

**NAV
TECH
DAYS
2016**

mibuso.com

MODERN DEVELOPER TOOLS FOR DYNAMICS 365 AND ON PREMISE NAV

MICHAEL HAMMOND – PRINCIPAL ENGINEERING LEAD
ESBEN NYHUUS KRISTOFFERSEN – PRINCIPAL ARCHITECT

WHEN YOU ARE PASSIONATE ABOUT MICROSOFT DYNAMICS NAV | www.navtechdays.com

**NAV
TECH
DAYS
2016**

mibuso.com

Session Objectives

- Understand extensions in the development context
- Enhancements to extensions V1
- See the new editors
- Understand the reference model
- Get excited

Customizations vs Extensions

Dynamics NAV

Allows customization of application logic to deliver customer, industry, or region specific functionality

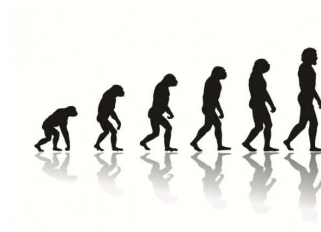
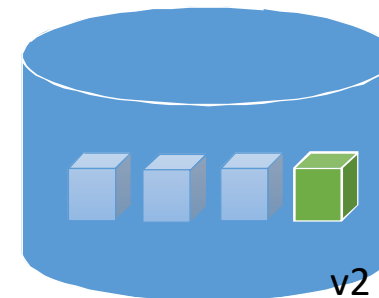
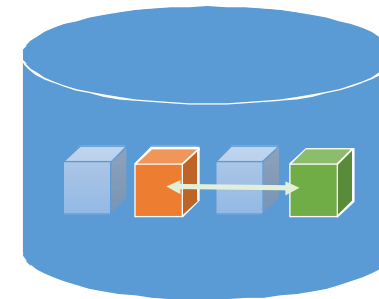
How?

Customizations: modification of base tables, code, and other objects

Extensions: addition of well-defined modules that are invoked alongside the base application

Why?

Upgrade ... and ...



Extensions – Roadmap

NAV 2016

Objects

- Pages (new and modified)
- Tables (new and modified)
- MenuSuites
- Codeunits

Data

- Permission sets (per tenant)

Developing

- Upgrading

NAV 2017

Objects

- Reports
- XML ports
- Queries

NET Add-ins

- Server-side .NET interop add-ins
- Client-side JavaScript extensibility control add-ins
- Client-side WinForms extensibility control add-ins

Data

- Web services (per tenant)
- Table data (new tables added for extension)
- Custom report layouts

Translations

- Language files

Developing

- Improved upgrade APIs
- Debugging and code coverage

Future vision

Platform Improvements

- Improvements to page and table changes
- Support for per-tenant profiles
- Ability to remove setup records on uninstall

Developing

- Improved developer tooling
- Localization and language enhancements
- In-client designer

Verticals

- Improved support for large applications

Customization

- Improved support for per-tenant customization

What's new for Extensions in NAV 2017 – (1/3)

Starting Data

- ❖ Export data from non-company specific table added as part of the extension using new cmdlet

Export-NAVAppTableData

- ❖ New database upgrade function

NAVAPP.LOADPACKAGEDATA(ID)

Custom Report Layouts

- ❖ Include in extension Custom Report layout registered in tenant database

- ❖ Export custom report layout using new cmdlet

Export-NAVAppReportLayout

Language files

- ❖ Include in extension multi-language captions and constants

- ❖ Create Language delta files using new cmdlet

Compare-NAVAppApplicationObjectLanguage

What's new for Extensions in NAV 2017 – (2/3)

Tenant Web Services

- ❖ Publish web services for single tenant or all tenants
- ❖ Published web services for all tenants it can't be unpublished in a single tenants

.NET Add-ins support

- ❖ Server-side .NET interop add-in referenced by CodeUnits
- ❖ Client-side JavaScript extensibility control add-in
- ❖ Client-side WinForms extensibility control add-in

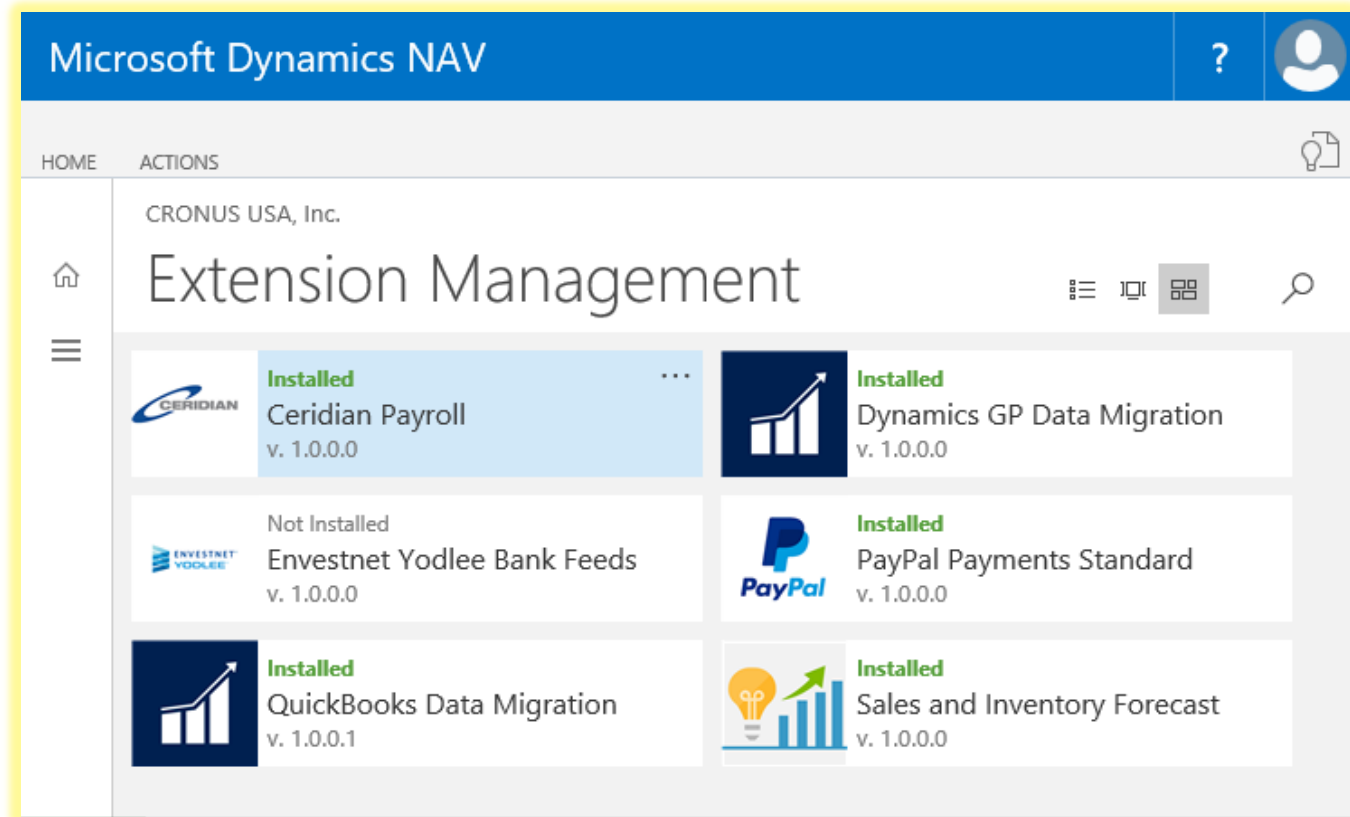
Debugging

- ❖ Debug installed extensions using Windows client
- ❖ Select extension objects the same as base objects
- ❖ Set breakpoints based on line number

What's new for Extensions in NAV 2017 – (3/3)

In Client Extensions Management

New Page to Install and Uninstall extensions



Building solutions for Dynamics 365 for Financials

- Extensions are the way to build for Software as a Service (SaaS)
- Allow you to sell your product in Microsoft AppSource - free advertising!
Increase your sales volume!
- Takes advantage of our upgrade investments
- Offer additional protection of your IP compared to the very “open source” nature of classic NAV
- Platform stability
- Reduced time to deployment

Our audience



Visual Studio Code

DEMO

What did we just see?

- Dynamics 365 for Financials*
- Visual Studio Code
- NAV Extension for Visual Studio Code
 - Context aware intellisense
 - Background compilation
 - Syntax highlighting
 - Auto completion
 - Snippets
 - Symbolic navigating
 - Unicode in source
- Extension object
- Unicode in source
- Build, package, and run integration

**Also works on premise*

Visual Studio Code – more details

Free

Not Visual Studio

Now part of Visual Studio

Code optimized

Debugger

Native Git support

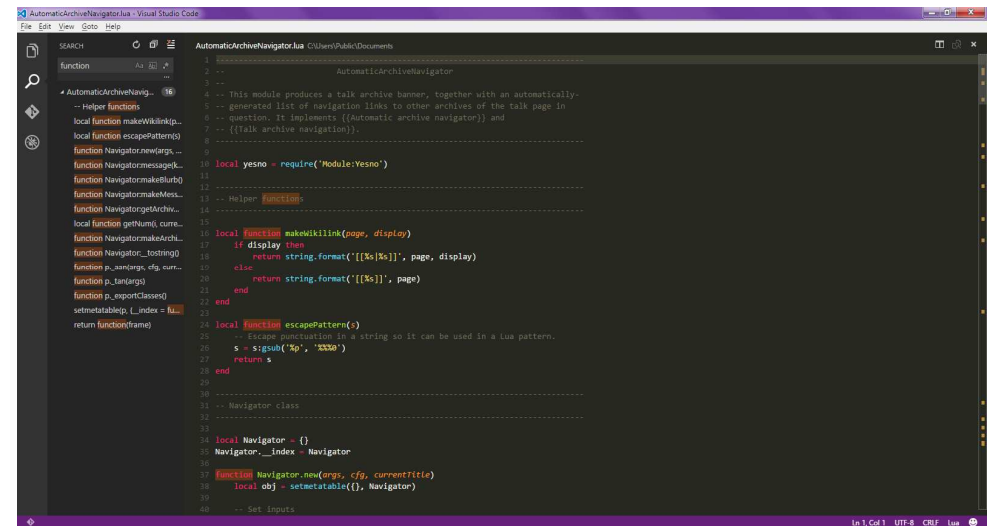
Always updated – monthly/nightly build

Mac OS and Linux platforms

Extensions

Community

Open source (MIT license)



The screenshot shows the Visual Studio Code interface with a file named 'AutomaticArchiveNavigator.lua' open. The editor has a dark theme. On the left, there is a sidebar with a 'SEARCH' panel and a file explorer. The main editor area displays Lua code. The code includes comments in Italian, a 'local yesno' function, a 'local function makeWikiLink' function, and a 'local function escapePattern' function. The code is well-formatted with indentation and line numbers. The status bar at the bottom indicates 'Ln 1, Col 1 UTF-8 CRLF lua'.

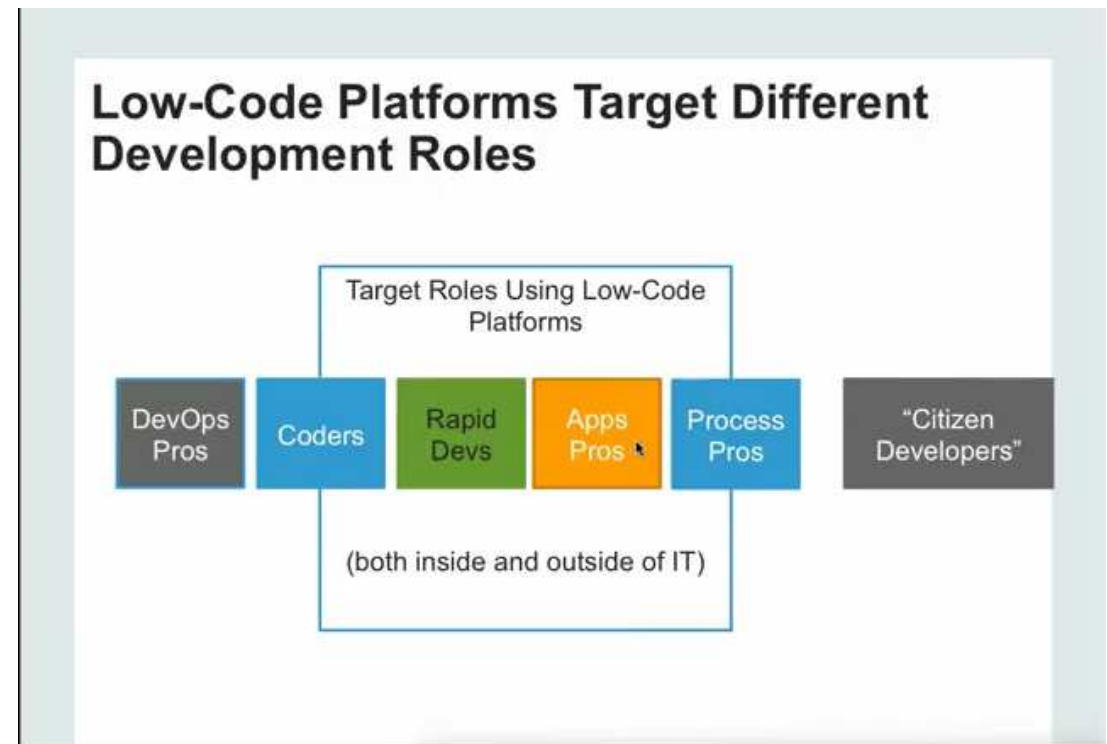
```
1 AutomaticArchiveNavigator.lua C:\Users\Public\Documents
2 -- AutomaticArchiveNavigator
3 --
4 -- This module produces a talk archive banner, together with an automatically-
5 -- generated list of navigation links to other archives of the talk page in
6 -- question. It implements ((Automatic archive navigator)) and
7 -- ((Talk archive navigation)).
8
9
10 local yesno = require('Module:Yesno')
11
12
13 -- Helper functions
14
15 local function getNum(s, curr)
16     local function makeWikiLink(page, display)
17         if display then
18             return string.format('{{%s|%s}}', page, display)
19         else
20             return string.format('{{%s}}', page)
21         end
22     end
23
24 local function escapePattern(s)
25     -- Escape punctuation in a string so it can be used in a Lua pattern.
26     s = string.gsub(s, '%p', '\\%0')
27     return s
28 end
29
30
31 -- Navigator class
32
33
34 local Navigator = {}
35 Navigator.__index = Navigator
36
37 function Navigator.new(args, cfg, currentTitle)
38     local obj = setmetatable({}, Navigator)
39
40     -- Set inputs
```

Two sides of the same coin

The right UX for the right user
Underneath everything is an extension

Visual for Business Consultant
Code for the developer

But why C/AL still?



Forrester Report 2016

In-client designer

DEMO

Extension Objects

- No more export-code-export-diff development process
- Coded with intent
- Compiler understands what is allowed in page and table extensions

- First-class objects of the platform
- No more duplicate objects in platform
- Will allow for scenarios that currently require code on pages
- Will allow access to the extended page through defined APIs

Visual Studio Code – take 2

DEMO

Companion Tables

- Extended table data will be stored in separate tables rather than modifying the existing schema
- Will help with naming conflicts between extensions
- Will allow for improvements to upgrade process
- Will make possible sharing schema between tenants

Visual Studio Code – take 3

DEMO

Syntax Changes

```

table 1337 MyTable
{
  fields
  {
    field(1; FirstName; Text[50]) { }
    field(2; LastName; Text[50]) { }
    field(13; ValidRecord; Option)
    {
      OptionString = Valid,NotValid;
      Description = 'Indicate if the record is
valid';
    }
  }
  keys
  {
    key(1;PK;Firstname)
    {
      Clustered = true;
    }
  }

  trigger OnInsert();
  begin
    //Do something very important here.
  end;
}

```

```

OBJECT Table 1337 MyTable
{
  OBJECT-PROPERTIES
  {
    Date=13-10-16;
    Time=[ 2:57:31 PM];
    Modified=Yes;
    Version List=;
  }
  PROPERTIES
  {
    TableType=Normal;
  }
  FIELDS
  {
    { 1 ; ;FirstName ;Text50 }
    { 2 ; ;LastName ;Text50 }
    { 3 ; ;validrecord ;Option ;OptionString=valid,invalid;
      Description=Indicate if the record is valid. }
  }
  KEYS
  {
    { ;FirstName ;Clustered=Yes }
  }
  FIELDGROUPS
  {
  }
  CODE
  {
    BEGIN
    END.
  }
}

```

Page and Extension syntax

```
page 86 MyPage
{
    PageType = List;
    SourceTable = MyTable;

    layout
    {
        area(content)
        {
            group(General)
            {
                field("Firstname";"Firstname") {}
                field("Lastname";"Lastname") {}
            }
        }
    }
}
```

```
pageextension 50100 MyPageExtension extends MyPage
{
    layout
    {
        addlast(address)
        {
            field(AddressX;MyNewField) {}
        }
    }
}
```

.NET Interop – what's the deal?


The introduction of the .NET Interop type was to allow easy access to .NET Framework and custom methods

Options

.NET through C/AL language methods

.NET through application objects

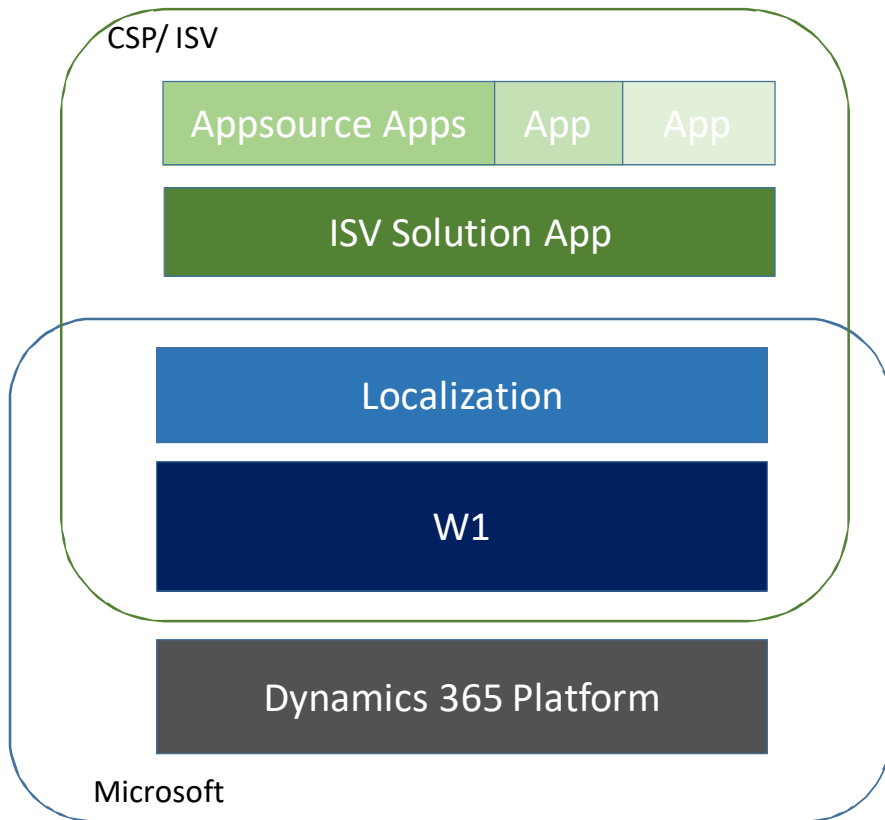
Azure functions



What did you
use .NET
interop for?

Future Extension Enhancements

Solution App model



- Solution apps will have more ability to modify the functionality of the base application
- Solution apps will be able to modify some elements of branding
- W1 and localizations may also become solution apps

Customer Apps

- Special type of solution apps that that apply to a single tenant
- Applied “on top” of all other extensions

Actions

When you leave...

- Start to refactor code
 - Remove code from modified objects where possible
 - Good: Use “Hook” pattern
 - Better: Use events
- Evaluate architecture of solution
 - How much code is reused between customers?
 - Can your solution be broken into multiple extensions?
- Start to build code into extensions where possible

One more thing...

