

**NAV
TECH
DAYS
2013**

mibuso.com

DYNAMICS NAV APPLICATION DESIGN

Bogdana Botez, Nikola Kukrika, Anders Larsen, Bogdan Sturzoiu
(Microsoft MDCC)

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

**NAV
TECH
DAYS
2013**

mibuso.com

DYNAMICS NAV APPLICATION DESIGN

Creating reusable solutions for Dynamics NAV Implementations

Session overview

- Introduction to NAV Design Patterns
- Design Patterns used in NAV - examples
- Mini App (C5) Introduction
- Design Patterns used in Mini App

INTRODUCTION TO NAV DESIGN PATTERNS

Purpose:

- Increase code quality
- Increase developer productivity

How

- Use a common language
- Explain the reasoning behind a certain design
- Share knowledge
- Create a community shared project

INTRODUCTION TO NAV DESIGN PATTERNS

Team:

- Abshishek Ghosh, Microsoft
- Anders Larsen, Microsoft
- Bardur Knudsen, Microsoft
- Bogdan Sturzoiu, Microsoft
- Bogdana Botez, Microsoft
- Nikola Kukrika, Microsoft
- Eric Wauters, iFacto, PRS
- Gary Winter, agiles, PRS
- Mark Brummel, Brummel Dynamics Services, PRS
- Vjeko Babic, Fortempo, PRS
- Luc van Vugt, Fluxxus
- Claus Lundstrøm, Abakion
- Natalie Karolak, Tectura
- David Studebaker, Studebaker Technology
- Mark Doster, Mergetool
- Søren Klemmensen, IndustryBuilt
- Arend-Jan Kauffmann, Xperit Products

COMMUNITY

Patterns have been published on NAV Team Blog, until now.

However

- We want this to be a community project

Community NAV Patterns Wiki on

<https://community.dynamics.com/nav/w/designpatterns/default.aspx>

Join us.

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



DESIGN PATTERNS USED IN NAV

The following patterns are lined up for the presentation:

- Journal Error Processing
- Temporary Dataset Reports
- Using Query Objects to Detect Duplicates
- Using Query Objects Instead of Nested Loops
- Mapping W1 Features to Local Data Structures

Find the complete content on the [NAV Patterns Wiki](#) and on [NAV TEAM Blog](#) (Published after TechDays.)

JOURNAL ERROR PROCESSING

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



JOURNAL ERROR PROCESSING

Context

Data entered by users can be affected by human error.

How to best help the user when data is: invalid, incomplete, or inconsistent?

Problem

- The “traditional” way of implementing error processing, serial data processing validation:
 - After entering data, the user invokes an action (e.g. SEPA export file)
 - The processing stops at the first error
 - The user fixes the error
 - Repeat
- Repetitive: The action is invoked multiple times. The user cannot predict the total volume of corrections needed.

Solution

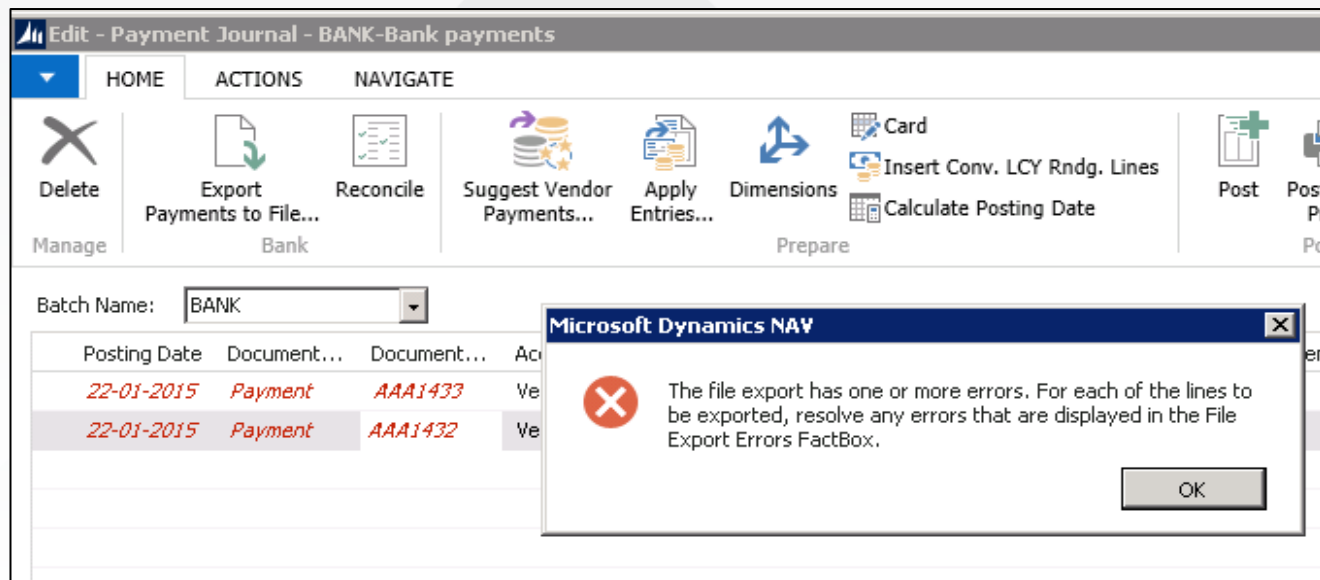
Show all errors once, at the end:

- Find and store all data validation errors
- Prompt for correction of all, at the end
- Show an overview of the affected lines (e.g. by sorting them first and highlight in red)

JOURNAL ERROR PROCESSING

The Final Result

Example from the implementation of SEPA Credit Transfer



JOURNAL ERROR PROCESSING

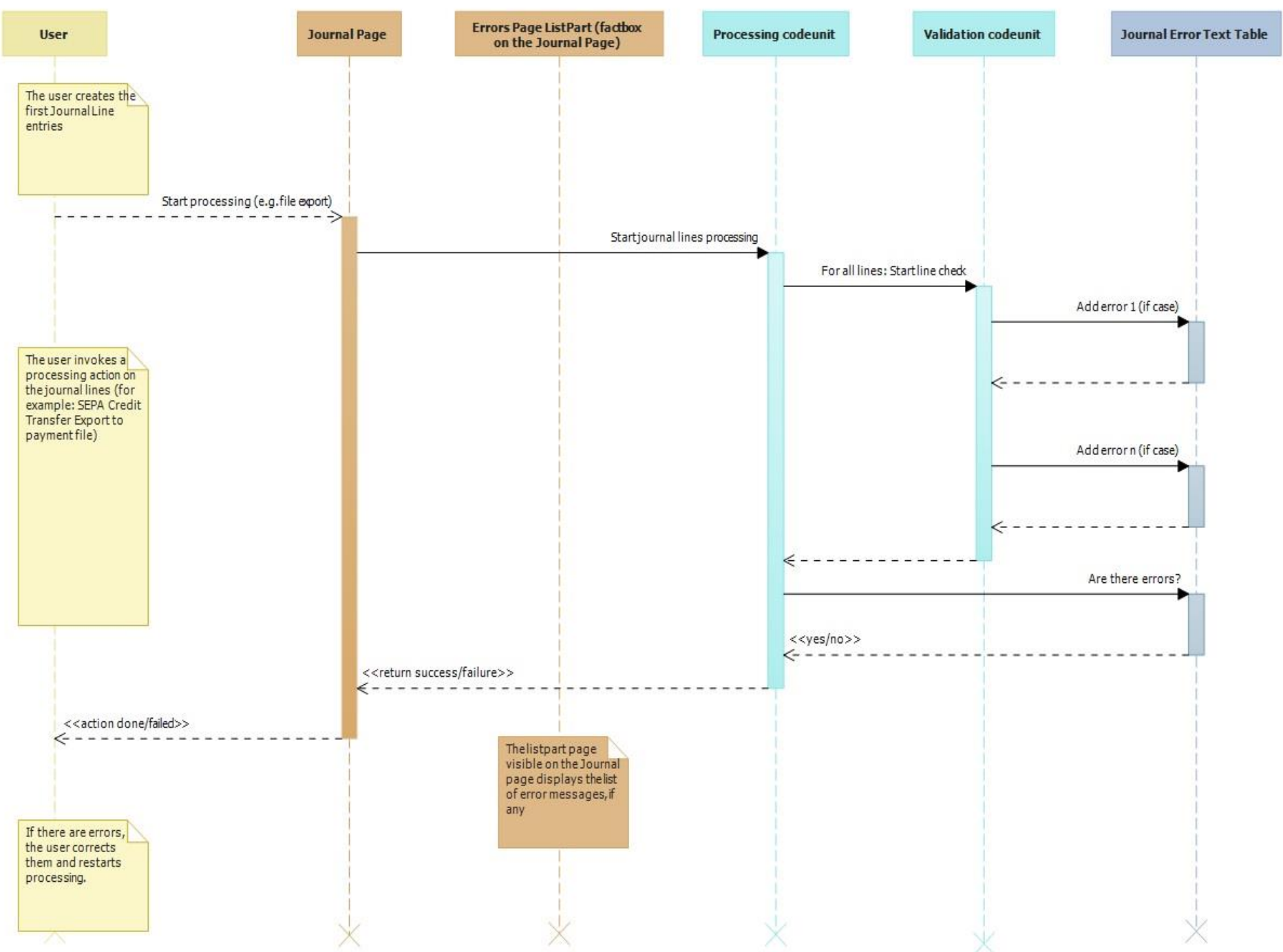
Example from the implementation of SEPA Credit Transfer

The screenshot shows the 'Edit - Payment Journal - BANK-Bank payments' window. The 'Batch Name' is 'BANK'. The table below shows a payment entry for 22-01-2015 with document AAA1432, account 10000, and recipient ECA. The description is 'London Postmaster'. The 'Payment File Errors' pane on the right lists two errors: 'Vendor Bank Account PAR must have a value in SWIFT Code.' and 'Vendor Bank Account PAR must have a value in IBAN.'.

Posting Date	Document...	Document...	Account T...	Account No.	Recipient ...	Description
22-01-2015	Payment	AAA1433	Vendor	20000	PAR	AR Day Property Ma
22-01-2015	Payment	AAA1432	Vendor	10000	ECA	London Postmaster

The screenshot shows the 'Edit - Payment Journal - BANK-Bank payments' window. The 'Batch Name' is 'BANK'. The table below shows a payment entry for 22-01-2015 with document AAA1432, account 10000, and recipient ECA. The description is 'London Postmaster'. The 'Payment File Errors' pane on the right lists two errors: 'Only transactions in euro (EUR) are allowed.' and 'Vendor Bank Account ECA must have a value in SWIFT Code.'.

Posting Date	Document...	Document...	Account T...	Account No.	Recipient ...	Description
22-01-2015	Payment	AAA1433	Vendor	20000	PAR	AR Day Property Ma
22-01-2015	Payment	AAA1432	Vendor	10000	ECA	London Postmaster



JOURNAL ERROR PROCESSING

```
Table 1228 Payment Jnl. Export Error Text - C/AL Editor

CreateNew(GenJnlLine : Record "Gen. Journal Line";NewText : Text)
SetLineFilters(GenJnlLine);
IF FINDLAST THEN;
"Journal Template Name" := GenJnlLine."Journal Template Name";
"Journal Batch Name" := GenJnlLine."Journal Batch Name";
"Document No." := GenJnlLine."Document No.";
"Journal Line No." := GenJnlLine."Line No.";
"Line No." += 1;
"Error Text" := COPYSTR(NewText,1,MAXSTRLEN("Error Text"));
INSERT;

JnlLineHasErrors(GenJnlLine : Record "Gen. Journal Line") : Boolean
SetLineFilters(GenJnlLine);
EXIT(NOT ISEMPY);

JnlBatchHasErrors(GenJnlLine : Record "Gen. Journal Line") : Boolean
SetBatchFilters(GenJnlLine);
EXIT(NOT ISEMPY);

DeleteJnlLineErrors(GenJnlLine : Record "Gen. Journal Line")
IF JnlLineHasErrors(GenJnlLine) THEN
DELETEALL;

DeleteJnlBatchErrors(GenJnlLine : Record "Gen. Journal Line")
IF JnlBatchHasErrors(GenJnlLine) THEN
DELETEALL;
```

JOURNAL ERROR PROCESSING

Examples in Dynamics NAV 2013

- SEPA Credit Transfer feature - for export of vendor payments
- SEPA Direct Debit feature - for export of customer payment instructions

The same concept of storing error messages (but with a different flow) is also present in:

- Planning Error Log table (5430) - Supply Planning feature
- Costing table (5890) - Costing feature

TEMPORARY DATASET REPORT

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



TEMPORARY DATASET REPORT

Problem

- The built-in NAV iterator is limited. It can only run through records written into the database.
- Some reports need complex data calculated at runtime, from more than one table

Solution

- Parse the data sources to create a record buffer in a temporary record variable.
- Iterate through a **DataItem** of the **Integer** table and display one record from the temporary recordset in each iteration.

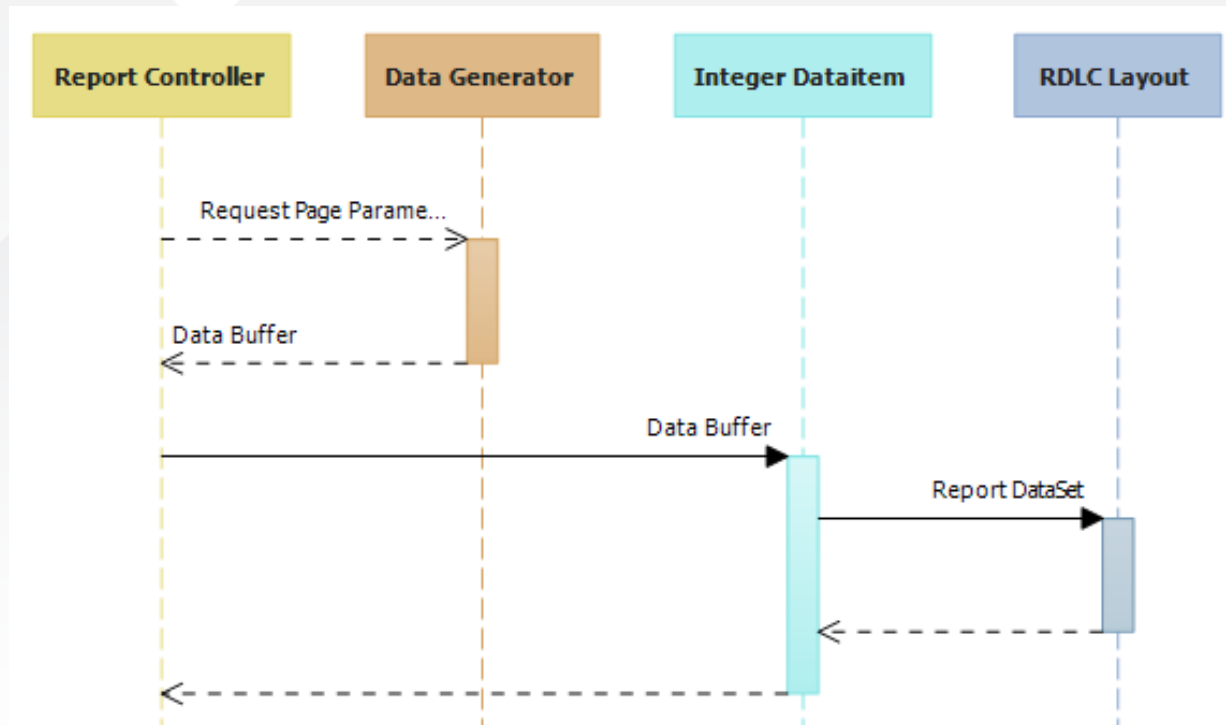
TEMPORARY DATASET REPORT

Step 1

Combining data sources to create a dataset

Step 2

Iterating through the Integer dataitem

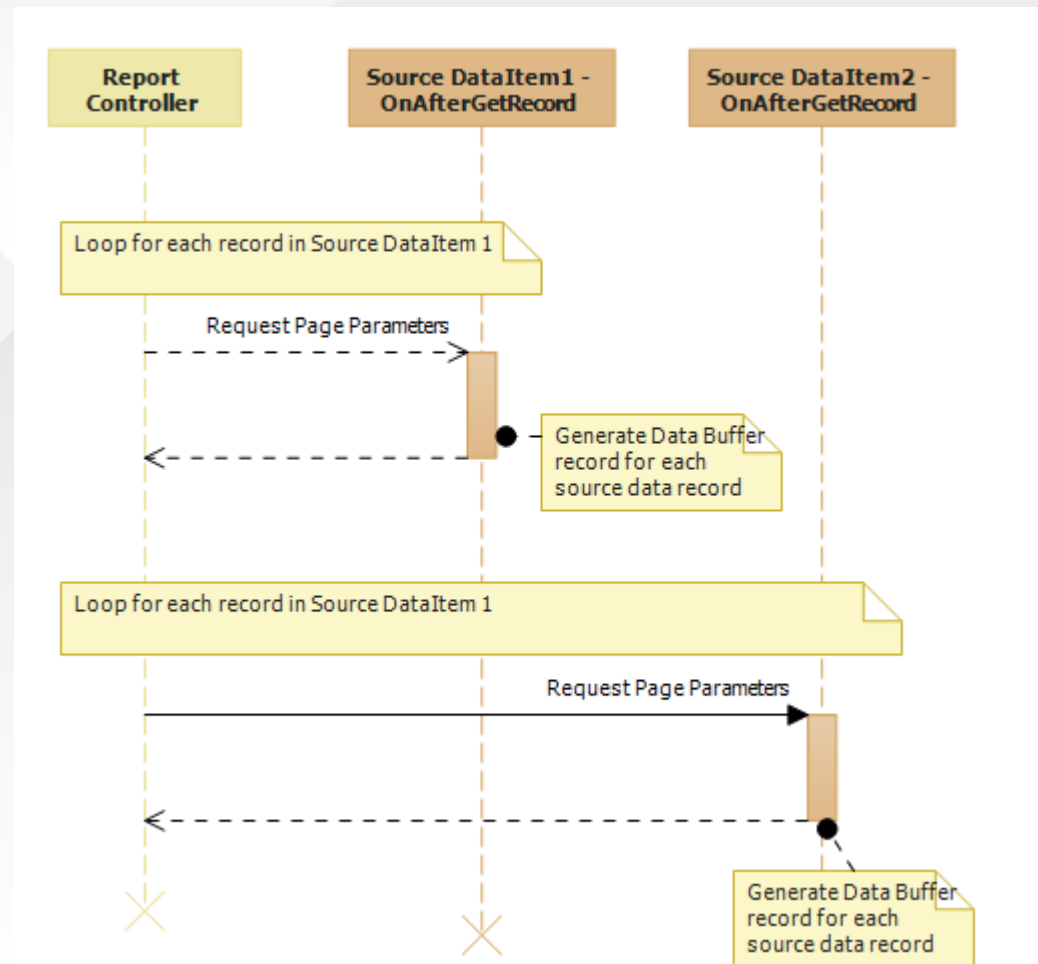


TEMPORARY DATASET REPORT

Step 1

Combining data sources to create a dataset:

- process the existing data
- create a temporary recordset



TEMPORARY DATASET REPORT

Step 2

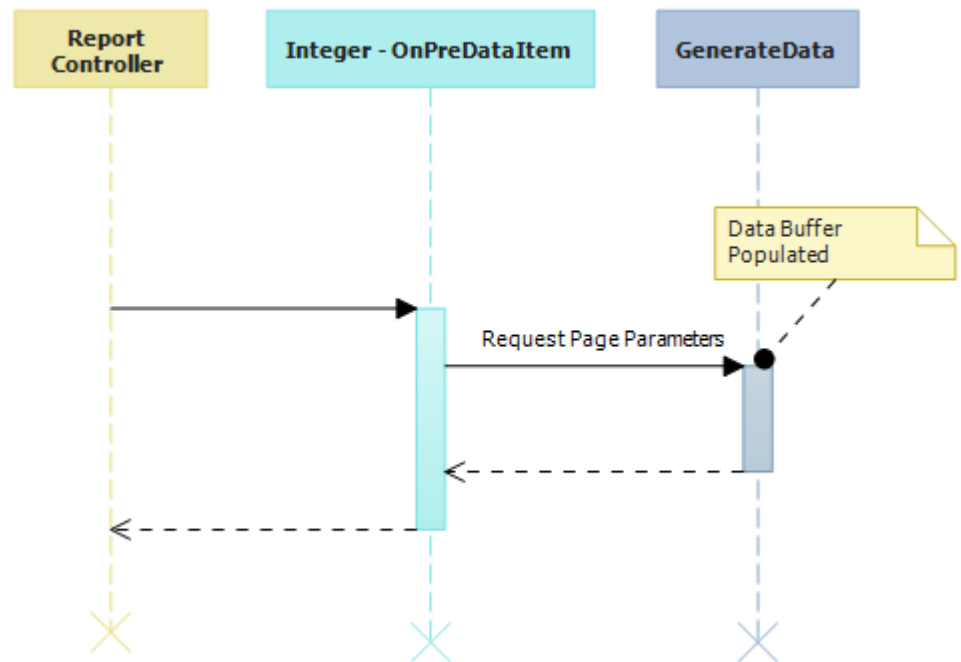
Iterate through the Integer dataitem

Problem: The report controller in NAV cannot iterate through temporary records or through the results of a query.

Solution: This is where the Integer table comes into use.

There are two ways to use the Integer dataitem to iterate through the temporary recordset:

- Loop for an indeterminate number of times until you reach the end of the recordset.
- Calculate the number of records in the temporary recordset in advance and then iterate that many number of times.



TEMPORARY DATASET REPORT

Step 2

Iterate through the Integer dataitem

Integer - OnPreDataItem:

```
ContactDocumentBuffer.SETCURRENTKEY("Document Date");  
SETRANGE(Number,1,ContactDocumentBuffer.COUNT);
```

Then we must move the record pointer by one record every time we loop through the Integer dataitem. So, in Integer – OnAfterGetRecord, we add the following lines of code:

```
IF Number = 1 THEN  
    ContactDocumentBuffer.FINDFIRST  
ELSE  
    ContactDocumentBuffer.NEXT;
```

TEMPORARY DATASET REPORT

Examples in Dynamics NAV 2013

- Report 204 – Sales - Quote
- Report 205 – Order Confirmation
- Report 206 – Sales - Invoice

USING C/AL QUERY OBJECTS TO DETECT DUPLICATES

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



USING C/AL QUERY OBJECTS TO DETECT DUPLICATES

Problem

Detection of duplicate data in a table is needed in many business scenarios, such as to:

- Find duplicate entries in a business entity table (e.g. Customer), as a first step for data clean-up.
- Find contacts that have the same name, to merge them.

Solution

Pre-Dynamics NAV 2013:

- Define the relevant field as a primary key (Not practical)
- Loop through the table and filter the same table in a sub-loop (Expensive due to extensive use of filters)

Starting with Dynamics NAV 2013, we can use the query object to solve this problem.

USING C/AL QUERY OBJECTS TO DETECT DUPLICATES

Pattern Elements

- A query object:
 - The dataitem is the table that we want to check for duplicates
 - The field we want to search for is a grouped field
 - A totaling field using the Count method
 - Filter on Count > 1
- The query is then invoked in code. If it returns any row, it means that there are duplicates on that field.

USING C/AL QUERY OBJECTS TO DETECT DUPLICATES

Implementation Example

- Before Dynamics NAV 2013, using nested loops and filtering:

```
HasDuplicateCustomers() : Boolean
IF Customer.FINDSET THEN
    REPEAT
        Customer2.SETRANGE(Name, Customer.Name);
        IF Customer2.COUNT > 1 THEN
            EXIT(TRUE);
        UNTIL Customer.NEXT = 0;
    EXIT(FALSE);
```


USING C/AL QUERY OBJECTS TO DETECT DUPLICATES

Implementation Example

- In Dynamics NAV 2013, using queries:

```
HasDuplicateCustomersWithQuery() : Boolean  
CustomerDuplicate.OPEN;  
EXIT(CustomerDuplicate.READ);
```

USING C/AL QUERY OBJECTS TO DETECT DUPLICATES

Examples in Dynamics NAV 2013

- Codeunit 762 Acc. Sched. Chart Management, methods CheckDuplicateAccScheduleLineDescription and CheckDuplicateColumnLayoutColumnHeader
- Codeunit 770 Analysis Report Chart Mgt., methods CheckDuplicateAnalysisLineDescription and CheckDuplicateAnalysisColumnHeader

Limitations

- Does not scale. A new query must be defined for every field (or group of fields) that needs to be checked.
- Queries cannot be created on the fly.
- Queries do not take parameters at runtime.

USING C/AL QUERY OBJECTS INSTEAD OF NESTED LOOPS

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



USING C/AL QUERY OBJECTS INSTEAD OF NESTED LOOPS

Problem

Table 274, Bank Acc. Reconciliation Line, and table 271, Bank Account Ledger Entry, are connected through the Bank Account No field. Identify the matching pairs of records based on having the same remaining amount and transaction date.

- Solving this kind of problem involves inspecting data from two different tables using a join operation.

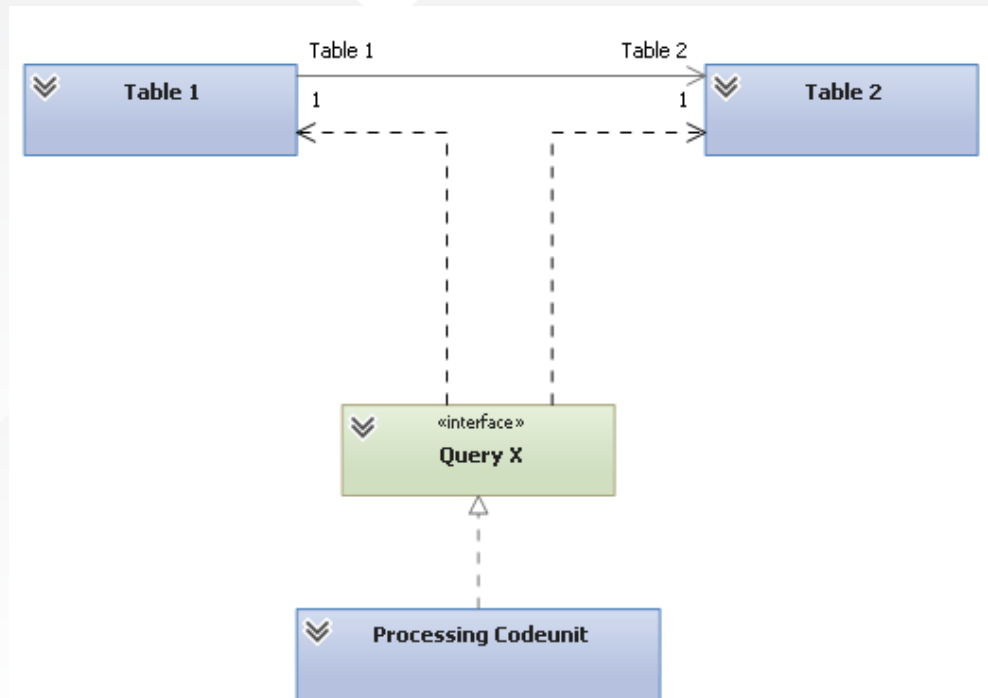
Solution

Pre-Dynamics NAV 2013:

- Loop through the first table, and then again through the second table.
 - Nested loops are costly, because they require intense database querying.
 - The code is hard to read and doesn't scale well with the number of connected tables.

Starting with Dynamics NAV 2013, we can use the query object to solve this problem.

USING C/AL QUERY OBJECTS INSTEAD OF NESTED LOOPS



USING C/AL QUERY OBJECTS INSTEAD OF NESTED LOOPS

Solution Using Nested Loops

The classic C/AL approach is to:

- Set the necessary filters on the left table, i.e. table 274.
- Loop through the filtered records.
- For each record in the filter, find the related records in the right table (table 271) and set the required filters on it.
- For each pair of records from the left and right table, decide if they are a solution and if so, apply them to each other.

USING C/AL QUERY OBJECTS INSTEAD OF NESTED LOOPS

Solution Using Query

The new query-based approach involves:

- Define a query that returns the full filtered join of tables 271 and 274.
- Loop through the records returned by the query.
- For each query record, decide if it represents a solution and then apply them.

CODE EXAMPLE: USING C/AL QUERY OBJECTS INSTEAD OF NESTED LOOPS

USING C/AL QUERY OBJECTS INSTEAD OF NESTED LOOPS

Advantages of the New Pattern

- A query joins the tables faster than loops.
- The query approach leverages the power of SQL Server.
- A query is scalable (in the sense that it allows reusing its definition for a larger set of problems).
- The code is cleaner and easier to read.
- Better and easier maintenance.

Limitations

- Requires defining new query objects.
- Queries have a static definition (cannot be created on the fly).

USING C/AL QUERY OBJECTS INSTEAD OF NESTED LOOPS

Example

In Microsoft Dynamics NAV 2013 R2, we can see the query object used in the bank account reconciliation matching algorithm.

The object is query 1252, Bank Rec. Match Candidates. It is called by the matching engine in codeunit 1252, Match Bank Rec. Lines.

MAPPING W1 FEATURES TO LOCAL DATA STRUCTURES

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



MAPPING W1 FEATURES TO LOCAL DATA STRUCTURES

Problem

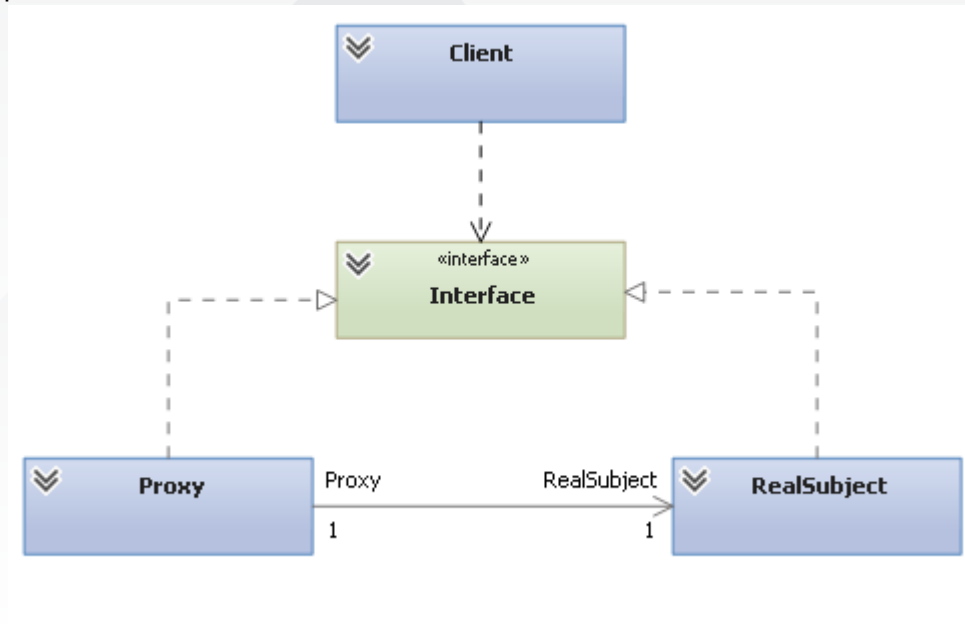
- A feature has been created for a country version at a certain time in history.
- Later, a similar feature is developed in W1.
- The two use different tables but have similar functionality.

How to make them coexist?

MAPPING W1 FEATURES TO LOCAL DATA STRUCTURES

Solution

- Use a proxy pattern



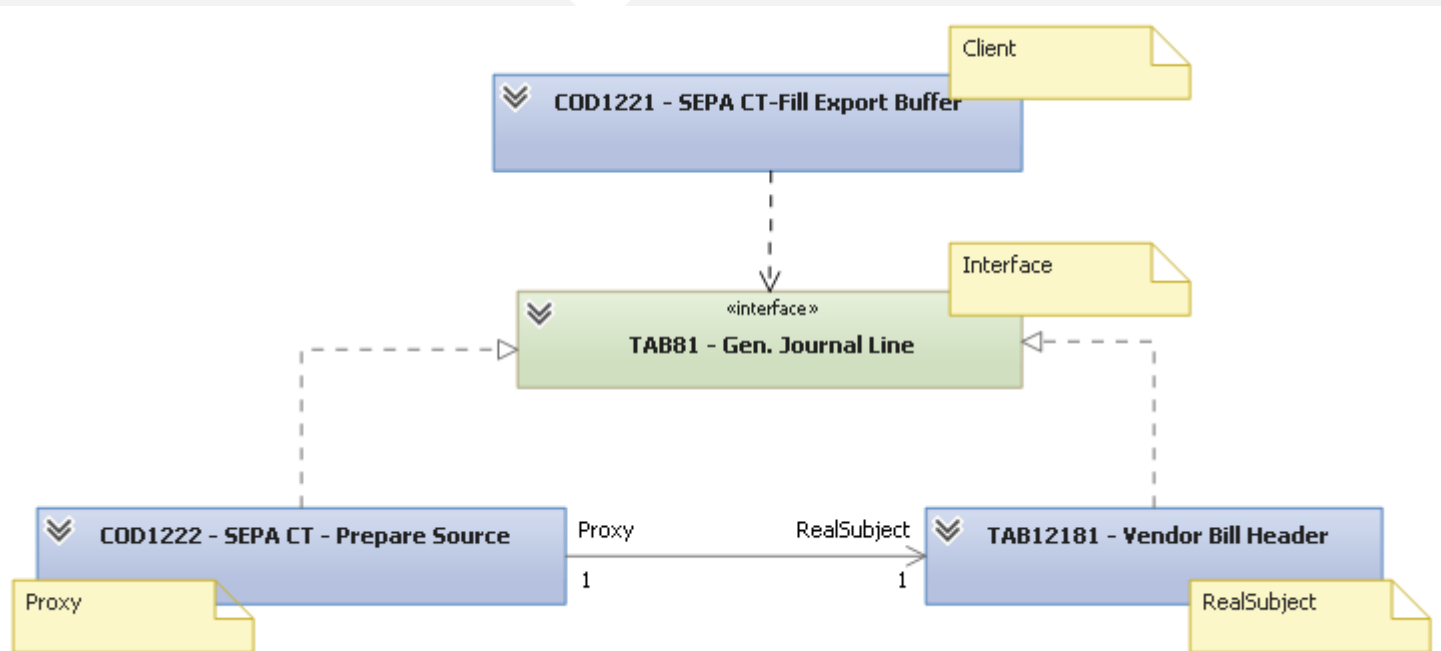
MAPPING W1 FEATURES TO LOCAL DATA STRUCTURES

Example

SEPA Credit Transfers

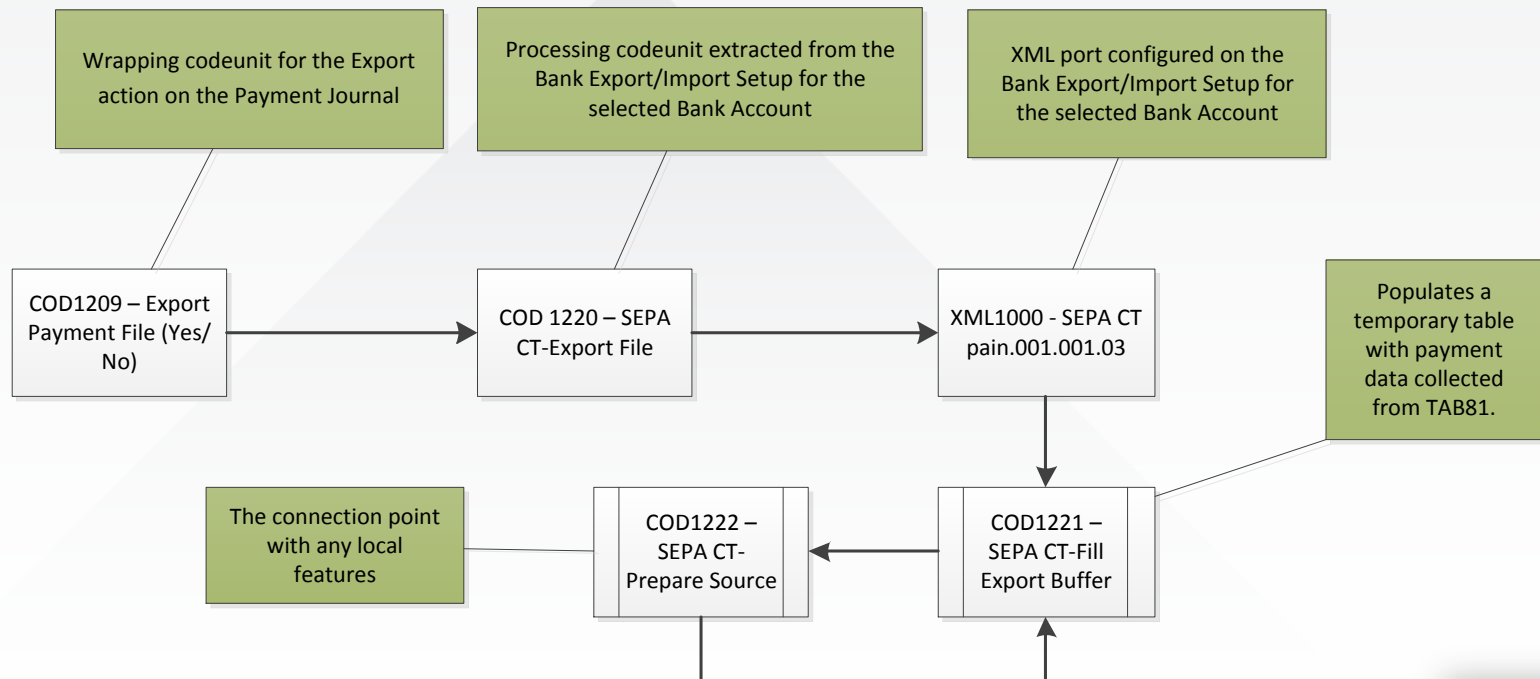
- Export of vendor payments to an xml file
- Performed through an action on the Payment Journal, based on the General Journal Line table (81)
- Local equivalent country features
 - Vendor Bills in Italy
 - Payment Slips in France
 - Cartera in Spain
 - Payment History in Holland

MAPPING W1 FEATURES TO LOCAL DATA STRUCTURES



MAPPING W1 FEATURES TO LOCAL DATA STRUCTURES

W1 Workflow



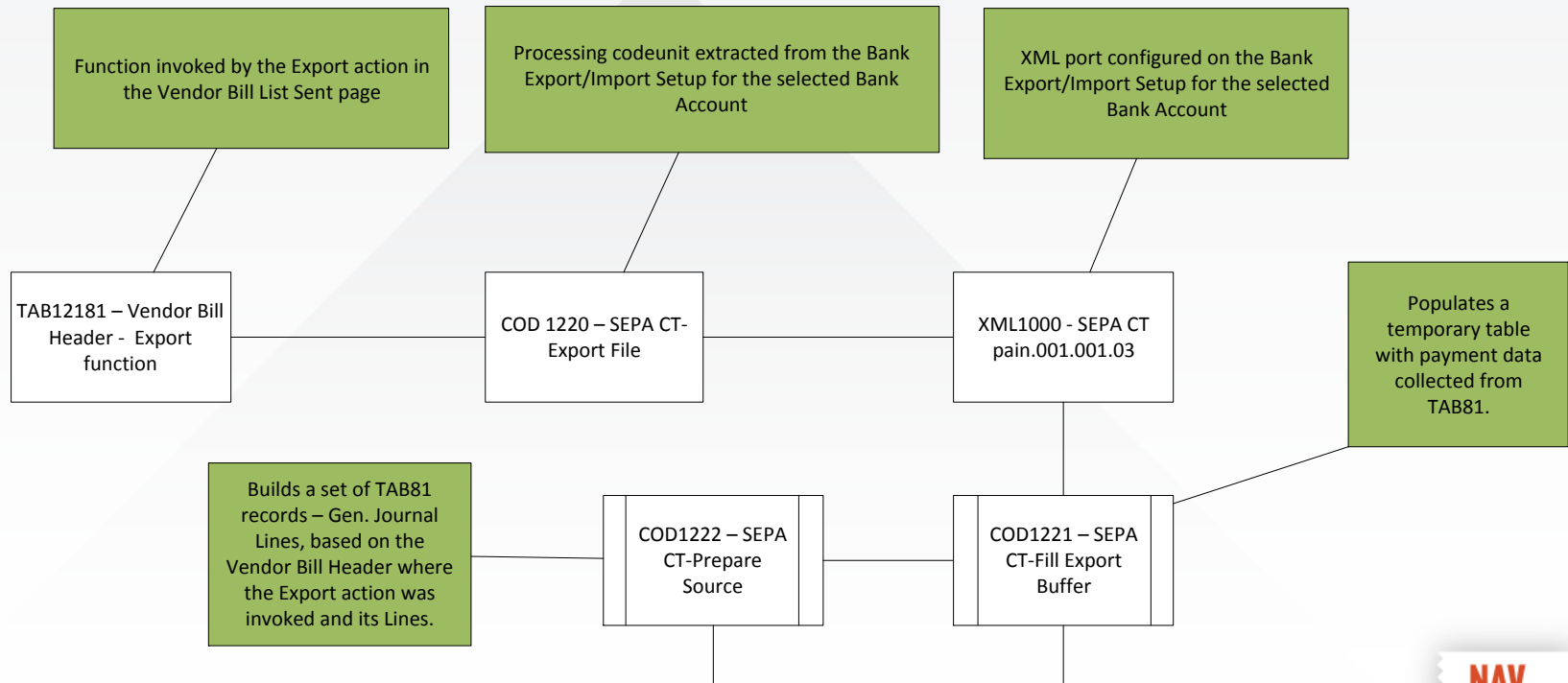
MAPPING W1 FEATURES TO LOCAL DATA STRUCTURES

Integration with the Country Version flow

- A proxy codeunit has been added in W1: 1222 – SEPA CT-Prepare Source.
- In W1, the codeunit simply outputs the same set of Gen. Journal Lines that it receives as an input
- In a country version, such as Italy, COD1222 :
 1. Gets an empty set of Gen. Journal Lines (as opposed to the real set of records to be exported, like in W1) that carry the local payment document key as a filter on the Document No. field.
 2. Selects the local payment data. E.g., in Italy, the Vendor Bill Header and Lines.
 3. Transforms the local payment data into temporary Gen. Journal Lines.
 4. Outputs the temporary Gen. Journal Lines, that will be further processed and exported exactly as in W1.

MAPPING W1 FEATURES TO LOCAL DATA STRUCTURES

Country Version Workflow



MAPPING W1 FEATURES TO LOCAL DATA STRUCTURES

Limitations

- The empty Gen. Journal Line record set carries a filter on the Document No. field.
 - The local payment document key must be maximum CODE 20 (size of the Document No. field in table 81)
- The pattern needs thorough testing on the country version
 - The constraints used when exporting in W1 might not match those of the legacy country feature
 - Easy to miss integration points.

MINI APP

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



INTRODUCTION TO MINI APP (C5 2014)

What?

Why?

How?

Mini App Demo

Mini App Patterns

INTRODUCTION TO MINI APP (C5 2014)

Goals

- Intuitive and usable out of the box
- Focus on Sales and Purchasing (Invoicing)
- Must not impair existing NAV functionality
- Simple upgrade from C5 to NAV (reuse existing code base)

INTRODUCTION TO MINI APP (C5 2014)

Implementation overview

- Added new set of objects (1300 – 1399 range)
- For behavior change we have:
 - Added code in CodeUnits (referenced from the pages)
 - Added logic to the tables (no modifications)
- Standard application pages are still accessible from Mini, but hard to get to
- Implemented automatic removal of the pages, parts and actions that are not within the license (check for object and source table) - few unsupported UI elements remaining on Standard objects

MINI APP DEMO

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



USE CONFIGURATION TEMPLATES TO SIMPLIFY AND SPEED UP DATA CREATION

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



USE CONFIGURATION TEMPLATES TO SIMPLIFY AND SPEED UP DATA CREATION

Problem

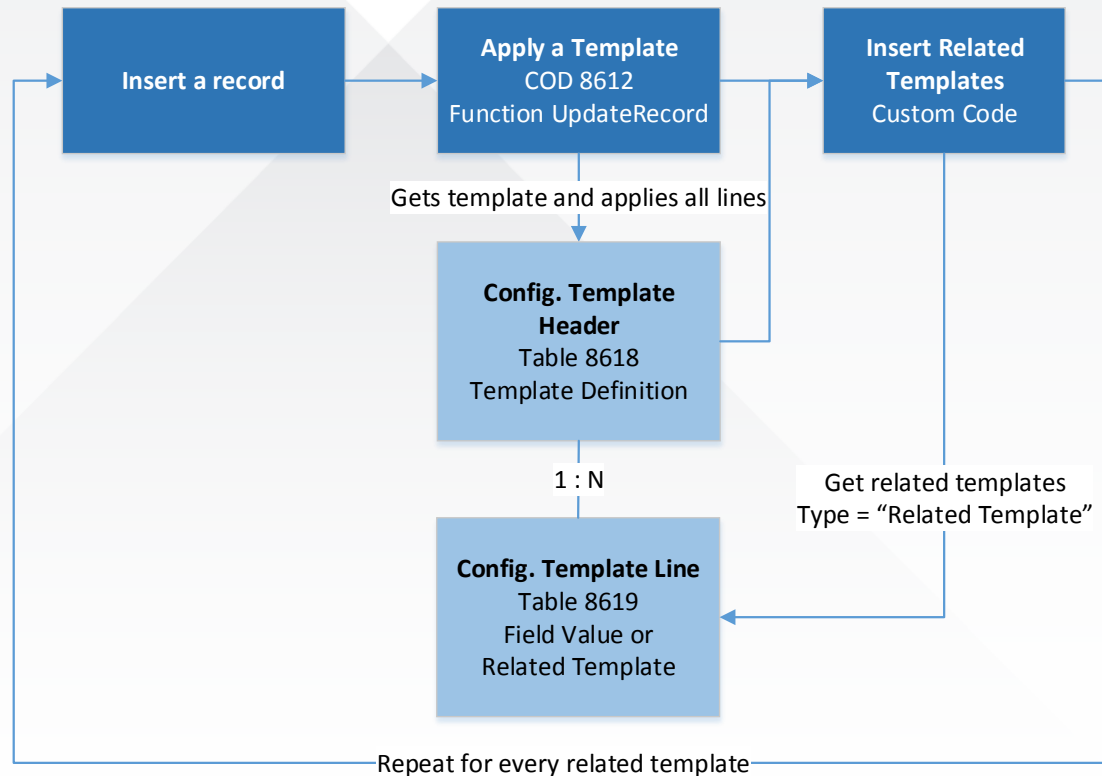
- Setting data combinations is difficult for new users (e.g. posting groups, dimensions - not sure if it is set right)
- Tedious
- Error prone

Solution

- Use templates to apply data combinations
- Users need to provide only information that varies

USE CONFIGURATION TEMPLATES TO SIMPLIFY AND SPEED UP DATA CREATION

Flow



USE CONFIGURATION TEMPLATES TO SIMPLIFY AND SPEED UP DATA CREATION

Code – Table 1300 Mini Customer Template

```
InsertCustomerFromTemplate(ConfigTemplateHeader : Record "Config. Template Header";VAR Customer : Record Customer)
// Insert a record
Customer.INSERT(TRUE);

// Apply a template
RecRef.GETTABLE(Customer);
ConfigTemplateMgt.UpdateRecord(ConfigTemplateHeader,RecRef);
RecRef.SETTABLE(Customer);

// Insert related templates
MiniDimensionsTemplate.InsertDimensionsFromTemplates(ConfigTemplateHeader,Customer."No.",DATABASE::Customer);
```

USE CONFIGURATION TEMPLATES TO SIMPLIFY AND SPEED UP DATA CREATION

Code sample to insert related templates

```
InsertDimensionsFromTemplates(ConfigTemplateHeader : Record "Config. Template Header";MasterRecordNo : Code[20];TableID : Integer)
// Find Related Templates
ConfigTemplateLine.SETRANGE(Type,ConfigTemplateLine.Type::"Related Template");
ConfigTemplateLine.SETRANGE("Data Template Code",ConfigTemplateHeader.Code);

IF ConfigTemplateLine.FINDSET THEN
REPEAT
    ConfigTemplateHeader.GET(ConfigTemplateLine."Template Code");
    // Check if it is a dimension template
    IF ConfigTemplateHeader."Table ID" = DATABASE::"Default Dimension" THEN
        // Insert Dimension and apply template
        InsertDimensionFromTemplate(ConfigTemplateHeader,MasterRecordNo,TableID);
    UNTIL ConfigTemplateLine.NEXT = 0;

LOCAL InsertDimensionFromTemplate(ConfigTemplateHeader : Record "Config. Template Header";MasterRecordNo : Code[20];TableID : Integer)
// Insert a record
DefaultDimension.INIT;
DefaultDimension."No." := MasterRecordNo;
DefaultDimension."Table ID" := TableID;
DefaultDimension."Dimension Code" := GetDefaultDimensionCode(ConfigTemplateHeader);
DefaultDimension.INSERT;

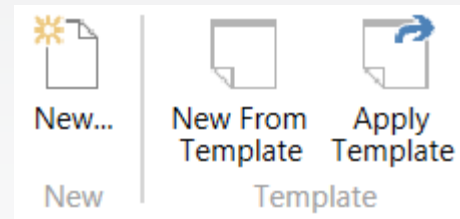
// Apply a template
RecRef.GETTABLE(DefaultDimension);
ConfigTemplateMgt.UpdateRecord(ConfigTemplateHeader,RecRef);
RecRef.SETTABLE(DefaultDimension);

// No Related Templates
```

USE CONFIGURATION TEMPLATES TO SIMPLIFY AND SPEED UP DATA CREATION

Surfacing actions in UI

- Recommended – Create New from template action
- Optional - Apply Template Action, used to change type



- Mini App Implementation – Remove New by configuration and add a application action named New

USE CONFIGURATION TEMPLATES TO SIMPLIFY AND SPEED UP DATA CREATION

Maintaining Template Definitions Option 1

- Use Configuration Template Header Page

Advantages:

- Generic – no additional code needed, can be used for any template.

Disadvantages

- Difficult to use and error prone – there is no validation, length of the field is unknown, no lookups.
- Many users will not be able to use it.

0000003

General

Code: 0000003 Table ID: 18

Description: Privatkunde (DKK, giro) Table Name: Customer

Lines

Type	Field Name	Field Caption	Template ...	Default Value	Skip...
Field		Prices Including VAT		Ja	<input type="checkbox"/>
Field		Application Method		Manuelt	<input type="checkbox"/>
Field		Payment Terms Code		14 DAGE	<input type="checkbox"/>
Field		Payment Method Code		GIRO	<input type="checkbox"/>
Field		Reminder Terms Code		DANMARK	<input type="checkbox"/>
Field		Fin. Charge Terms Code		1,5 DANSK	<input type="checkbox"/>
Field		Print Statements		Ja	<input type="checkbox"/>

OK

USE CONFIGURATION TEMPLATES TO SIMPLIFY AND SPEED UP DATA CREATION

Maintaining Template Definitions Option 2

- Make a page that resembles a document
- Use Temporary Record
- Store data to Configuration Template Header and Configuration Template Lines table

Advantages

- Better usability – lookups, simple validation, form resembles the document
- Users will be able to use it themselves

Disadvantages

- More code needed

Edit - Customer Template - Kontantbetaling/detailkunde (DKK, kontant)

HOME ACTIONS

View Edit New Delete Dimensions OneNote Notes Links Refresh Clear Filter Page

Kontantbetaling/detailkunde (DKK, kontant)

General

Template Name: Kontantbetaling/detailkunde (DKK, kontant)

Address Details

Post Code: Country/Region Code: DK

City:

Invoicing

Posting Details

Gen. Bus. Posting Group: INDLAND

VAT Bus. Posting Group: INDLAND

Customer Posting Group: DANMARK

Prices and Discounts

Customer Price Group:

Customer Disc. Group:

Allow Line Disc.: ☒

Prices Including VAT: ☒

Foreign Trade

Currency Code:

Language Code:

E-invoicing

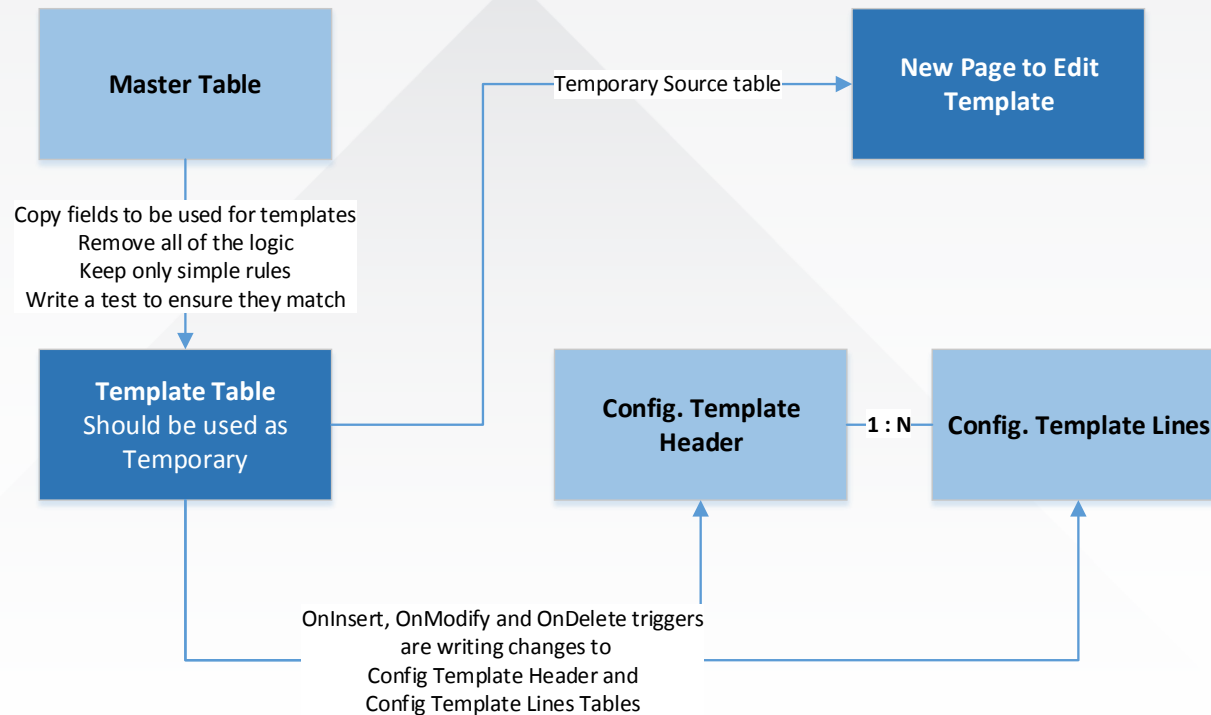
OIOUBL Profile Code:

OIOUBL Profile Code Required: ☐

OK

USE CONFIGURATION TEMPLATES TO SIMPLIFY AND SPEED UP DATA CREATION

Implementation



USE CONFIGURATION TEMPLATES TO SIMPLIFY AND SPEED UP DATA CREATION

Code – Table 1300 Mini Customer Template

```
OnInsert()  
TESTFIELD("Template Name");  
RecRef.GETTABLE(Rec);  
  
CreateFieldRefArray(FieldRefArray,RecRef);  
  
ConfigTemplateManagement.CreateConfigTemplateAndLines(Code,"Template Name",DATABASE::Customer,FieldRefArray);  
  
OnModify()  
TESTFIELD(Code);  
TESTFIELD("Template Name");  
RecRef.GETTABLE(Rec);  
  
CreateFieldRefArray(FieldRefArray,RecRef);  
  
ConfigTemplateManagement.UpdateConfigTemplateAndLines(Code,"Template Name",DATABASE::Customer,FieldRefArray);  
  
OnDelete()  
IF ConfigTemplateHeader.GET(Code) THEN BEGIN  
    ConfigTemplateManagement.DeleteRelatedTemplates(Code,DATABASE::"Default Dimension");  
    ConfigTemplateHeader.DELETE(TRUE);  
END;
```

USE CONFIGURATION TEMPLATES TO SIMPLIFY AND SPEED UP DATA CREATION

Code – Table 1300 Mini Customer Template

```
CreateFieldRefArray(VAR FieldRefArray : ARRAY [23] OF FieldRef; RecRef : RecordRef)
I := 1;

AddToArray(FieldRefArray, I, RecRef.FIELD(FIELDNO(City)));
AddToArray(FieldRefArray, I, RecRef.FIELD(FIELDNO("Credit Limit (LCY)")));
AddToArray(FieldRefArray, I, RecRef.FIELD(FIELDNO("Customer Posting Group")));
AddToArray(FieldRefArray, I, RecRef.FIELD(FIELDNO("Currency Code")));
AddToArray(FieldRefArray, I, RecRef.FIELD(FIELDNO("Customer Price Group")));
// More fields ...

LOCAL AddToArray(VAR FieldRefArray : ARRAY [23] OF FieldRef; VAR I : Integer; CurrFieldRef : FieldRef)
FieldRefArray[I] := CurrFieldRef;
I += 1;
```

USE CONFIGURATION TEMPLATES TO SIMPLIFY AND SPEED UP DATA CREATION

Test implementation

```
[Test] CustomerTemplateFieldDefinitionsMatchCustomerFields()
Initialize;

RecRefCust.OPEN(DATABASE::Customer);

TempMiniCustomerTemplate.INIT;
RecRefMiniCustTemp1.GETTABLE(TempMiniCustomerTemplate);
TempMiniCustomerTemplate.CreateFieldRefArray(FieldRefArray,RecRefMiniCustTemp1);

TemplateFieldDefinitionsMatchTableFields(RecRefCust,FieldRefArray);

LOCAL TemplateFieldDefinitionsMatchTableFields(RecRef : RecordRef;FieldRefArray : ARRAY [100] OF FieldRef)
FOR I := 1 TO ARRAYLEN(FieldRefArray) DO BEGIN
    FieldRefTemplate := FieldRefArray[I];
    FieldRefTable := RecRef.FIELD(FieldRefTemplate.NUMBER);
    ValidateFieldDefinitionsMatch(FieldRefTable,FieldRefTemplate);
END;

LOCAL ValidateFieldDefinitionsMatch(FieldRef1 : FieldRef;FieldRef2 : FieldRef)
Assert.AreEqual(FieldRef1.NAME,FieldRef2.NAME,ErrorMessageForFieldComparison(FieldRef1,FieldRef2,'names'));
Assert.AreEqual(FieldRef1.CAPTION,FieldRef2.CAPTION,ErrorMessageForFieldComparison(FieldRef1,FieldRef2,'captions'));
Assert.IsTrue(FieldRef1.TYPE = FieldRef2.TYPE,ErrorMessageForFieldComparison(FieldRef1,FieldRef2,'types'));
Assert.AreEqual(FieldRef1.LENGTH,FieldRef2.LENGTH,ErrorMessageForFieldComparison(FieldRef1,FieldRef2,'lengths'));
Assert.AreEqual(
    FieldRef1.OPTIONSTRING,FieldRef2.OPTIONSTRING,ErrorMessageForFieldComparison(FieldRef1,FieldRef2,'option string'));
Assert.AreEqual(
    FieldRef1.OPTIONCAPTION,FieldRef2.OPTIONCAPTION,ErrorMessageForFieldComparison(FieldRef1,FieldRef2,'option caption'));
Assert.AreEqual(FieldRef1.RELATION,FieldRef2.RELATION,ErrorMessageForFieldComparison(FieldRef1,FieldRef2,'table relation'));
```

USE CONFIGURATION TEMPLATES TO SIMPLIFY AND SPEED UP DATA CREATION

Ideas for Improvement

- Implement the solution in the standard application
- Personalized templates – templates per user

EASY UPDATE OF SETUP OR SUPPLEMENTARY INFORMATION

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



EASY UPDATE OF SETUP OR SUPPLEMENTARY INFORMATION

Problem

- Setup information is missing and validation error is shown
- User has to stop task at hand and figure out where to find a page in NAV and enter information needed
- Difficult and interrupting
- Then user can go back and continue task that was in process

Solution

- Instead of raising validation error prompt a page where user can supply requested information

EASY UPDATE OF SETUP OR SUPPLEMENTARY INFORMATION

Implementation

```
OnRun(VAR Rec : Record "Sales Header")
SalesSetup.VerifyAndSetOIOUBLPathSetup(SalesHeader."Document Type");
CODEUNIT.RUN(CODEUNIT::"Sales-Post + XML File",SalesHeader);
```

```
VerifyAndSetOIOUBLPathSetup("Document Type" : 'Quote,Order,Invoice,Credit Memo,Blanket Order,Return Order,Finance Charge,Reminder')
GET;
IF IsOIOUBLPathSetupAvailble("Document Type") THEN
    EXIT;
```

```
IF CONFIRM(SetupOIOUBLQst,TRUE) THEN BEGIN
    OIOUBLSetupPage.SETRECORD(Rec);
    OIOUBLSetupPage.EDITABLE(TRUE);
    IF OIOUBLSetupPage.RUNMODAL = ACTION::OK THEN
        OIOUBLSetupPage.GETRECORD(Rec);
END;
```

```
IF NOT IsOIOUBLPathSetupAvailble("Document Type") THEN
    ERROR(MissingSetupOIOUBLerr);
```

```
LOCAL IsOIOUBLPathSetupAvailble("Document Type" : 'Quote,Order,Invoice,Credit Memo,Blanket Order,Return Order,Finance Charge,Reminder') : Boolean
IF NOT FileMgt.CanRunDotNetOnClient THEN
    EXIT(TRUE);
```

```
CASE "Document Type" OF
    "Document Type"::Order,"Document Type"::Invoice:
        EXIT("OIOUBL Invoice Path" <> '');
    "Document Type"::"Return Order","Document Type"::"Credit Memo":
        EXIT("OIOUBL Cr. Memo Path" <> '');
    "Document Type"::"Finance Charge":
        EXIT("OIOUBL Fin. Chrg. Memo Path" <> '');
    "Document Type"::Reminder:
        EXIT("OIOUBL Reminder Path" <> '');
```

```
ELSE
    EXIT(TRUE);
END;
```


EASY UPDATE OF SETUP OR SUPPLEMENTARY INFORMATION

Ideas for Improvement

- Make a more generic platform implementation that launches the corresponding card page for Rec on Rec.testfield with an asterisk mark for the field that needs a proper value.

Anti-Pattern

- The anti-pattern is to do a testfield on a field that is not in the table that you are currently updating.

CREATING CUSTOM CHARTS WITH THE BUSINESS CHART ADD-IN (CHARTS IN THE WEB CLIENT)

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



CREATING CUSTOM CHARTS WITH THE BUSINESS CHART ADD-IN (CHARTS IN THE WEB CLIENT)

Problem

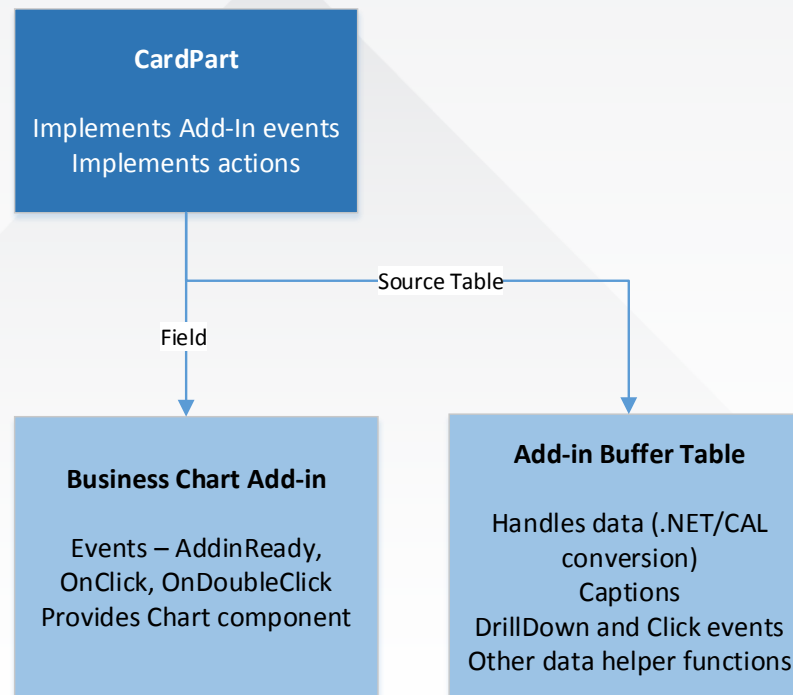
- We would like to display charts in the Web Client and Windows Client
- Cannot add custom functionality to Generic Chart
- Reuse same chart on multiple pages

Solution

- Implement CardParts with Business Chart add-in

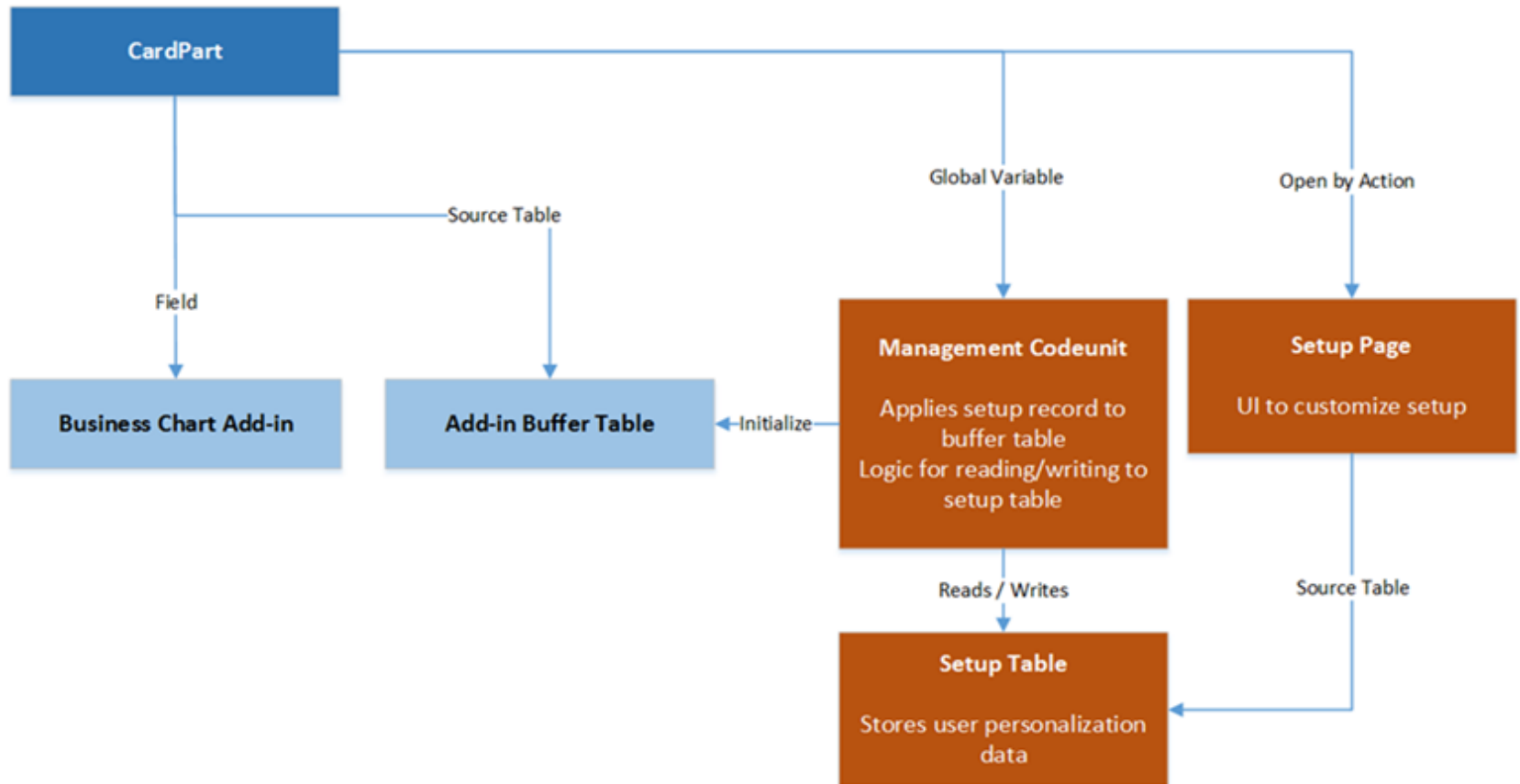
CREATING CUSTOM CHARTS WITH THE BUSINESS CHART ADD-IN (CHARTS IN THE WEB CLIENT)

Implementation



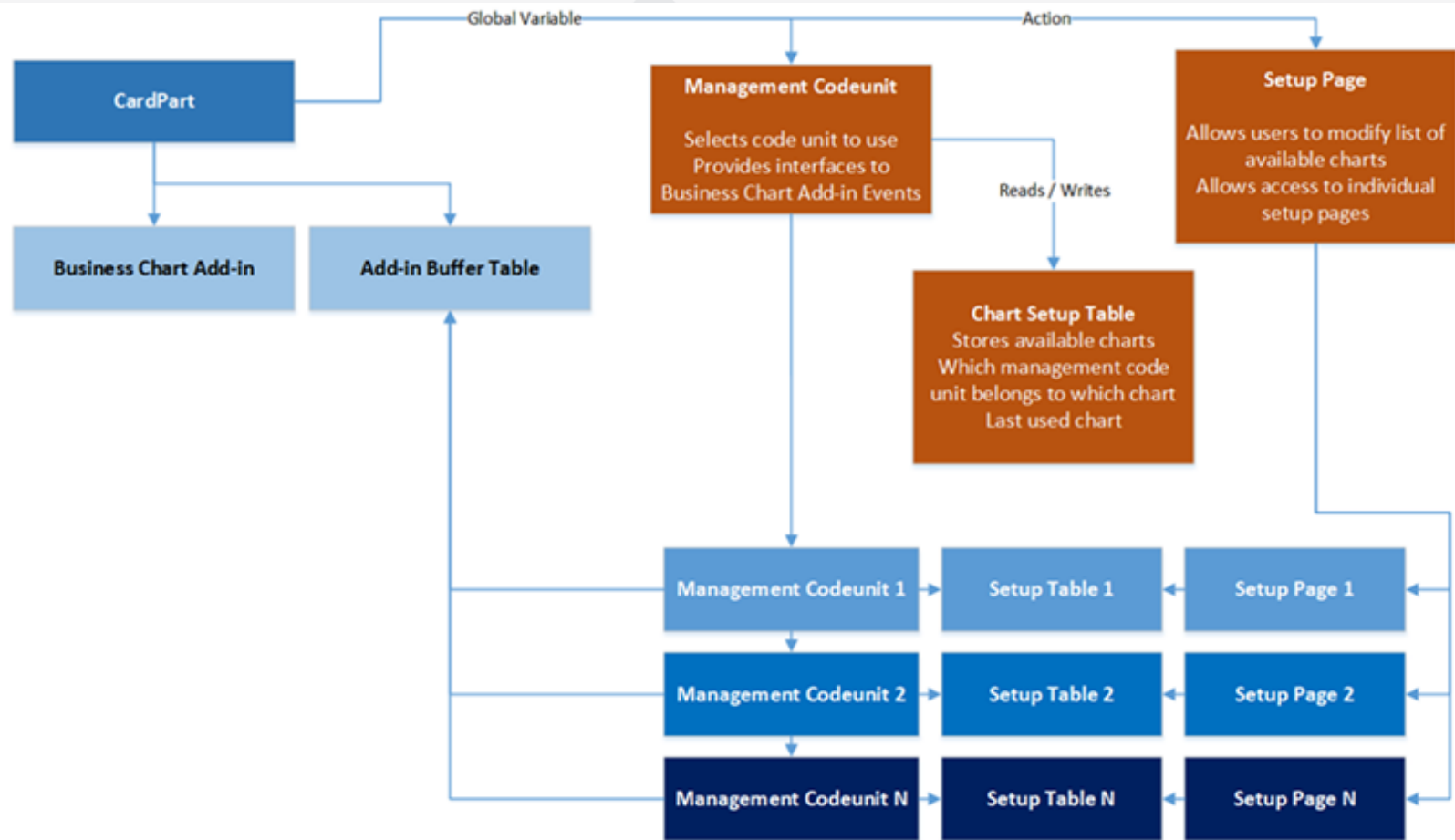
CREATING CUSTOM CHARTS WITH THE BUSINESS CHART ADD-IN (CHARTS IN THE WEB CLIENT)

Preserving and applying user personalization



CREATING CUSTOM CHARTS WITH THE BUSINESS CHART ADD-IN (CHARTS IN THE WEB CLIENT)

Showing more charts inside single part



ENHANCING LEARNABILITY WITH INTEGRATING HELP AND INSTRUCTIONS WITHIN UI

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



ENHANCING LEARNABILITY WITH INTEGRATING HELP AND INSTRUCTIONS WITHIN UI

Problem

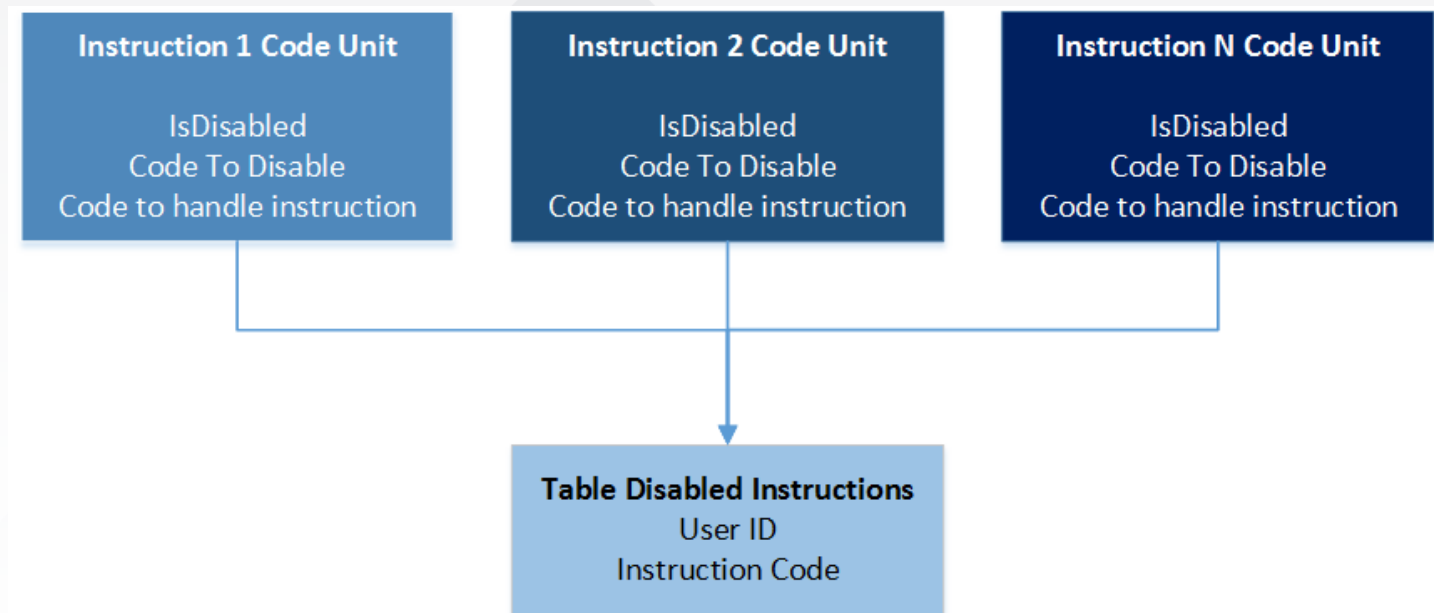
- Some functionality in NAV is difficult to use or undiscoverable
- It is impossible or very difficult to fix the the problem
- Difficult to provide training to end users

Solution

- Integrate instructions within UI
- Instructions must be dismissable – not to impair productivity or annoy the user

ENHANCING LEARNABILITY WITH INTEGRATING HELP AND INSTRUCTIONS WITHIN UI

Implementation overview



ENHANCING LEARNABILITY WITH INTEGRATING HELP AND INSTRUCTIONS WITHIN UI

Getting Started Group on Role Center

- Implement an action group on Activities hosting instructional actions
- Link help topics by creating an empty action with help icon – will invoke help topic
- Create instructional videos by using VideoPlayer add-in

Dismissable dialogs

- Use strmenu to avoid adding a new page and users dismissing the dialogs without reading it

Fast Tabs with instructional text

- Instructional text on fast tab can hold longer text without elipsis which makes it ideal for showing messages.

Tooltips

- Possible to implement tooltips for fields and actions

ENHANCING LEARNABILITY WITH INTEGRATING HELP AND INSTRUCTIONS WITHIN UI

Ideas for Improvement

- Implement tooltips and help actions across the NAV – based on the feedback we could provide out of box tooltips and some of the help topics for rest of the NAV pages. Tooltips for existing fields could be populated by a tool.
- Provide the support for the Invoking of the Help topics from C/AL code. Then we would be able to promote help actions or launch them from C/AL code if needed.

LAST BUT NOT LEAST

[NAV Patterns Wiki](#)

Any Questions?

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



THANK YOU

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





LUNCH BREAK

see you back in 60 min.

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

