_Vendor_No_; "Buy-from Vendor No.")
            {
            }
            column(Buy_from_Vendor_Name; "Buy-from Vendor Name")
            {
            }
            column(Order_Date; "Order Date")
            {
            }
            column(Currency_Code; "Currency Code")
            {
            }
            column(Amount_Including_VAT; "Amount Including VAT")
```

```
            {
            }
            dataitem(Purchase_Line; "Purchase Line")
            {
                DataItemLink = "Document Type" =
Purchase_Header."Document Type",
                "Document No." = Purchase_Header."No.";
                column(No_; "No.")
                {
                }
                column(Description; Description)
                {
                }
                column(Quantity; Quantity)
                {
                }
                column(Amount; Amount)
                {
                }
                dataitem(Item; Item)
                {
                    DataItemLink = "No." = Purchase_Line."No.";
                    column(Inventory; Inventory)
                    {
                    }
                }
            }
        }
    }
}
```

Task 2: Create a Temporary Table

High Level Steps

1. Create a Temporary Table
2. Save the Temporary Table as 50105,name it "ZYPurchaseOrderQueryTable"

Detailed Steps

1. Create a new Temporary Table
2. Create a new AL file with the name ZYPurchaseOrderQueryTable.al
3. Use the snippet ttable
4. Set the TableType property to Temporary.
5. Change the id to 50105 and the name to "ZYPurchaseOrderQueryTable"

```
table 50105 ZYPurchaseOrderQueryTable
{
    DataClassification = CustomerContent;
    TableType = Temporary;

    fields
    {
        field(1; RowNo; Integer)
        {
            Caption = 'Row No.';
            DataClassification = CustomerContent;
        }
        field(10; "Buy-from Vendor No."; Code[20])
        {
            Caption = 'Buy-from Vendor No.';
            DataClassification = CustomerContent;
        }
        field(20; "Buy-from Vendor Name"; Text[100])
        {
            Caption = 'Buy-from Vendor Name';
            DataClassification = CustomerContent;
        }
        field(30; "Order Date"; Date)
        {
            Caption = 'Order Date';
            DataClassification = CustomerContent;
        }
        field(35; "Currency Code"; Code[10])
        {
            Caption = 'Currency Code';
            TableRelation = Currency;
            DataClassification = CustomerContent;
        }
        field(40; "Amount Including VAT"; Decimal)
        {
            AutoFormatExpression = "Currency Code";
            AutoFormatType = 1;
            Caption = 'Amount Including VAT';
            DataClassification = CustomerContent;
        }
        field(50; "No."; Code[20])
        {
            Caption = 'No.';
            DataClassification = CustomerContent;
        }
        field(60; Description; Text[100])
        {
```

```
            Caption = 'Description';
            DataClassification = CustomerContent;
        }
        field(70; Quantity; Decimal)
        {
            Caption = 'Quantity';
            DecimalPlaces = 0 : 5;
            DataClassification = CustomerContent;
        }
        field(80; Amount; Decimal)
        {
            Caption = 'Amount';
            AutoFormatExpression = "Currency Code";
            AutoFormatType = 1;
            DataClassification = CustomerContent;
        }
        field(90; Inventory; Decimal)
        {
            Caption = 'Inventory';
            DecimalPlaces = 0 : 5;
            DataClassification = CustomerContent;
        }

    }

    keys
    {
        key(PK; RowNo)
        {
            Clustered = true;
        }
    }
}
```

Task 3: Create a new page to show the fields.

High Level Steps

1. Create a new page
2. Save the Page as 50106,name it "ZYPurchaseOrderQueryPage"

Detailed Steps

1. Create a new Page
2. Create a new AL file with the name ZYPurchaseOrderQueryPage.al
3. Use the snippet tpage

4. Transfer the data in Query to the table
5. Change the id to 50106 and the name to "ZYPurchaseOrderQueryPage"
6. Publish it

```
page 50106 ZYPurchaseOrderQueryPage
{
    Caption = 'ZY Purchase Order Query';
    PageType = List;
    UsageCategory = Lists;
    ApplicationArea = All;
    SourceTable = ZYPurchaseOrderQueryTable;

    layout
    {
        area(Content)
        {
            repeater(Group)
            {

                field(RowNo; Rec.RowNo)
                {
                    ApplicationArea = All;
                }
                field("Buy-from Vendor No."; Rec."Buy-
from Vendor No.")
                {
                    ApplicationArea = All;
                }
                field("Buy-from Vendor Name"; Rec."Buy-
from Vendor Name")
                {
                    ApplicationArea = All;
                }
                field("Currency Code"; Rec."Currency Code")
                {
                    ApplicationArea = All;
                }
                field("Amount Including VAT"; Rec."Amount Includ
ing VAT")
                {
                    ApplicationArea = All;
                }
                field("Order Date"; Rec."Order Date")
                {
                    ApplicationArea = All;
```

```
                }
                field("No."; Rec."No.")
                {
                    ApplicationArea = All;
                }
                field(Description; Rec.Description)
                {
                    ApplicationArea = All;
                }
                field(Quantity; Rec.Quantity)
                {
                    ApplicationArea = All;
                }
                field(Amount; Rec.Amount)
                {
                    ApplicationArea = All;
                }
                field(Inventory; Rec.Inventory)
                {
                    ApplicationArea = All;
                }
            }
        }
    }

    trigger OnOpenPage()
    var
        ZYPurchaseOrderQuery: Query "ZY Purchase Order Query";
    begin
        if ZYPurchaseOrderQuery.Open() then begin
            while ZYPurchaseOrderQuery.Read() do begin
                Rec.Init();
                Rec.RowNo := Rec.RowNo + 1;
                Rec."Buy-
from Vendor No." := ZYPurchaseOrderQuery.Buy_from_Vendor_No_;
                Rec."Buy-
from Vendor Name" := ZYPurchaseOrderQuery.Buy_from_Vendor_Name;
                Rec."Currency Code" := ZYPurchaseOrderQuery.Curr
ency_Code;
                Rec."Amount Including VAT" := ZYPurchaseOrderQue
ry.Amount_Including_VAT;
                Rec."Order Date" := ZYPurchaseOrderQuery.Order_D
ate;
                Rec."No." := ZYPurchaseOrderQuery.No_;
                Rec.Description := ZYPurchaseOrderQuery.Descript
ion;
                Rec.Quantity := ZYPurchaseOrderQuery.Quantity;
```

```
                Rec.Amount := ZYPurchaseOrderQuery.Amount;
                Rec.Inventory := ZYPurchaseOrderQuery.Inventory;
                Rec.Insert();
            end;
            ZYPurchaseOrderQuery.Close();
        end;
    end;
}
```

Lab 9.1 Creating a Basic Report

Scenario

Simon is a developer working for CRONUS International Ltd. CRONUS International. Simon is asked to create a report to list all customers. This report must have a RDLC and Word Layout.

Exercise 1: Build the report.

Exercise Scenario

Create a report that has the following requirements:

- Create this report with these information:
  - Prints all Customer information.
  - Create RDLC report layout
  - Create Word report layout

Task 1: Create an extension for the Customer List.

High Level Steps

1. Create the page extension.

Detailed Steps

1. Create the page extension.
   a. In Visual Studio Code, add a new file, with the name MyCustomerListExtension.al
   b. Create an extension for the Customer List, using the following code:

      pageextension 50123 MyExtension extends "Customer List"
      {

```
    trigger OnOpenPage();
    begin
      report.Run(Report::LAB_CustomerList);
    end;
}
```

Task 2: Create a new report.

High Level Steps

1. Create the dataset.

Detailed Steps

1. Create the dataset.
    a. In Visual Studio Code, add a new file with the name: LAB_CustomerList.al
    b. Create the dataset for the report, using the following code:

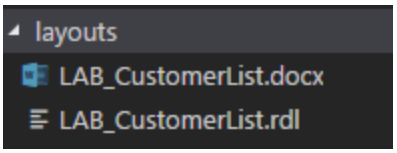
```
report 50105 LAB_CustomerList
{
  CaptionML = ENU = 'LAB_CustomerList';
  RDLCLayout = 'layouts/LAB_CustomerList.rdl';
  WordLayout = 'layouts/LAB_CustomerList.docx';

  PreviewMode = Normal;
  WordMergeDataItem = Customer;

  dataset
  {
    dataitem(Customer; Customer)
    {
      DataItemTableView = SORTING(Name);

      column(No; "No.")
      {
        IncludeCaption = true;
      }
      column(Name; Name)
      {
        IncludeCaption = true;
      }
      column(Balance; "Balance (LCY)")
      {
        IncludeCaption = true;
      }
    }
  }
}
```

    c. Add a subfolder named layouts in the project
    d. Build the report, using Ctrl+Shift+B, this will generate the RDLC and WORD layouts:



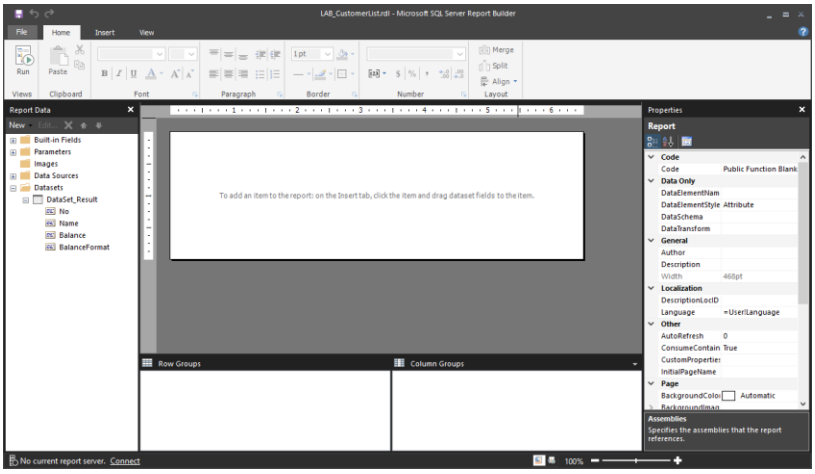
Task 3: Create a RDLC Layout

High Level Steps

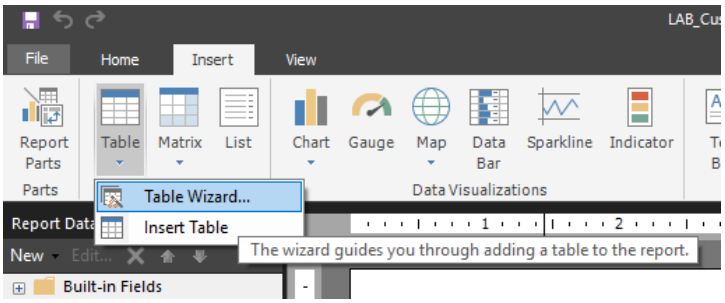
1. Create a layout for the report.

Detailed Steps

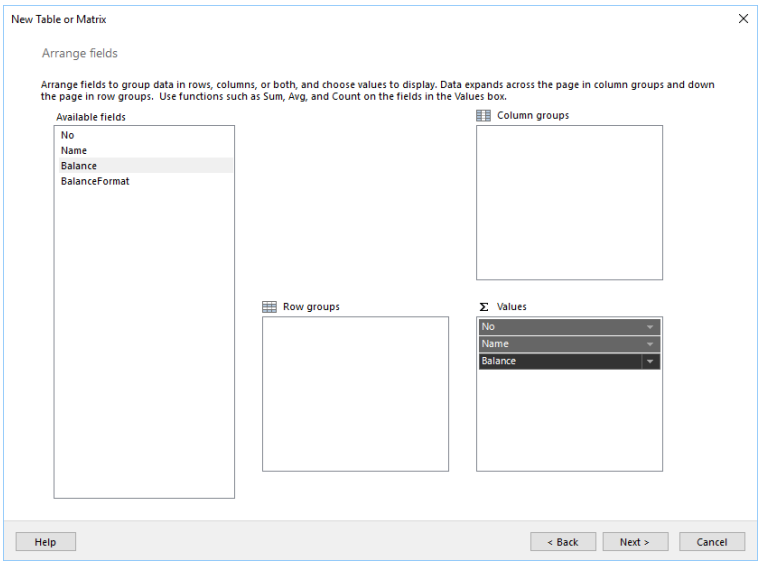
1. Create a layout for the report.
    a. Right Click the LAB_CustomerList.rdl file and select: Open Externally
    b. Report Builder now opens:



    c. In the Insert menu, select Table, Table Wizard:



    d. In the New Table or Matrix window, select DataSet_Result and click Next
    e. Drag the No, Name and Balance fields onto the Values section:



f. Click Next
g. Click Next
h. Click Finish
i. Select the Name Column and make it wider
j. Select the first row and align the content to the left
k. Save and Close Report Builder

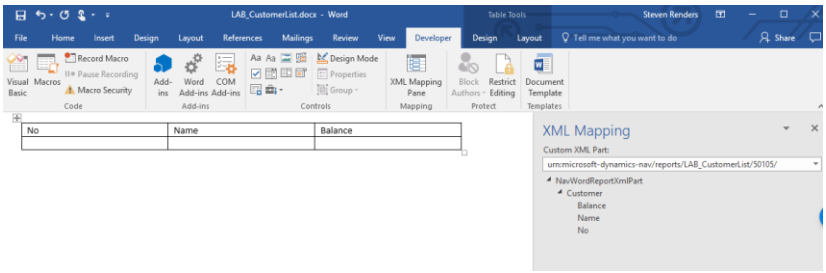
Task 4: Create a Word Layout

High Level Steps

1. Create a layout for the report.

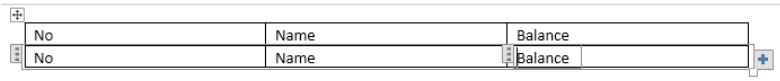
Detailed Steps

1. Create a layout for the report.
    a. Right Click the LAB_CustomerList.docx file and select: Open Externally
    b. Word now opens:
    c. Create Table in Word with two rows and three columns:
        i. On tab Insert, click on Table
        ii. Choose two rows and three columns
    d. In the ribbon right click and select Customize the Ribbon
        i. In the column on the right, enable Developer
        ii. Click OK
    e. On Developer tab in the ribbon, click on XML Mapping Pane.
    f. On Custom XML Part, choose: *urn:microsoft-dynamics-nav/reports/LAB_CustomerList/50105/*
    g. Expand Customer
    h. On the first row, type No, Name, Balance in the first, second and third column:

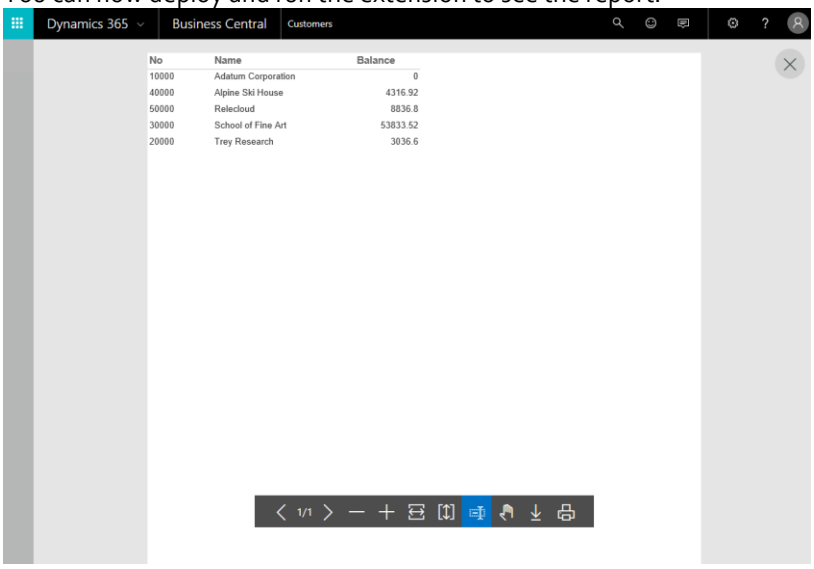


i. Mark the second row.
j. Right-click on Customer; choose Insert Content Control > Repeating.

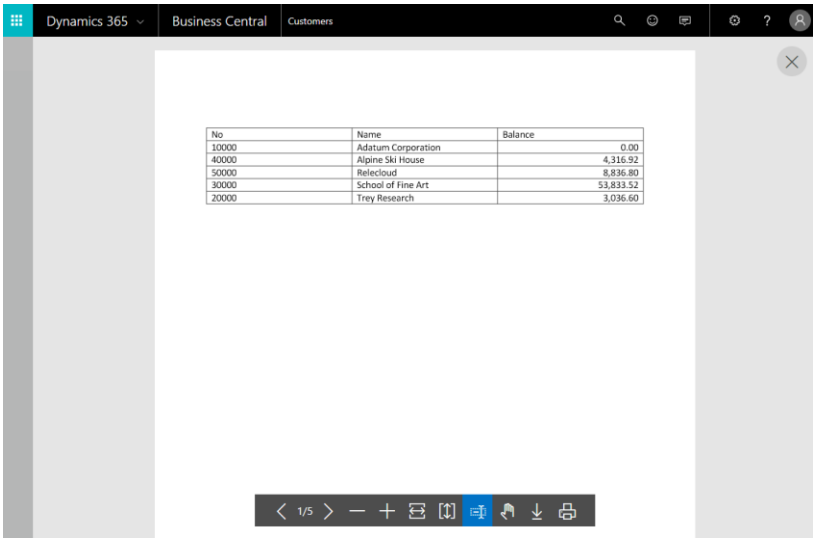
k. Set the following DataSet Result fields using right-click in customer field and Insert Content Control > Plain Text in the appropriate columns of the second row of the table:
    i. No
    ii. Name
    iii. Balance



l. Close Word document and click Save.
m. You can now deploy and run the extension to see the report:
n.



o.


Dynamics 365
Business Central
Customers
No
Name
Balance
10000
Adatum Corporation
0.00
40000
Alpine Ski House
4,316.92
50000
Relecloud
8,836.80
30000
School of Fine Art
53,833.52
20000
Trey Research
3,036.60
1/5

Lab 9.2 Creating a Processing Only Report

Scenario

Simon will create a non-printing report that enable a user to adjust the Name 2 field in the customer table. The user enters the name by which to adjust the field on the request page. The user can use the filters on the request page to filter the data and apply the adjustment to only the filtered data.

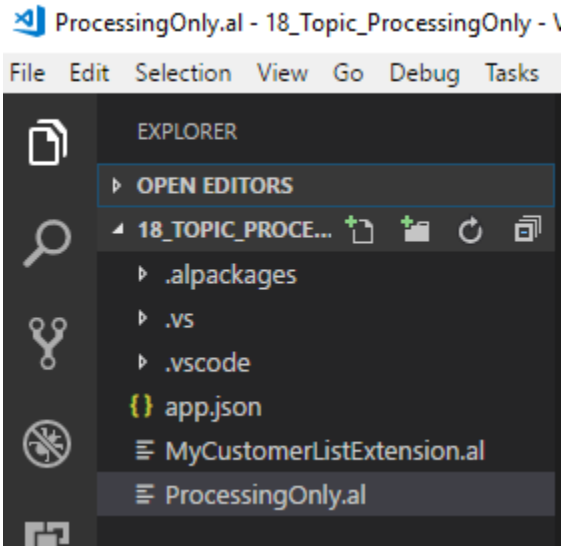
Task 1: Defining the Data Model

High Level Steps

1. Define the Data Model

Detailed Steps

1. To Define the Data Model:
    a. In Visual Studio Code, add e new file to the project and name it ProcessingOnly.al



    b. Use the treport snippet to create the report dataset:

```
0 references
report Id MyReport
{
    dataset
    {
        dataitem(DataItemName; SourceTableName)
        {
            column(ColumnName; SourceFieldName)
            {

            }
        }
    }

    requestpage
    {
        layout
        {
            area(content)
            {
                group(GroupName)
                {
                    field(Name; SourceExpression)
                    {

                    }
                }
            }

            actions
            {
                area(processing)
                {
                    action(ActionName)
                    {

                    }
                }
            }
        }

        var
            myInt: Integer;
}
```

c. In the var section declare the following variables:

Name2ToApply : Text;
Customer2 : Record Customer;
ReplaceExisting : Boolean;
Counter : Integer;

```
var
    2 references
    Name2ToApply : Text;
    3 references
    Customer2 : Record Customer;
    2 references
    ReplaceExisting : Boolean;
    3 references
    Counter : Integer;
```

d. In the dataset section, declare a dataitem for the Customer table:

```
dataset
{
    3 references
    dataitem(Customer; Customer)
    { ···
    }
}
```

e. In the Customer dataitem add an OnPreDataItem trigger with the following code:

```
trigger OnPreDataItem();
begin
  clear(Customer);
  clear(Counter);
  if not ReplaceExisting then
   Customer.SetRange("Name 2",'');
end;
```

f. In the Customer dataitem, add an OnAfterGetRecord trigger with the following code:

```
trigger OnAfterGetRecord();
begin
  if Customer2.get(Customer."No.") then
  begin
    Customer2."Name 2" := Name2ToApply;
    Customer2.Modify;
    Counter += 1;
  end;
end;
```

g. In the Customer dataitem, add an OnPostDataItem trigger with the following code:

```
trigger OnPostDataItem();
```

```
begin
  Message('Ready!, %1 Customers were updated.',Counter);
end;
```

h. The customer dataitem should now look as follows:

```
dataitem(Customer; Customer)
{
        trigger OnPreDataItem();
        begin
            clear(Customer);
            clear(Counter);
            if not ReplaceExisting then
              Customer.SetRange("Name 2",'');
        end;

        trigger OnAfterGetRecord();
        begin
            if Customer2.get(Customer."No.") then
            begin
                Customer2."Name 2" := Name2ToApply;
                Customer2.Modify;
                Counter += 1;
            end;
        end;

        trigger OnPostDataItem();
        begin
            Message('Ready!, %1 Customers were updated.',Counter);
        end;
}
```

Task 2: Adding Controls to the Request Page

High Level Steps

1. Adding controls to the request page

Detailed Steps

1. To add controls to the request page:
    a. In the requestpage section, add the following code:

```
requestpage
{
  SaveValues = true;

  layout
  {
    area(content)
    {
      group(Options)
      {
        Caption = 'Options';
        field(Name2ToApply;Name2ToApply)
```

```
                {
                    ApplicationArea = All;
                    Caption = 'Name2ToApply';
                    ToolTip = 'Specifies Name2 To Apply.';
                }
                field(ReplaceExisting;ReplaceExisting)
                {
                    ApplicationArea = All;
                    Caption = 'ReplaceExisting';
                    ToolTip = 'ReplaceExisting';
                }
            }
        }
    }
}
```

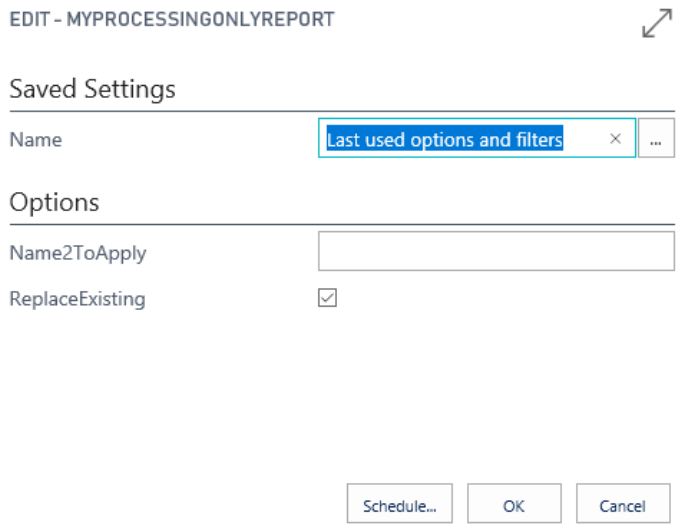
Task 3: Testing the Report

High Level Steps

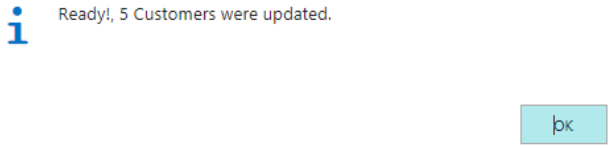
1. Testing the Report

Detailed Steps

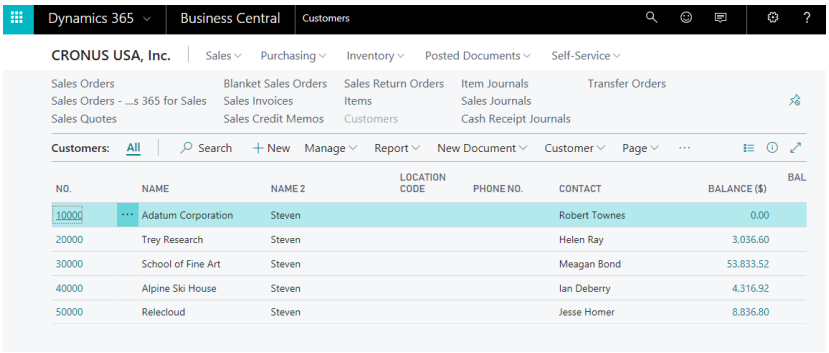
1. To test the report:
    a. Deploy the project by pressing the F5 key:
    b. The Request Page for the report displays and looks af follows:



    c. In the Name2ToApply field enter your name, for example Steven.
    d. Select the RelaceExisting option, to make sure existing information in the Name 2 field will be overwritten.
    e. Select the OK button to run the report.
    f. A message appears, like for example the following:



g. Select OK and Go to the Customer List page.
h. The Name 2 field should now contain the name you entered in the request page:



Lab 10.1 Use Microsoft Dynamics 365 Business Central as a Power BI Data Source

Exercise Scenario

Isaac is a Business Intelligence developer for Cronus International Ltd. Isaac would like to use Power BI Desktop and connect it to Microsoft Dynamics 365 Business Central to use it as a data source.

High Level Steps

1. Download and install Power BI Desktop
2. Add Microsoft Dynamics 365 Business Central as a data source in Power BI Desktop

Detailed Steps

1. Download and install Power BI Desktop
    a. Open your browser and navigate to: https://powerbi.microsoft.com/en-us/desktop/
    b. On the Power BI website, click the download button.
    c. Wait for the download to finish and then install the Power BI Desktop software.
2. Add Microsoft Dynamics 365 Business Central as a data source in Power BI Desktop
    a. In Power BI Desktop, in the left navigation pane, click **Get Data**.
    b. In the **Get Data** window, click **Online Services**, click **Microsoft Dynamics 365 Business Central**, and then click the **Connect** button.
    c. Power BI displays a wizard that will guide you through the connection process. The first step will be to enter an OData URL and the company name that is associated with your Dynamics 365 Financials account.
    d. For the OData URL, you can copy the OData V4 URL of any of the web services that are listed in the Web Services page in Microsoft Dynamics 365 Business Central, such as https://mycompany.financials.dynamics.com:7048/MS/ODataV4/.
    e. As company name, use the name that is shown in the **Name** field in the **Company Information** window in Microsoft Dynamics 365 Business Central. If your Microsoft Dynamics 365 Business Central contains multiple companies, choose the relevant company name from the list in the **Companies** window. In both cases, make sure that the name that you specify in the Power BI wizard matches exactly the text shown in Microsoft Dynamics 365 Business Central, such as "My Company".
    f. Once you have entered the information, click **OK**.
    g. Note: If there are other authentication options available in the left-hand navigation, choose **Basic**.
    h. Enter your username and password. You can find this information in the **Users** window in Microsoft Dynamics 365 Business Central. Use the web access key as your password. (For example, your username is ADMIN, and the web service access key that serves as your password is: EgzeUFQ9Uv005OolUMyqCz01ueUW9yRF3SsLU=.)

i. Click the **Connection** button to continue. The Power BI wizard shows a list of Microsoft Dynamics 365 Business Central data sources. These data source represent all the web services that you have published from your Financials.
j. Alternatively, create a new web service URL in Microsoft Dynamics 365 Business Central by using the **Create Data Set** action in the **Web Services** page, using the **Set Up Reporting** Assisted Setup guide, or by choosing the **Edit in Excel** action in any lists.
k. Specify the data you want to add to your data model, and then click the **Load** button.
l. Repeat the previous steps to add additional Microsoft Dynamics 365 Business Central data to your Power BI data model.
m. Once you have successfully connected to Microsoft Dynamics 365 Business Central, you will not be prompted again for the OData URL, username, or password.
n. Once the data is loaded it will appear in the right navigation on the page. At this point, you have successfully connected to your Dynamics 365 data and are ready to begin building your Power BI report.