NAV TECH DAYS 2018

mibuso.com

# What's new in Developing for Business Central

ALEX TOADER, ESBEN NYHUUS KRISTOFFERSEN
MICROSOFT DEVELOPMENT CENTER COPENHAGEN

NAV TECH DAYS 2018
mibuso.com

# Improved Extensibility

“

To enable partners to *efficiently* develop *maintainable* solutions on top of our application while still allowing Microsoft to *evolve* the application.

”

# (Extensible) Enums

New top-level type
List of possible values (outcomes)
Values have an ordinal value used for persistence
Values can have captions
Enums can be used instead of options
Long-term replacement for the option type

# (Extensible) Enums - Declaration

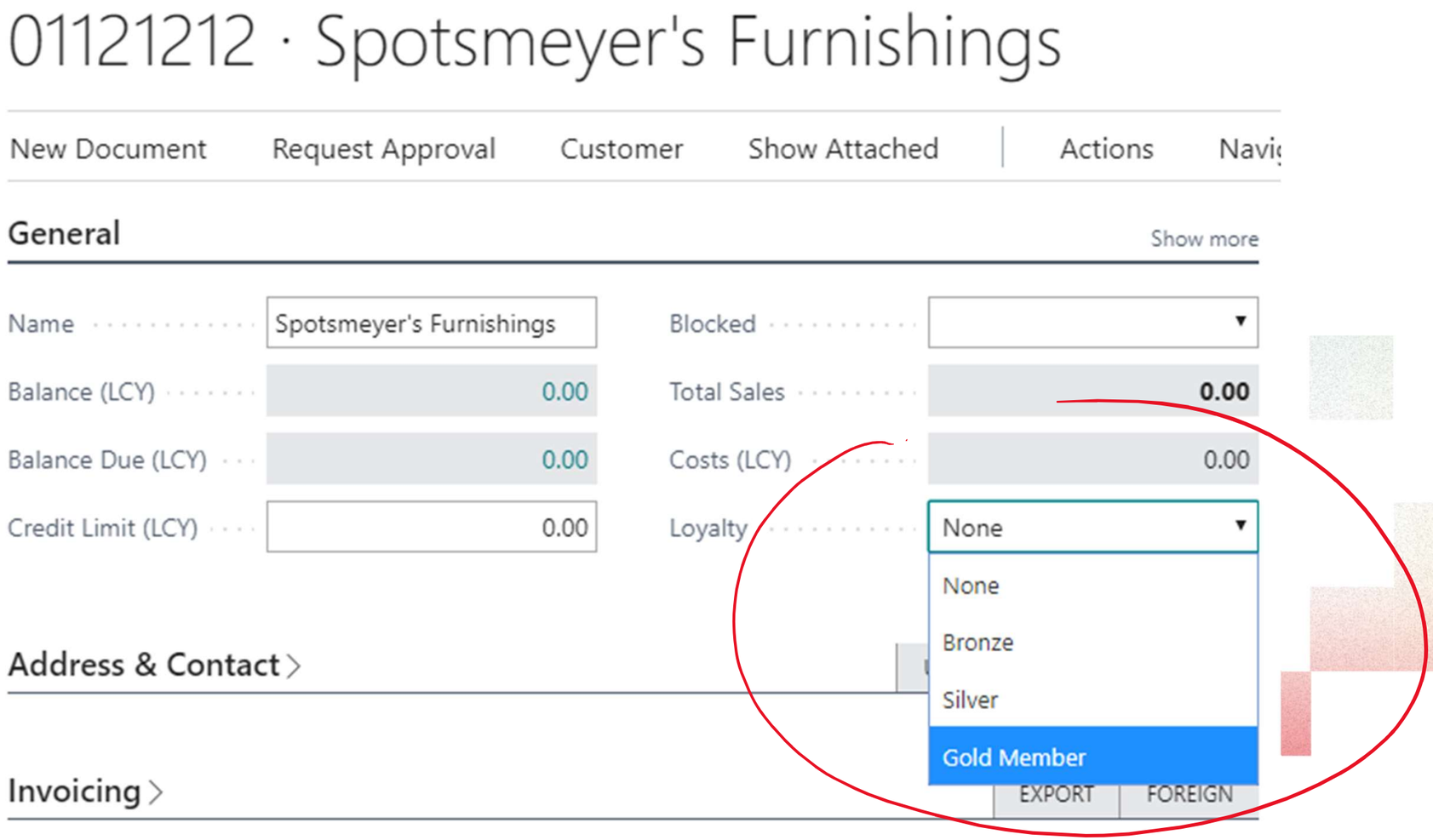

```
enum 50100 Loyalty
{
  Extensible = true;

  value(0; None) { }
  value(1; Bronze) { }
  value(2; Silver) { }
  value(3; Gold)
    {
      Caption = 'Gold Member';
    }
}
```

# (Extensible) Enums - Usage


```
tableextension 50100 CustomerExt extends Customer
{
    fields
    {
        field(50100; Loyalty; enum Loyalty) { }
    }

    procedure UseEnum(p: enum Loyalty)
    var
        v: enum Loyalty;
    begin
        if p = Loyalty::Gold then
            v := p;
    end;
}
```

# Demo: Enum

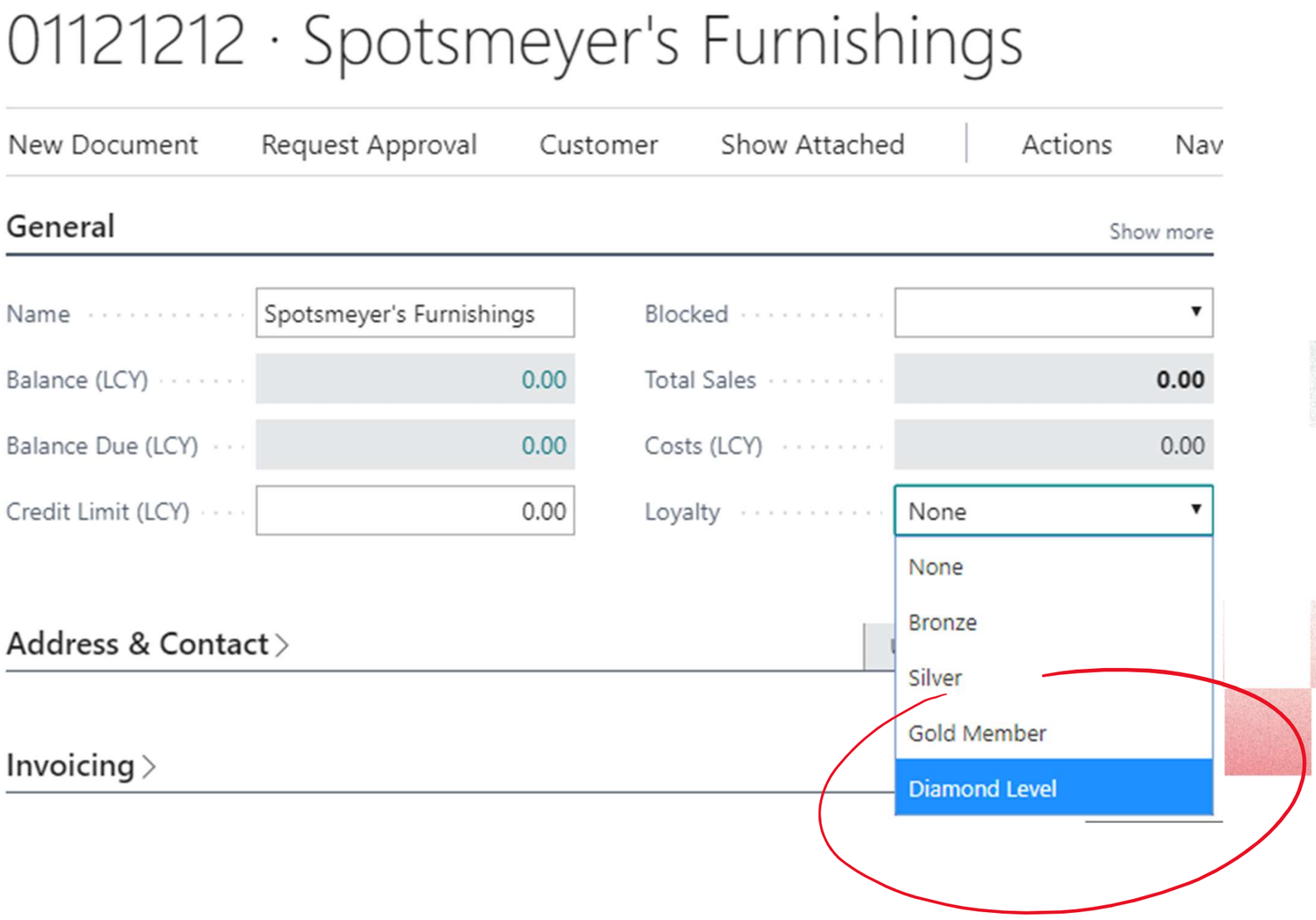
# (Extensible) Enums - UI



# Extensible Enums - Extensions

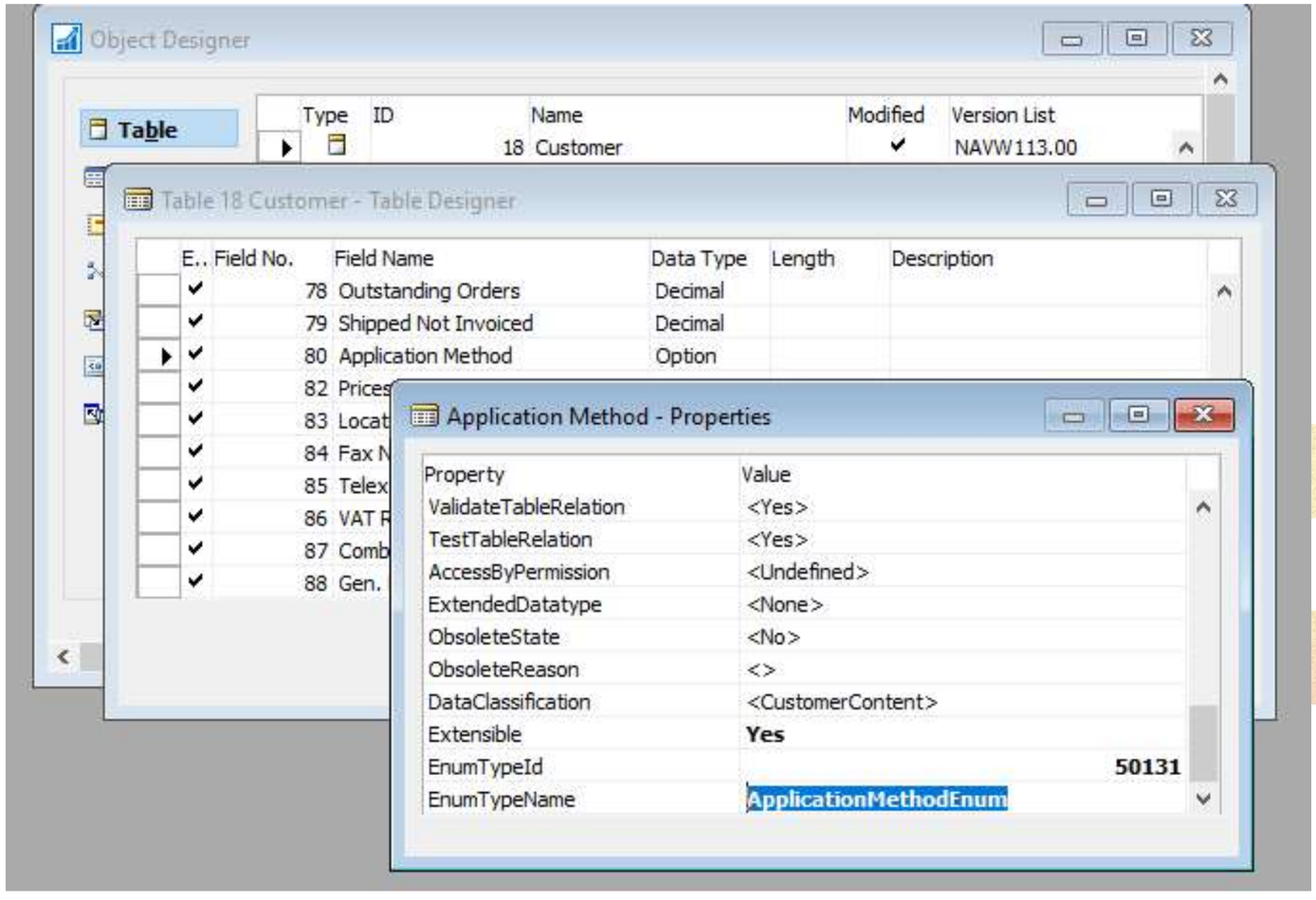
```
enumextension 50100 DiamondLoyalty extends Loyalty
{
    value(50100; Diamond)
    {
        Caption = 'Diamond Level’;
    }
}
```

# Extensible Enums - UI



# Extensible Enums - C/AL Integration

- Table Field Options can be marked as Extensible
- Requires an Id and a Name
- No conflict checking for Id/Name at design time



# Demo: Extensible Enum

# Extensible Enums – Application uptake



Codeunit 1314 – "Purch. Doc. From Sales Doc."

```
case SalesLine.Type of
    SalesLine.Type::" ":
        PurchaseLine.Type := PurchaseLine.Type::" ";
    SalesLine.Type::Item:
        PurchaseLine.Type := PurchaseLine.Type::Item;
    else
        Error(TypeNotSupportedErr,Format(SalesLine.Type));
end;
```

# Extensible Enums - Conversions


- Enums can only be assigned to enums of the same type
- No implicit conversion to/from Integer
- Implicit conversion to/from Options is supported, but is considered legacy
- Conversion methods are coming in a future update

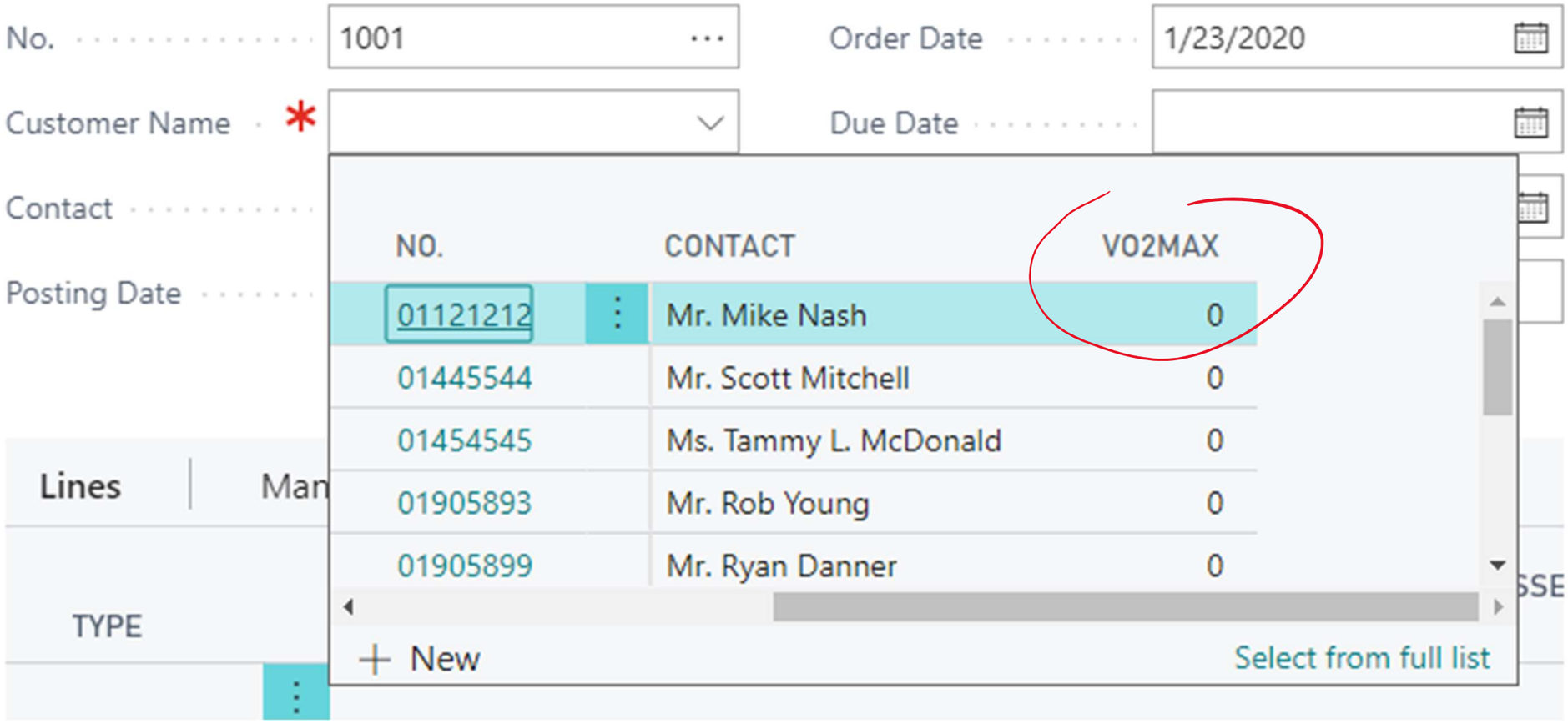
# Fieldgroup Extensibility


```
tableextension 50100 CustomerHealthStats extends Customer
{
    fields
    {
        field(50100; VO2Max; Integer) { }
    }
    fieldgroups
    {
        addlast(DropDown; VO2Max) { }
    }
}
```

NAV TECH DAYS 2018
mibuso.com

# Demo: Fieldgroups

# Fieldgroup Extensibility - UI



# Extensible Help

HelpLink on new Pages, Reports, and Xmlports

```
page 50100 MyPage
{
    HelpLink = 'https://www.mydocumentationwebsite.com/{0}/business-central/my-page';
}
```

app.json

```
"supportedLocales": [
    "da-DK", "en-US"
]
```

# Events

## Event Recorder

Record Events

| CALL ORDER | EVENT TYPE | HIT COUNT | OBJECT TYPE | OBJECT NAME | EVENT NAME | ELEMENT NAME | CALLING OBJECT TYPE | CALLING OBJECT NAME |
|---|---|---|---|---|---|---|---|---|
| 14 | Trigger Event | 1 | Page | Aged Acc. Recei... | OnOpenPageEvent | | | |
| 15 | Trigger Event | 1 | Page | CRM Statistics F... | OnOpenPageEvent | | | |
| 16 | Trigger Event | 1 | Page | Social Listening ... | OnOpenPageEvent | | | |
| 17 | Trigger Event | 1 | Page | Social Listening ... | OnOpenPageEvent | | | |
| 18 | Custom Eve... | 1 | Codeunit | ClientTypeMana... | OnAfterGetCurrentClientType | | Codeunit | ClientTypeMana... |
| 19 | Custom Eve... | 1 | Codeunit | Office Host Man... | OnIsAvailable | | Codeunit | Office Host Man... |
| 20 | Trigger Event | 1 | Page | Sales Hist. Sell-t... | OnOpenPageEvent | | | |
| 21 | Custom Eve... | 10 | Codeunit | UI Helper Triggers | GetCueStyle | | | |
| 22 | Trigger Event | 1 | Page | Customer Statist... | OnOpenPageEvent | | | |
| 23 | Trigger Event | 1 | Page | Dimensions Fact... | OnOpenPageEvent | | | |
| 24 | Trigger Event | 1 | Page | Workflow Status... | OnOpenPageEvent | | | |
| 25 | Trigger Event | 1 | Page | Customer Card | OnAfterGetRecordEvent | | | |

# Events

Improved event signature completion

```
[EventSubscriber(ObjectType::Codeunit, Codeunit::"Reporting Triggers", 'SubstituteReport', '', true, true)]
0 references
local procedure SubstituteCustomerReport(ReportId: Integer; var NewReportId: Integer;)
begin
    if ReportId = Report::"Customer Listing" the
        NewReportId := 50141;
    end;
end;
```

```
Event Parameter
RecordRef: RecordRef
RequestPageXml: Text
RunMode: Option
var
```

NAV TECH DAYS 2018
mibuso.com

# Do more with AL

NAV TECH DAYS 2018
mibuso.com

# Reporting

# Reports – Layout Generation

RDLC and Word layouts are now generated directly from VS Code

The dataset in the layout definition gets automatically updated when the report object is changed

# Report Substitution

Reports can be substituted when executed

- Through an action with RunObject
- One of the following static methods on the Report class: Run, RunModal, SaveAsHtml, SaveAsXml, SaveAsPdf, SaveAsExcel, SaveAsWord, RunRequestPage, Execute, Print, SaveAs

Reports CANNOT be substituted when executed from a variable

# Report Substitution - Sample


```
codeunit 50100 ReportSubstitution
{
  [EventSubscriber(ObjectType::Codeunit, 44, 'OnAfterSubstituteReport', '', true, true)]
  procedure OnSubstituteReport(ReportId: Integer; VAR NewReportId: Integer)
  begin
    if ReportId = Report::"Customer - Order Summary" then
      NewReportId := Report::”My Amazing Customer Report";
  end;
}
```

NAV TECH DAYS 2018
mibuso.com

# Demo: Report Substitution

# Reports “Reflection” - scenario

New virtual tables that provide reflection-like information:
- All Control Fields
- Report Data Items

These can be used to:
- Generate alternative Request Page UI in external service
- Generate report parameters XML and execute it via web-service call

# Reports “Reflection” - schema



**All Control Fields**

| Column Name | Data Type |
|---|---|
| Object Type | Option (Page, Report, XmlPort) |
| Object ID | Integer |
| Control ID | Integer |
| Control Name | Text |
| Data Type | Option (same as Field table) |
| Data Type Length | Integer |
| Option String | Text |
| Option Caption | Text |
| Caption | Text |
| Related Table ID | Integer |
| Related Field ID | Integer |
| Source Expression | Text |

**Report Data Items**

| Column Name | Data Type |
|---|---|
| Report ID | Integer |
| Data Item ID | Integer |
| Name | Text |
| Request Filter Fields | Text |
| Related Table ID | Integer |
| Indentation Level | Integer |
| Sorting Fields | Text |
| Data Item Table View | Text |

# Reports “Reflection” - sample usage in AL


```
codeunit 50103
{
    procedure '
    var
        "Report
        "Xml Pa
        "Result
    begin
        "Report
        "Report
        "Xml Pa
        "Xml Pa
        "Xml Pa
        "Xml Pa
        "Xml Pa
        Result
        Result.
    end;
    exit(Result);
end;
```

```
local procedure "Create Xml Options"("Report ID": Integer): XmlElement
var
    "All Cont
    "Xml Opti
    "Xml Fiel
begin
    "All Cont
    "All Cont
    "All Cont
    "Xml Opti
    repeat
        "Xml
        "Xml
        "Xml
    until "Al
    exit("Xml
end;
```

```
local procedure "Create Xml Data Items"("Report ID": Integer): XmlElement
var
    "Report Data Items": Record "Report Data Items";
    "Xml Data Items": XmlElement;
    "Xml Data Item": XmlElement;
begin
    "Report Data Items".SetRange("Report ID", "Report ID");
    "Report Data Items".Find('-');
    "Xml Data Items" := XmlElement.Create('DataItems');

    repeat
        "Xml Data Item" := XmlElement.Create('DataItem');
        "Xml Data Item".SetAttribute('name', "Report Data Items".Name);
        "Xml Data Item".Add(XmlText.Create("Report Data Items"."Data Item Table View"));

        "Xml Data Items".Add("Xml Data Item");
    until "Report Data Items".Next() = 0;

    exit("Xml Data Items");
end;
```

NAV TECH DAYS 2018
mibuso.com

# .NET Interoperability – On-Premise Only!

# .NET interoperability – On-Premise Only!


app.json

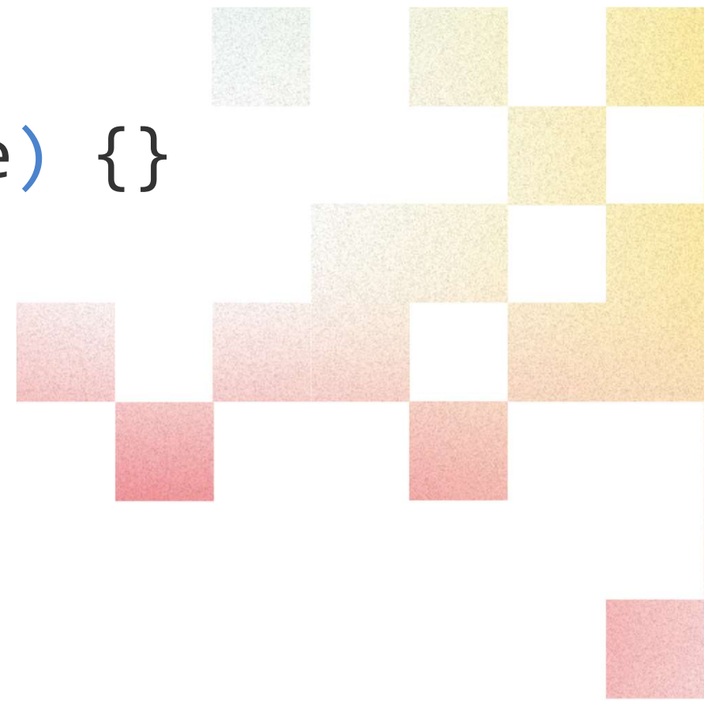
```
"target": "Internal"
```

settings.json

```
"al.assemblyProbingPaths": [
  "./.netpackages",
  "c:/Windows/assembly",
  "C:/Program Files/Microsoft Dynamics 365 Business Central/130/Service/Add-ins"
]
```

# .NET interoperability - Declaration


```
dotnet
{
  assembly(mscorlib)
  {
    type(System.TimeSpan) {}
    type(System.DateTime; MyDateTime) {}
  }
}
```



# .NET interoperability - Usage



```
codeunit 50101 DotNetUsage
{
  procedure ParseDateTime(s: Text): DateTime
  var
    dt: dotnet MyDateTime;
  begin
    exit(dt.Parse(s));
  end;
}
```

# Demo

# .NET interoperability – Usage Events


```
dotnet
{
  assembly("System")
  {
    Version = '4.0.0.0’;

    type(System.Timers.Timer; MyTimer) { }
    type(System.Timers.ElapsedEventArgs) { }
  }
}
```

```
pageextension 50101 DotNetUsage extends "Customer Card"
{
  var
    [WithEvents]
    timer: dotnet MyTimer;

  trigger OnOpenPage()
  begin
    SetupTimer();
  end;

  procedure SetupTimer()
  begin
    timer := timer.Timer(2000);
    timer.AutoReset := true;
    timer.Enabled := true;
    timer.Start();
  end;

  trigger timer::Elapsed(sender: Variant; e: DotNet ElapsedEventArgs)
  begin
    Message('The Elapsed event was raised at %1', e.SignalTime);
  end;
}
```

# .NET interoperability – Control Add-Ins


```
page 50100 PageUsingLegacyControlAddIns
{
    layout
    {
        area(Content)
        {
            usercontrol(MyPingPong; "Microsoft.Dynamics.Nav.Client.PingPong")
            {
                trigger AddInReady()
                begin
                end;

                trigger Pong()
                begin
                end;
            }
        }
    }
}
```

# .NET interoperability – Control Add-Ins


```
dotnet
{
      assembly("Microsoft.Dynamics.Nav.Client.PingPong")
      {
            type(Microsoft.Dynamics.Nav.Client.PingPong.PingPongAddIn; "Microsoft.Dynamics.Nav.Client.PingPong")
            {
                  IsControlAddIn = true;
            }
      }
}
```

# Isolated Storage

Allows storing and retrieving data in complete isolation from other applications.

The information can be stored per:

- Module
- CompanyAndUser
- Company
- User

# OData Bound Actions in AL

```
[ServiceEnabled]
0 references
procedure CreateCustomerCopy(var actionContext: WebServiceActionContext)
var
    createdCustomerGuid: Guid;
    customer: Record Customer;
begin
    actionContext.SetObjectType(ObjectType::Page);
    actionContext.SetObjectId(Page::"Customer Card");
    actionContext.AddEntityKey(customer.fieldNo(Id), createdCustomerGuid);
    actionContext.SetResultCode(WebServiceActionResultCode::Created);
end;
```

http://<Server>:<WebServicePort>/<ServerInstance>/ODataV4/Company(<CompanyId>)/<ServiceName>(<Record Id>)/Microsoft.NAV.**CreateCustomerCopy**

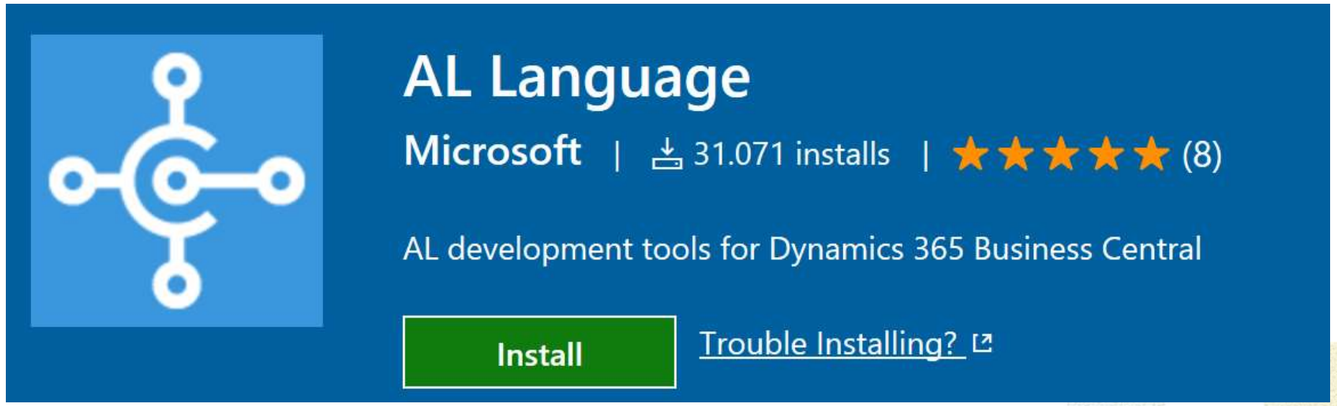
NAV TECH DAYS 2018
mibuso.com

# Do more with tools

# Runtime Versioning

app.json

```
”runtime": ”2.0”
```




Runtime versions

| Version | Release |
|---|---|
| 1.0 | Spring 2018 |
| 2.0 | Fall 2018 |
| 3.0 | Spring 2019 |

# Runtime Versioning – What is versioned



- **Syntax**
  e.g. enum, dotnet
- **Properties**
  e.g. ReplicateData
- **Library Functions**
  e.g. IsPathTemporary
- **Attributes**
  e.g. TryFunction
- **App.json**
  e.g. HelpBaseUrl
- **Data Types**
  e.g. DataScope
- **Server Communication**

# Demo – Runtime Version

# Debugger Enhancements

launch.json

```
"configurations": [{
  ...
  "breakOnError": true,
  "breakOnRecordWrite": false
  }]
```

Set breakpoints in external code (C/AL and AL)

# Browse base app code from VS Code

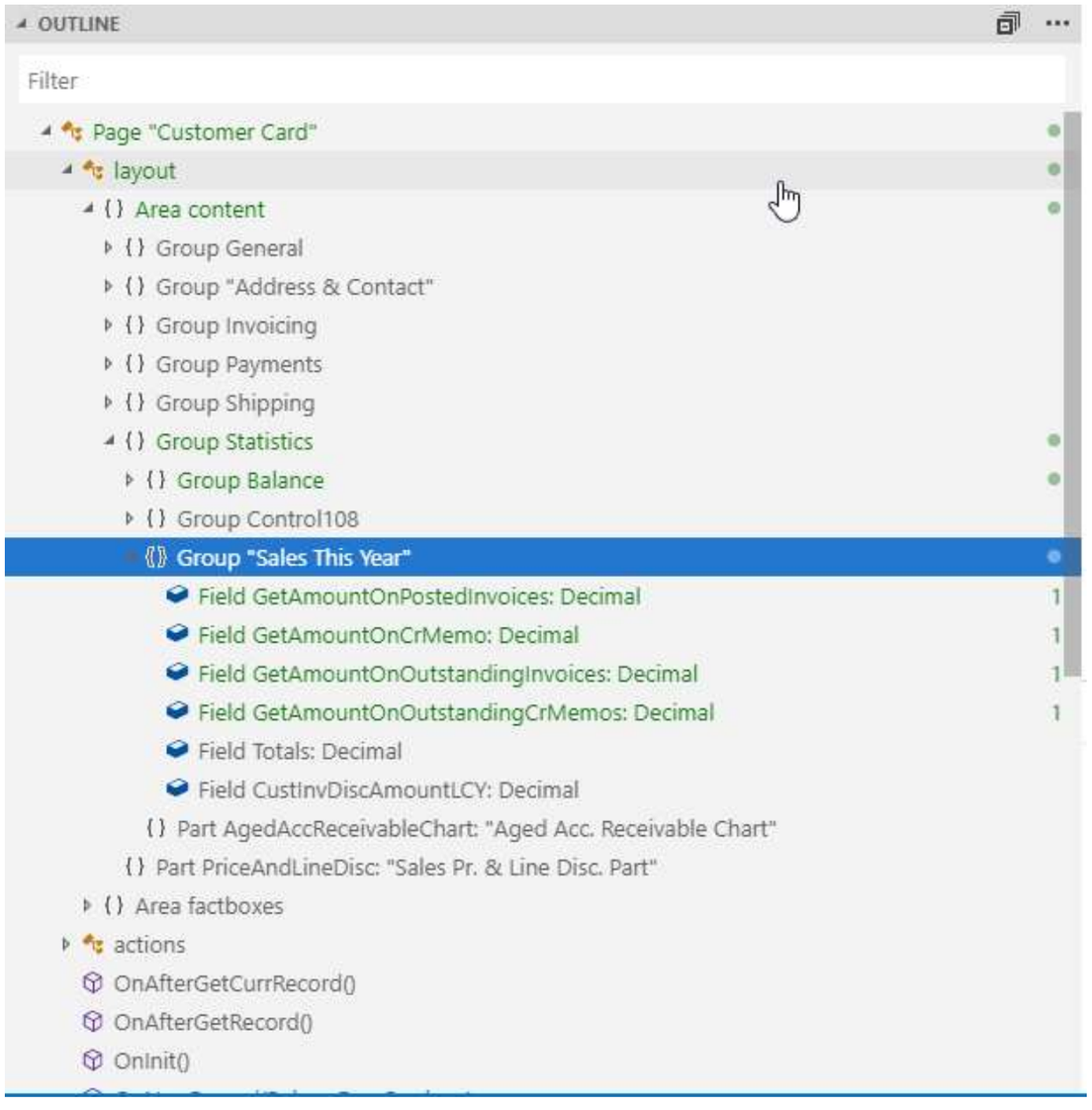
NAV TECH DAYS 2018
mibuso.com

Go-to Definition in external code (C/AL and AL)

Customer.dal

```
1   OnInsert()
2     IF "No." = '' THEN BEGIN
3       SalesSetup.GET;
4       SalesSetup.TESTFIELD("Customer Nos.");
5       NoSeriesMgt.InitSeries(SalesSetup."Customer Nos.",xRec."No. Series",0D,"No.","No. Series");
6     END;
7
8     IF "Invoice Disc. Code" = '' THEN
9       "Invoice Disc. Code" := "No.";
10
11    IF NOT (InsertFromContact OR (InsertFromTemplate AND (Contact <> '')) OR ISTEMPORARY) THEN
12      UpdateContFromCust.OnInsert(Rec);
13
```

# Outline View




- Hierarchical view of the elements in a file
- Advanced filtering and sorting
- Instant navigation to location

# Improved IntelliSense

Properties with help links

```
pageextension 50100 ItemCardExt extends "Item Card"
{
    DataCaptionExpression
}
```

(property) DataCaptionExpression: Expression

Sets an AL expression that is evaluated and displayed to the left of the page caption.

Get help

Image Properties

```
addlast(Creation)
{
    action(Hello)
    {
        image =
    }
}
```

- 1099Form
- 8ball
- AboutNav
- Absence
- AbsenceCalendar
- AbsenceCategories

AbsenceCategories Image

# Launch Browser


launch.json

```
…
{
   "configurations": [
   {
      …
      "launchBrowser": false
   }
]
}
```

# Multiple Id Ranges

```
"idRange":
{
   "from": 50000,
   "to": 60000
}
```

```
"idRanges":
[
   {
      "from": 50000,
      "to": 50100
   },
   {
      "from": 59000,
      "to": 60000
   }
]
```

# Generating permission sets from VS Code

NAV TECH DAYS 2018
mibuso.com

extensionsPermissionSet.xml ✖ NewTables.al

```xml
<?xml version="1.0" encoding="utf-8"?>
<PermissionSets>
  <PermissionSet RoleID="PERMISSIONS" RoleName="Permissions">
    <Permission>
      <ObjectType>0</ObjectType>
      <ObjectID>50100</ObjectID>
      <ReadPermission>1</ReadPermission>
      <InsertPermission>1</InsertPermission>
      <ModifyPermission>1</ModifyPermission>
      <DeletePermission>1</DeletePermission>
      <ExecutePermission>0</ExecutePermission>
      <SecurityFilter />
    </Permission>
```

# Analyzers – What are they



Analyzers are separate assemblies that provide additional code analysis
Analysis results are shown together with compiler output
Work for both Visual Studio Code and command-line compiler
Severity of rules can be controlled with RuleSets

# Analyzers – Shipped

| Analyzer Name | Description |
|---|---|
| CodeCop | CodeCop is an analyzer that enforces the official AL Coding Guidelines |
| UICop | UICop is an analyzer that ensures that your pages will work correctly in the Web Client and follow UI best practices |
| AppSourceCop | AppSourceCop is an analyzer that enforces rules that must be respected by extensions meant to be published to Microsoft AppSource |
| PerTenantExtensionCop | PerTenantExtensionCop is an analyzer that enforces rules that must be respected by extensions meant to be installed for individual tenants |

# Analyzers – How to enable

```
{
    "al.enableCodeAnalysis": true,
    "al.codeAnalyzers": [
        "${CodeCop}", "${AppSourceCop}"
    ]
}
```

# Analyzers – Rule sets

Analyzer rules can be individually enabled/disabled to suit your project needs

Use the **truleset** and **trule** snippets provided by the AL Language extension to create your ruleset.

MyRules.ruleset.json

```
{
    "name": "Strict rules",
    "description": "Make everything an error",
    "rules": [
        {
            "id": "CA001",
            "action": "Error",
            "justification": "It's the right thing to do!"
        }
    ]
}
```

# Analyzers – config file

To get the most out of AppSourceCop you need AppSourceCop.json and provide your prefix/suffix. It will also check if you made any schema breaking changes.

AppSourceCop.json:

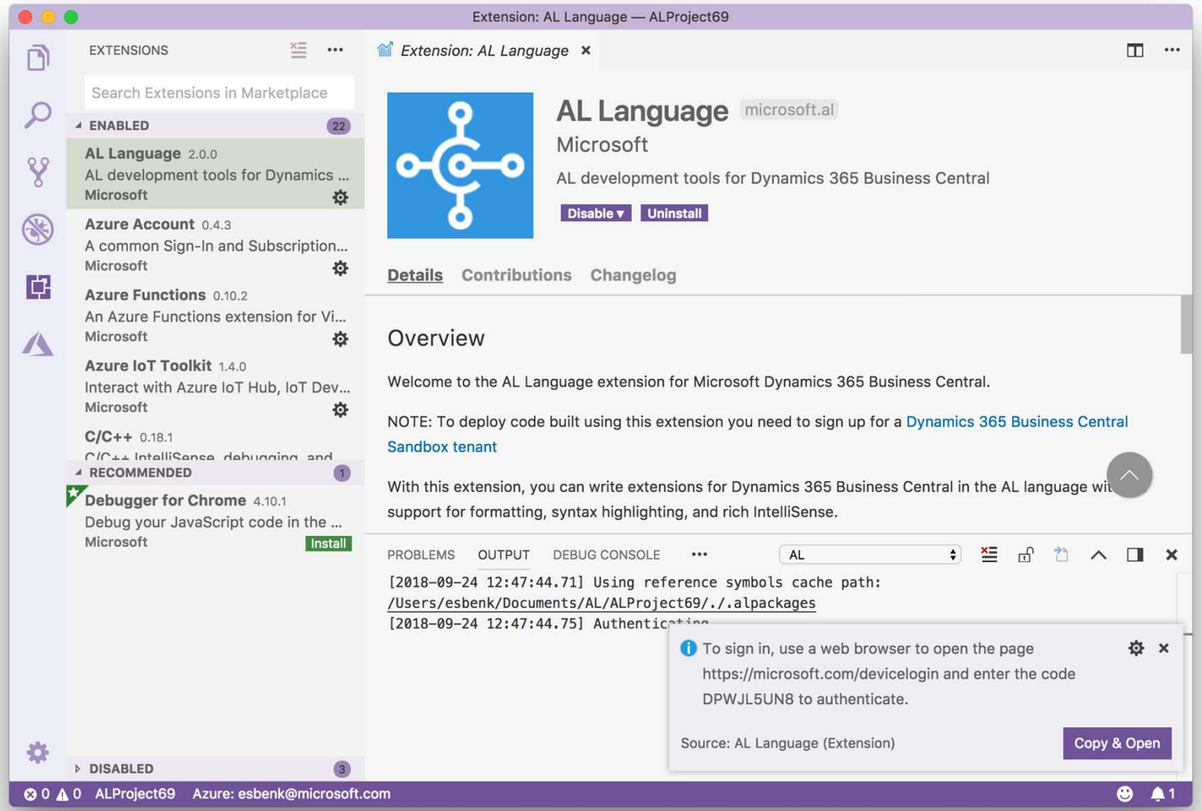
```
{
    // The following 3 properties are used to check for schema breaking changes
    // compared to the previous version of your app
    "name" : "",
    "publisher" : "",
    "version": "",

    "mandatoryPrefix" : "", // prefix to use for fields and table names
    "mandatorySuffix": "" //suffix to use for fields and table names
}
```

# MacOS

Same .vsix

Limited report functionality

- No RDL layout editing
- Limited Word layout editing
- Dataset, code, and request page can still be designed, modified, compiled, and deployed



NAV TECH DAYS 2018
mibuso.com

# Translations

# Translations - Multilanguage Syntax vs Translation File

App.json:

```
"features": [
  "TranslationFile"
]
```

You can choose one or the other but cannot mix the two modes.

For Per Tenant Extensions we will continue supporting the legacy translations approach for simplicity.

For App Source you need to use XLIFF based translations.

# Translations - Annotations

We made it easier to identify which translation unit corresponds to which AL element:

```
table 50100 ShippingRates
{
    fields
    {
        0 references
        field(1; Name; Text[40])
        {
            Caption = 'Rate Name';
            DataClassification = CustomerContent;
        }
    }
}
```

```
<trans-unit id="Table 56008245 - Field 2961552353 - Property 2879900210" size-unit="char" translate="yes" xml:space="prese
  <source>Rate Name</source>
  <note from="Developer" annotates="general" priority="2"></note>
  <note from="Xliff Generator" annotates="general" priority="3">Table ShippingRates - Field Name - Property Caption</note>
</trans-unit>
```

NAV TECH DAYS 2018
mibuso.com

# Any Questions?

NAV TECH DAYS 2018
mibuso.com

Thank you

# THANK YOU