

**NAV
TECH
DAYS
2016**

mibuso.com

DESIGN PATTERNS IN NAV 2017

ANDERS LARSEN, NIKOLA KUKRIKA
(MICROSOFT MDCC)

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

**NAV
TECH
DAYS
2016**

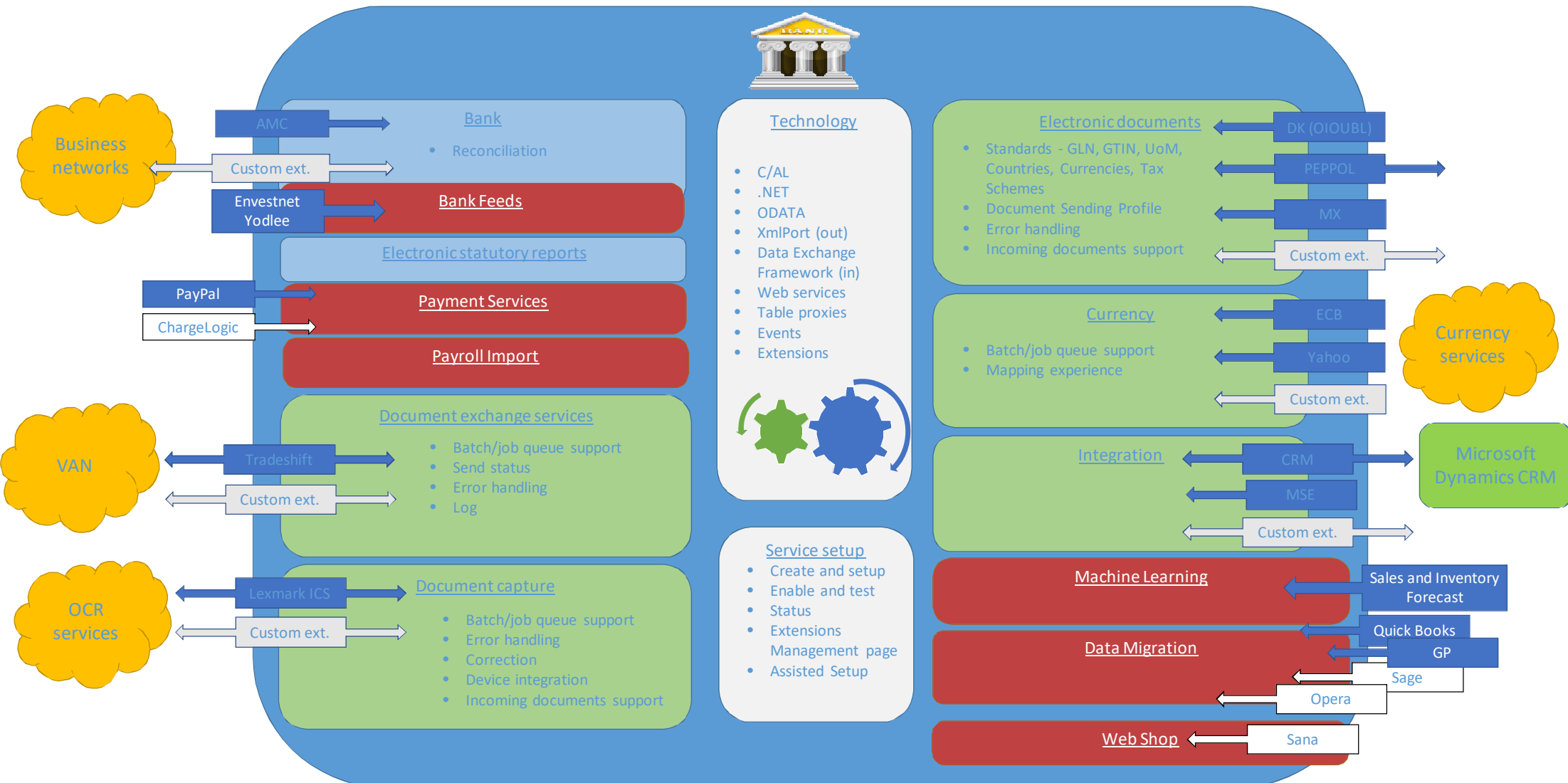
mibuso.com

Modular design in NAV?
Is it even possible?

OUR TOOLS



NAV Integration to other services



Agenda

Hammer I Events

Observer Event Pattern
Discovery
Handled
Targeted Subscriber

Hammer II Extensions

Extension Blue Print
Cached Web Service Calls
Referenced Setup Table
Record Sets

Other Cool Stuff

Feedback



Common mistakes



Common Mistakes – Try Functions

Microsoft Dynamics NAV 2016

Try functions in C/AL enable you to handle errors that occur in the application during code execution. For example, with try functions, you can provide more user-friendly error messages to the end user than those thrown by the system. You can use try functions to catch errors/exceptions that are thrown by Microsoft Dynamics NAV or exceptions that are thrown during .NET Framework interoperability operations. Try functions catch errors similar to a conditional Codeunit.Run function call, except with the following differences:

- Try function calls do not require that write transactions are committed to the database.
- Changes to the database that are made with a try function are not rolled back.

The following example illustrates the use of a try function together with codeunit 1291 **DotNet Exception Handler** to handle .NET Framework Interoperability exceptions. The code is in text file format and has been simplified for illustration. The CallTryPostingDotNet function runs the try function TryPostSomething in a conditional statement to catch .NET Framework Interoperability exceptions. Errors other than IndexOutOfRangeException type are re-thrown.

```
[TryFunction]
PROCEDURE TryPostingSomething@1();
BEGIN
    CODEUNIT.RUN(CODEUNIT::"Purch.-Post");
END;
PROCEDURE CallTryPostingDotNet @2();
VAR
    MyPostingCodeunit@1 : Codeunit 90;
```

Don't blame yourself: blame the designer.

Donald Norman

Agenda

Hammer I Events

Observer Event Pattern
Discovery
Handled
Targeted Subscriber

Hammer II Extensions

Extension Blue Print
Cached Web Service Calls
Referenced Setup Table
Record Sets

Other Cool Stuff

Feedback

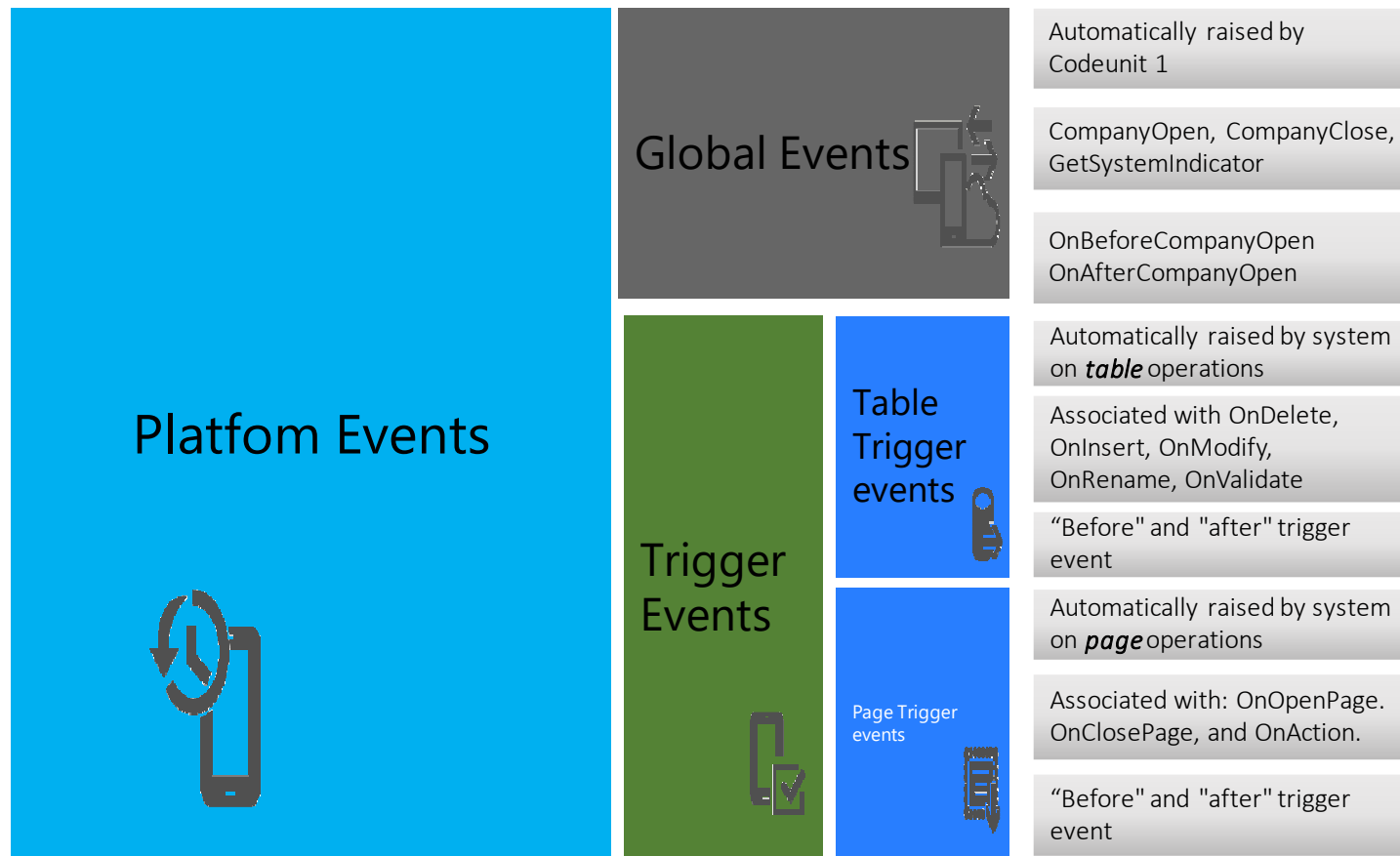


Common mistakes

Hammer I



Observer (platform) event pattern



Observer (platform) event pattern

Problem – is it possible to invoke code without calling the trigger

Solution Global triggers –fires if the trigger was not invoked, possible to track all the changes the



xRec – not the previous record



Fire on temp records



Will the code work if we have 20 subscribers on
OnAfterModifyCustomer?

Integration Events Patterns

Discovery event pattern

- Used for Setup

Handled event pattern

- Exclusive

Targeted subscriber event pattern

- Used as dedicated calls

Event patterns

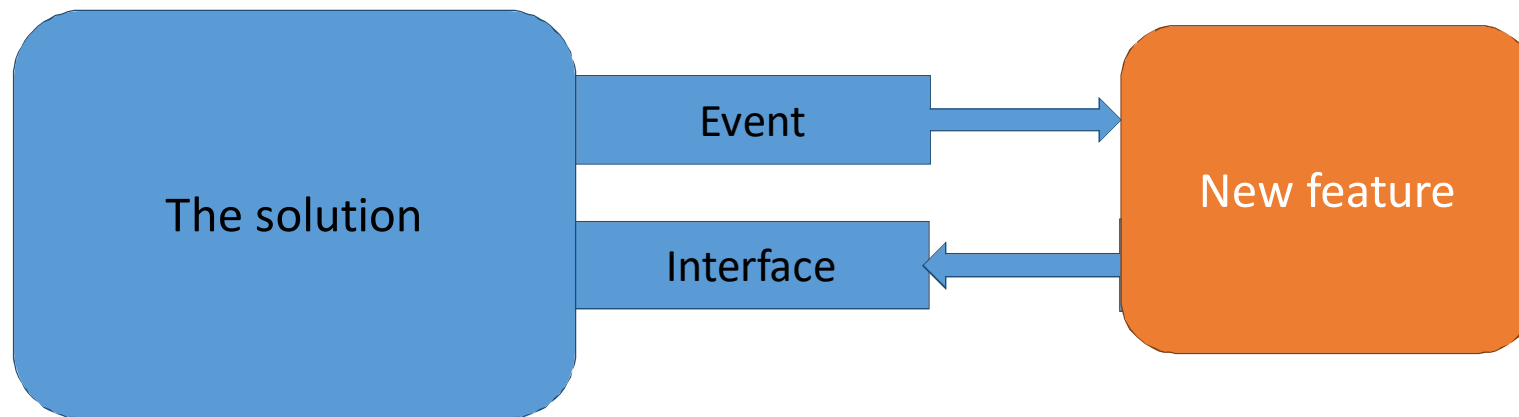
Benefits

- Easy to deploy and use
- No data upgrade at all

Consequences

- Force the developer to think of separation.

Discovery event



The discovery event pattern

All-called -> all-served

OnRun()

TheDiscoveryEvent(UpdateMe);

LOCAL [IntegrationEvent] TheDiscoveryEvent(UpDateMeRec : Record "Update Me")

LOCAL [EventSubscriber] DiscoverySubscriber1(UpdateMe : Record "Update Me")

UpdateMe.Guid := Subscriber1Guid;

UpdateMe.Text := 'I hope that I am no. 1';

UpdateMe.INSERT;

LOCAL [EventSubscriber] DiscoverySubscriber2(UpdateMe : Record "Update Me")

UpdateMe.Guid := Subscriber2Guid;

UpdateMe.Text := 'I hope that I am no. 1';

UpdateMe.INSERT;

Handled event pattern

First-called -> first-served

Used to branch code

```
OnRun()
```

```
IF NOT Handled THEN
```

```
    HandledEvent(Handled);
```

```
LOCAL [IntegrationEvent] HandledEvent(VAR Handled : Boolean)
```

```
LOCAL [EventSubscriber] HandledSubscriber1(VAR Handled : Boolean)
```

```
IF Handled THEN
```

```
    EXIT;
```

```
Handled := TRUE;
```

```
LOCAL [EventSubscriber] HandledSubscriber2(VAR Handled : Boolean)
```

```
// I didn't get called :-(
```

```
IF Handled THEN
```

```
    EXIT;
```

```
Handled := TRUE;
```


Targeted subscriber event pattern

Served when matched

OnRun()

TargetedEvent(Guid);

LOCAL [IntegrationEvent] TargetedEvent(VAR Ref : GUID)

LOCAL [EventSubscriber] TargetedSubscriber1(VAR Ref : GUID)

IF Ref <> Subscriber1Guid THEN

EXIT;

// do maciq 1

LOCAL [EventSubscriber] TargetedSubscriber2(VAR Ref : GUID)

IF Ref <> Subscriber2Guid THEN

EXIT;

// do maciq 2

Subscriber/event pattern



Parameter must match by name.





```
≡OnRun()  
  IF NOT Handled THEN  
    HandledEvent(Handled);  
    
≡LOCAL [IntegrationEvent] HandledEvent(VAR Handled : Boolean)  
    
≡LOCAL [EventSubscriber] HandledSubscriber1(VAR Handle : Boolean)  
  IF Handle THEN  
    EXIT;  
  Handle := TRUE;
```

Demo

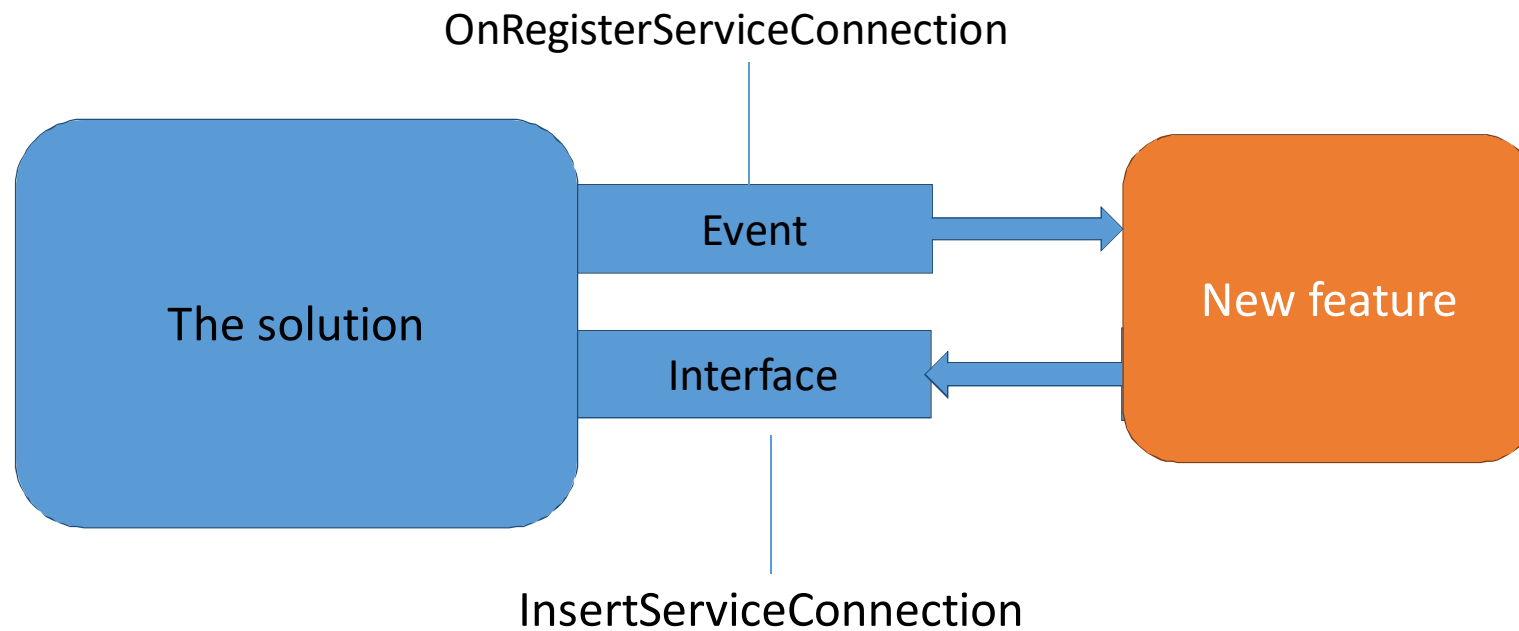
Events use cases



Discovery Event service connections

VIEW - SERVICE CONNECTIONS					
Name		Status			
Bank Data Conv. Service Setup	...	Enabled			
CRM Connection Setup	...	Disabled			
Yahoo Currency Exchange Rates Setup	...	Disabled			
Doc. Exch. Service Setup	...	Disabled			
OCR Service Setup	...	Disabled			
Online Map Setup	...	Enabled			
SMTP Mail Setup	...	Disabled			
Social Engagement Setup	...	Disabled			

Discovery Event service connections



Discovery Event service connections

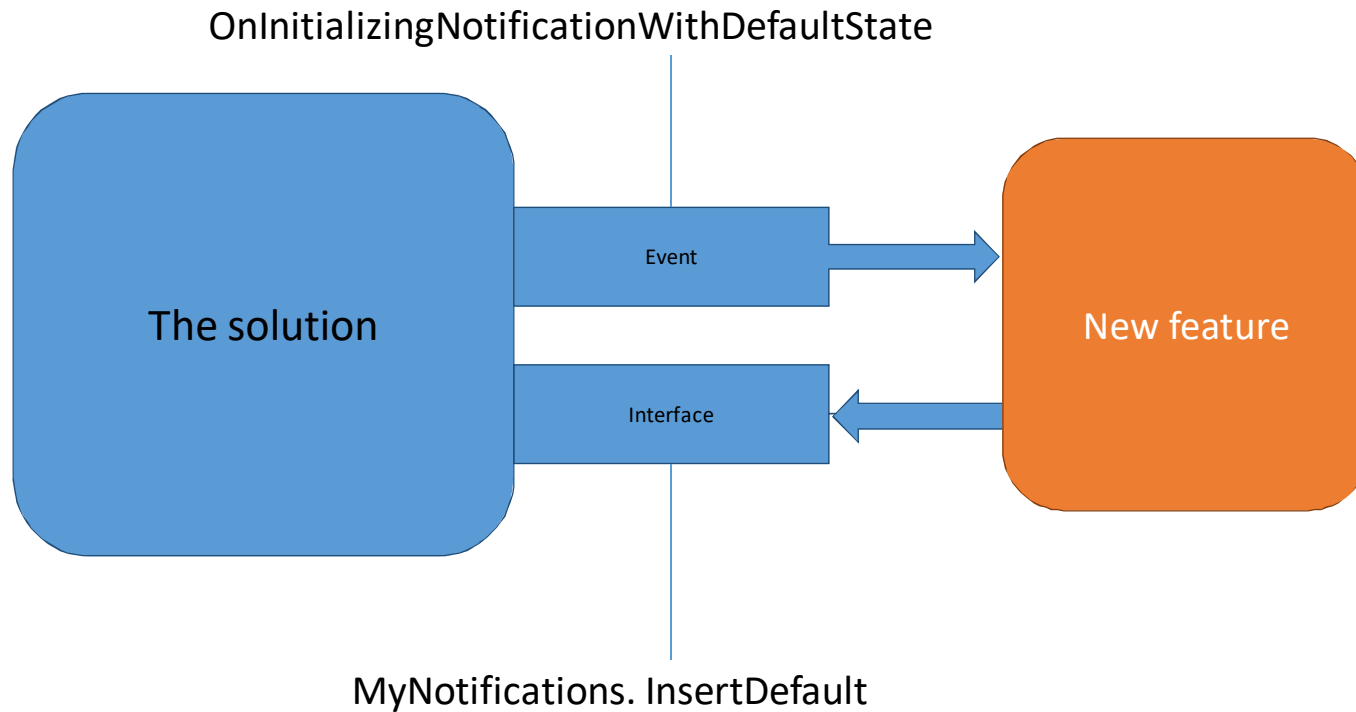
Table 1400 Service Connection - C/AL Globals	
Variables	Text Constants
Functions	
Name	
	OnRegisterServiceConnection
▶	InsertServiceConnection

Discovery Event - used for notifications

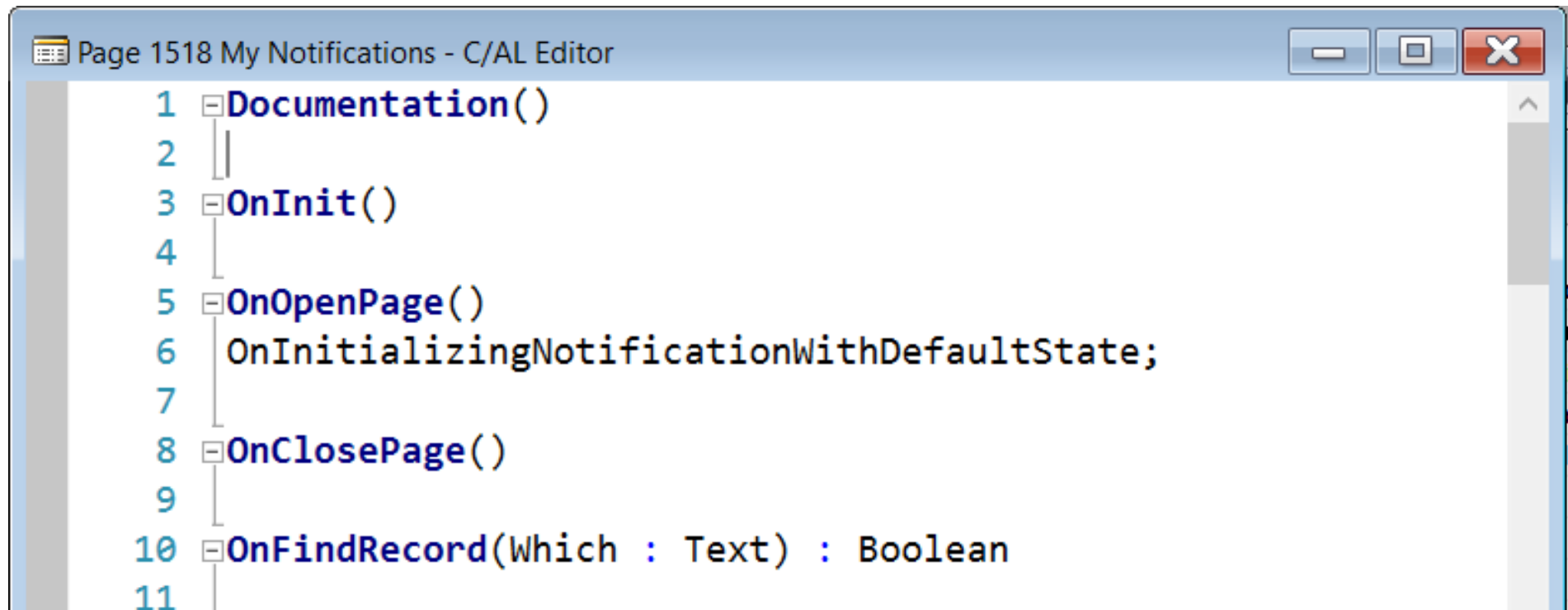
 [Edit - My Notifications](#)

Notification	Ena...	Conditions
<u>Item availability is low.</u>	<input checked="" type="checkbox"/>	(View filter details)
Customer exceeds credit limit.	<input checked="" type="checkbox"/>	(View filter details)
Warn about unposted documents.	<input checked="" type="checkbox"/>	
Confirm posting outside the fisca...	<input checked="" type="checkbox"/>	
Customer has overdue balance.	<input checked="" type="checkbox"/>	(View filter details)
Confirm after posting documents.	<input checked="" type="checkbox"/>	

Discovery Event - used for notifications



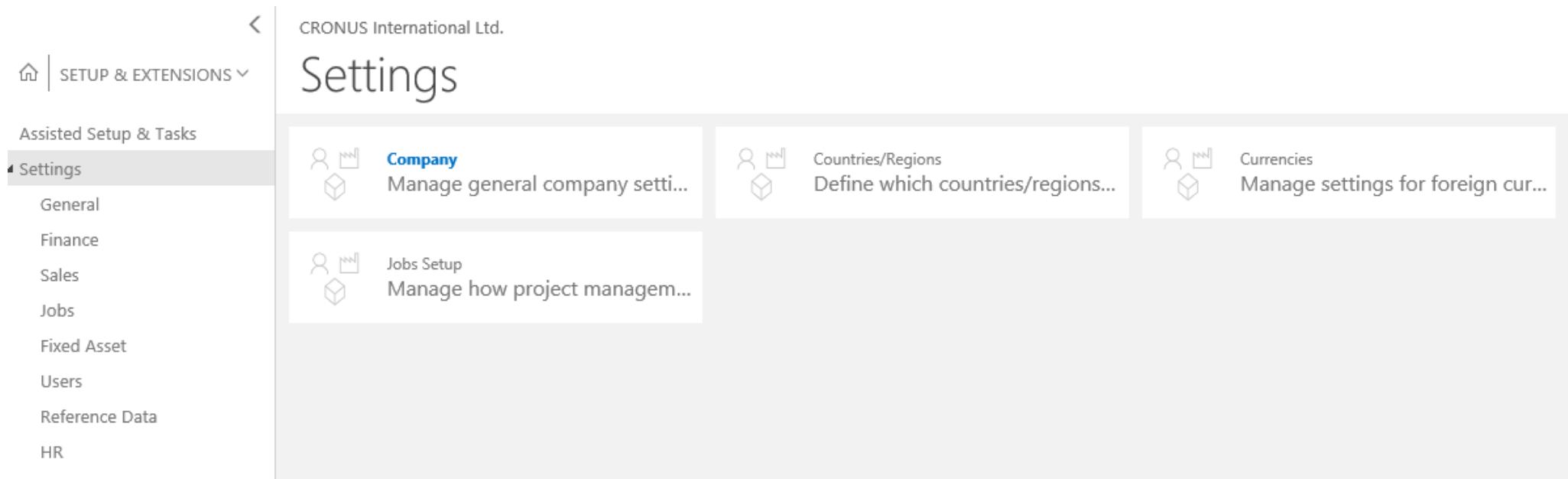
Discovery Event - used for notifications



The screenshot shows a window titled "Page 1518 My Notifications - C/AL Editor". The window contains a list of events, each preceded by a minus sign icon in a square box. The events are listed with line numbers on the left:

```
1 - Documentation()  
2 |  
3 - OnInit()  
4 |  
5 - OnOpenPage()  
6   OnInitializingNotificationWithDefaultState;  
7 |  
8 - OnClosePage()  
9 |  
10 - OnFindRecord(Which : Text) : Boolean  
11 |
```

Discovery Event - used for setup in the next release



Discovery Event & Targeted subscriber used for import of financial transactions

Problem

- Offend there is a need for importing financial from another system like Payroll

Solution

- Use the Discovery Event Plus Targeted subscriber to add the support for this


Discovery Event for import of financial transactions


Let's start with seeing the solution


HOME


ACTIONS


NAVIGATE



Delete



Post



Post and Print



Get Standard Journals...


Save as Standard Journal...


Dimensions


Reconcile


Apply Entries...


Import Payroll Transactions

Manage

Process

Bank

Application

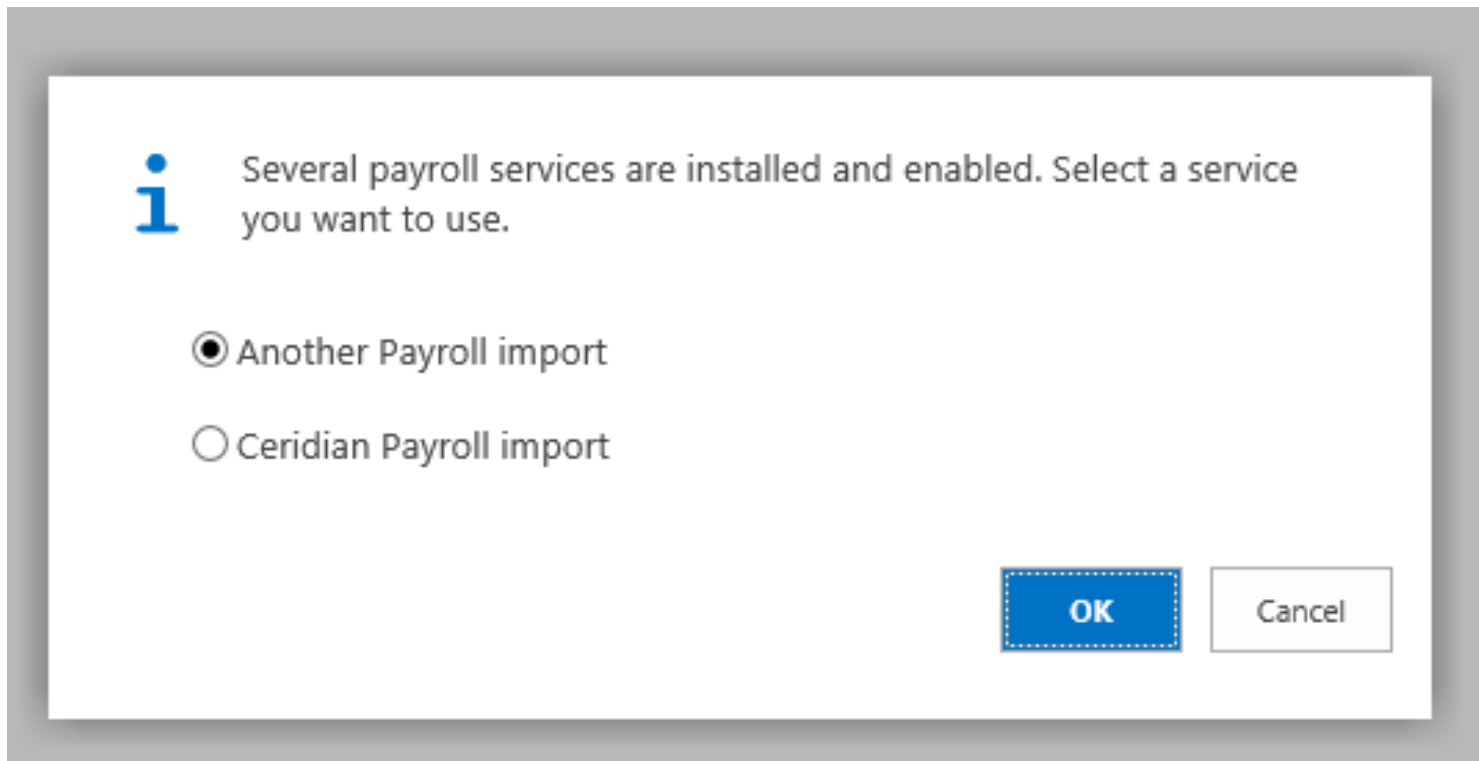
Payroll

EDIT - GENERAL JOURNAL

Batch Name

DEFAULT

Targeted subscriber used for import of financial transactions



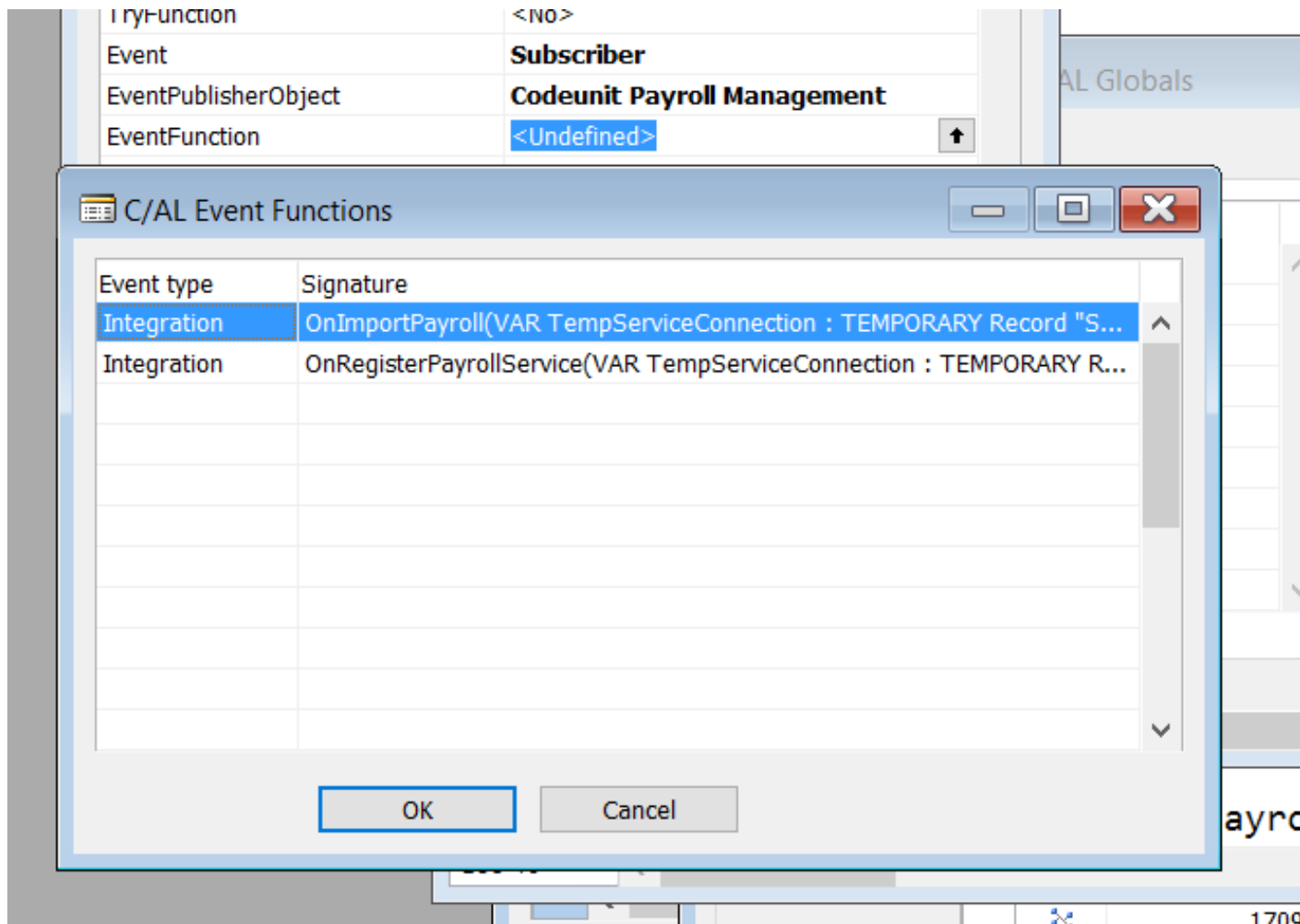
i Several payroll services are installed and enabled. Select a service you want to use.

☒ Another Payroll import

☐ Ceridian Payroll import

OK Cancel

Discovery Event & Targeted subscriber used for import of financial transactions



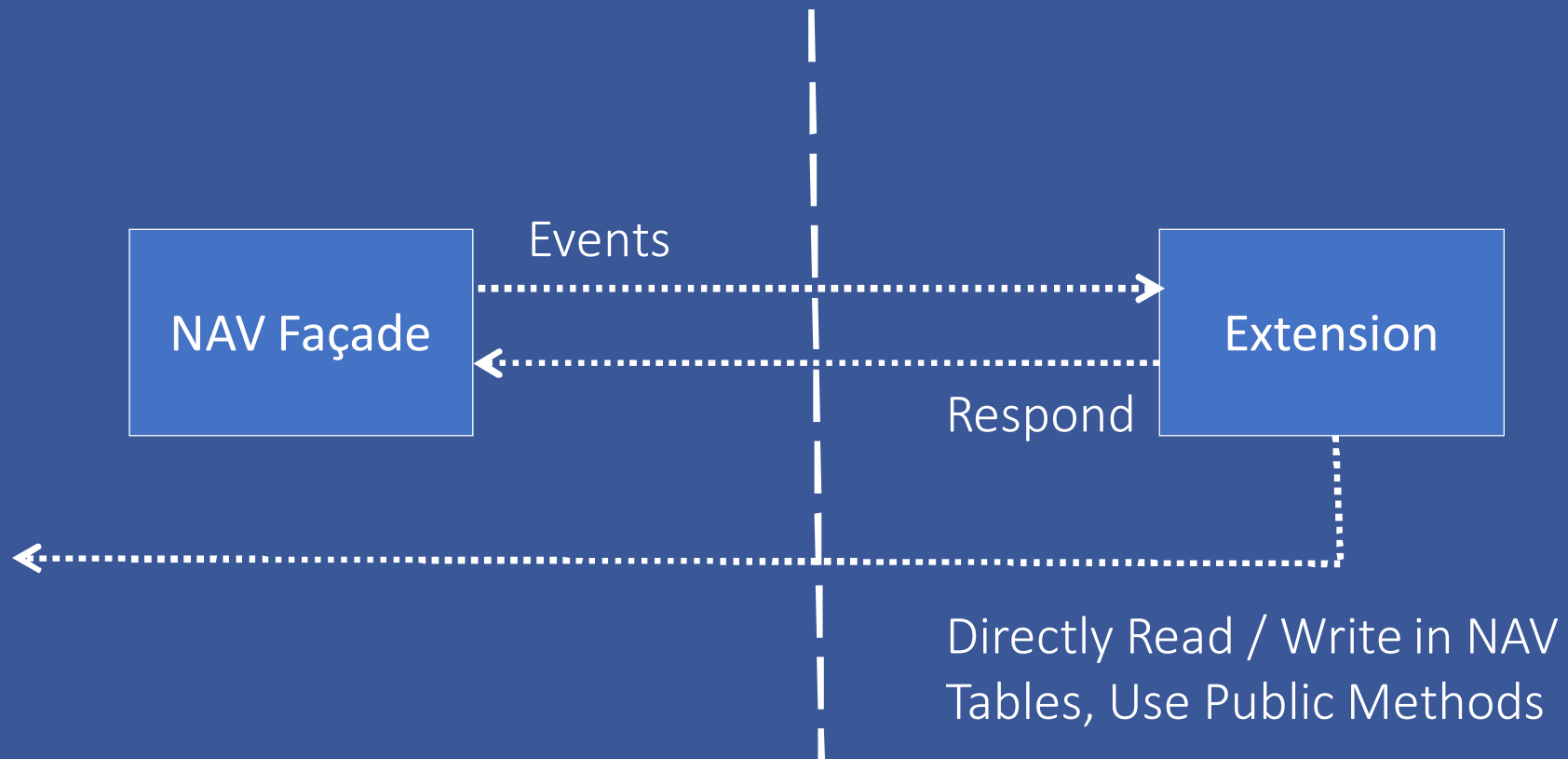
Hammer II



Extensions blueprint

NAV

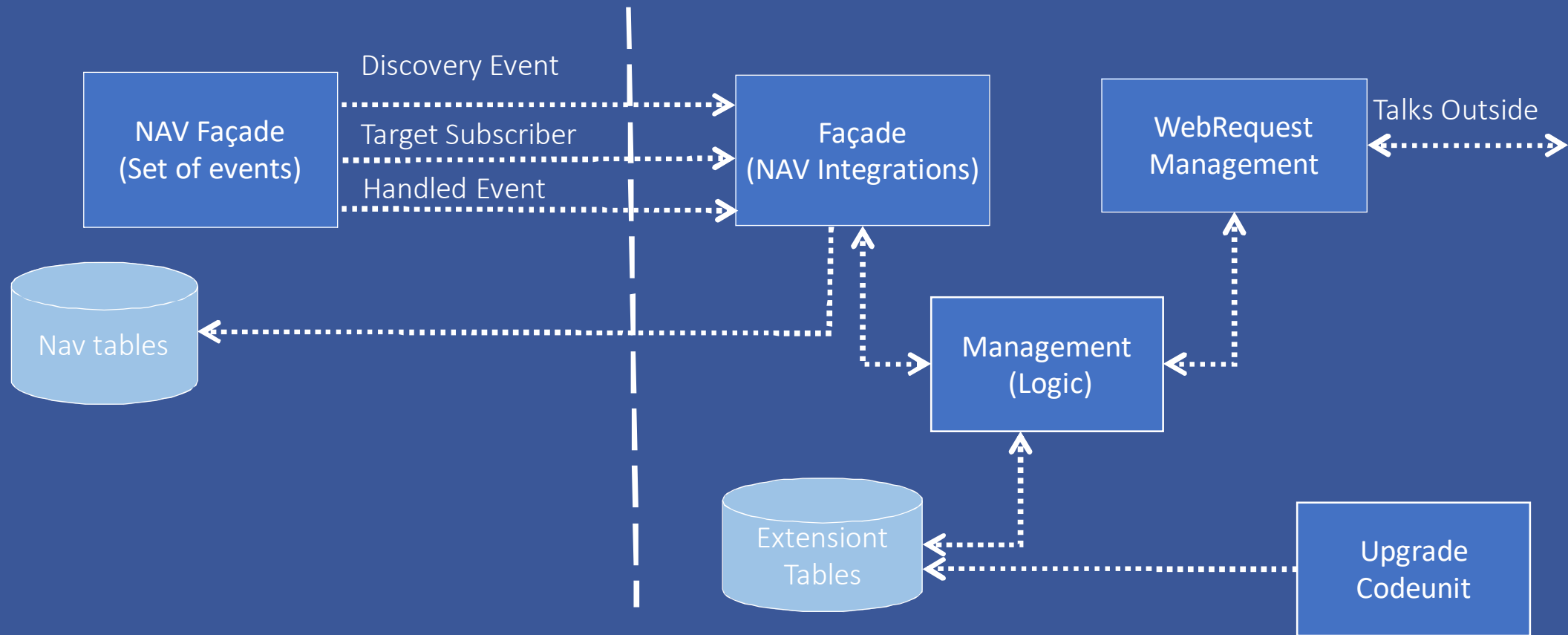
Extension



Extensions blueprint

NAV

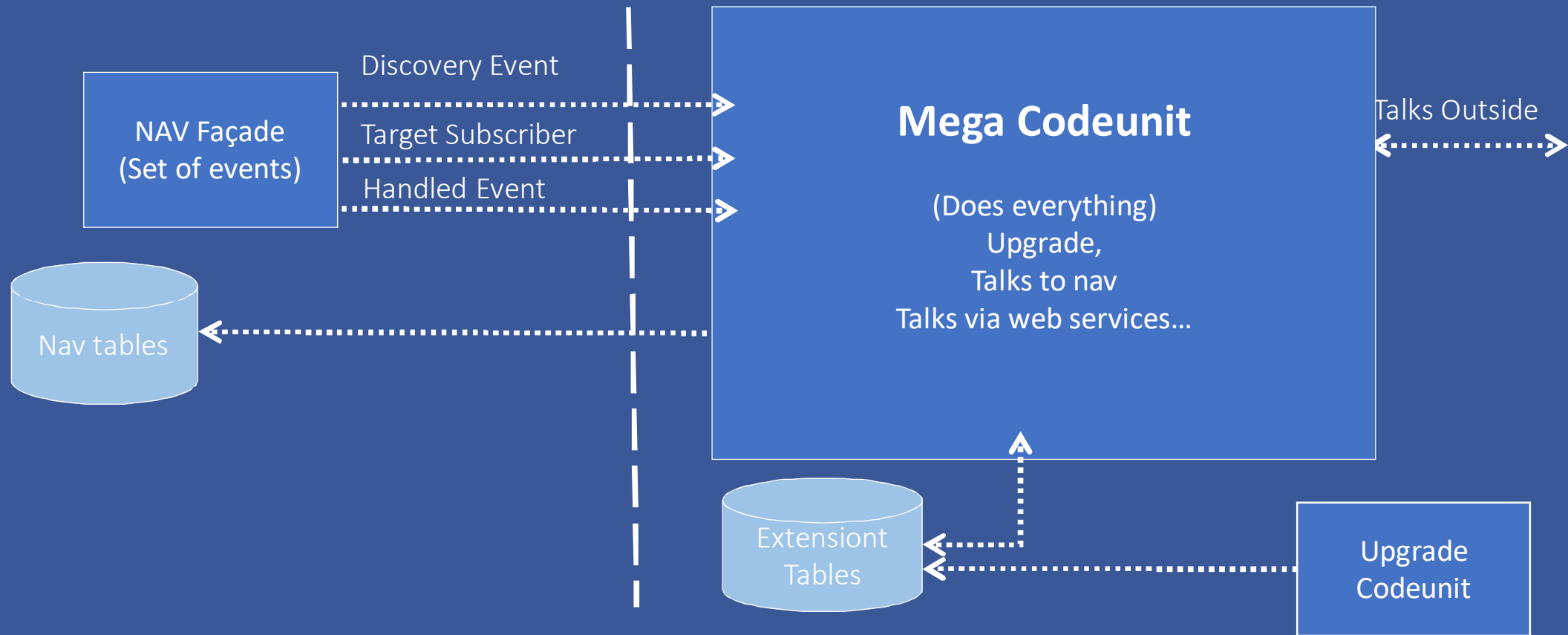
Extension



Avoid Mega Codeunit

NAV

Extension



5 Commandments (We are working on it...)

1. Name specific Codeunits with proper name

Why? Readability + not hardcoding to a specific concept

2. Never ever reference specific concept from base code - All calls must go through an event

Why? With events we can replace one implementation with another

3. Provide proper concepts and models

Start by asking what is Lexmark, PayPal, Yodlee...

The name will give you a larger Concept (Lexmark is OCR Service, Yodlee is Bank Feed Aggregator, PayPal is payment service). All events and objects should follow the naming

Why? – Readability, maintainability, ensuring we have a proper events in place

4. There is no one to clean up

Make sure that the data is in the good condition and you can do proper upgrade. Prefer using Extensions tables (Better performance, more control)

Why??? – This will help you make sure you have a modular design + plus it is how the extensions work, so...

5. Do proper upgrade

Why? Because you have to



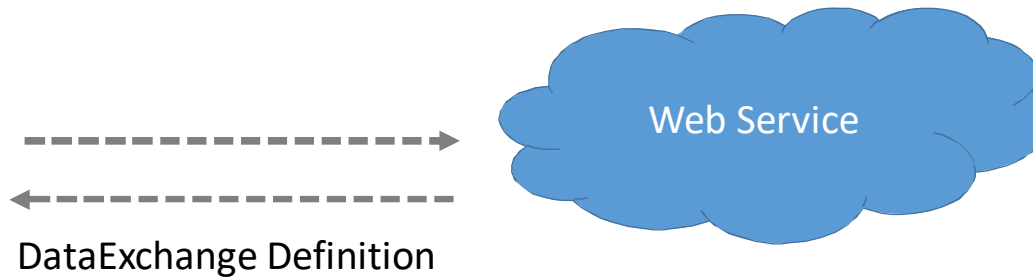
Proper upgrade

Upgrade Horror Story

Based on actual code

Proper upgrade

```
[- OnNavAppUpgradePerDatabase()  
|  
[- OnNavAppUpgradePerCompany()  
  // Some Code  
  LoadDataExchangeDefinition;  
  // Other Code
```



Several problems:

1. Did not Handle Exceptions - Must not throw exceptions in the Upgrade Code
2. *Did not understand how the "upgrade" works*

How Upgrade works

```
[- OnNavAppInstallPerDatabase()
```

```
[- OnNavAppInstallPerCompany()
```

```
  // Some Code
```

```
  LoadDataExchangeDefinition;
```

```
  // Other Code
```

----->

-----<

DataExchange Definition



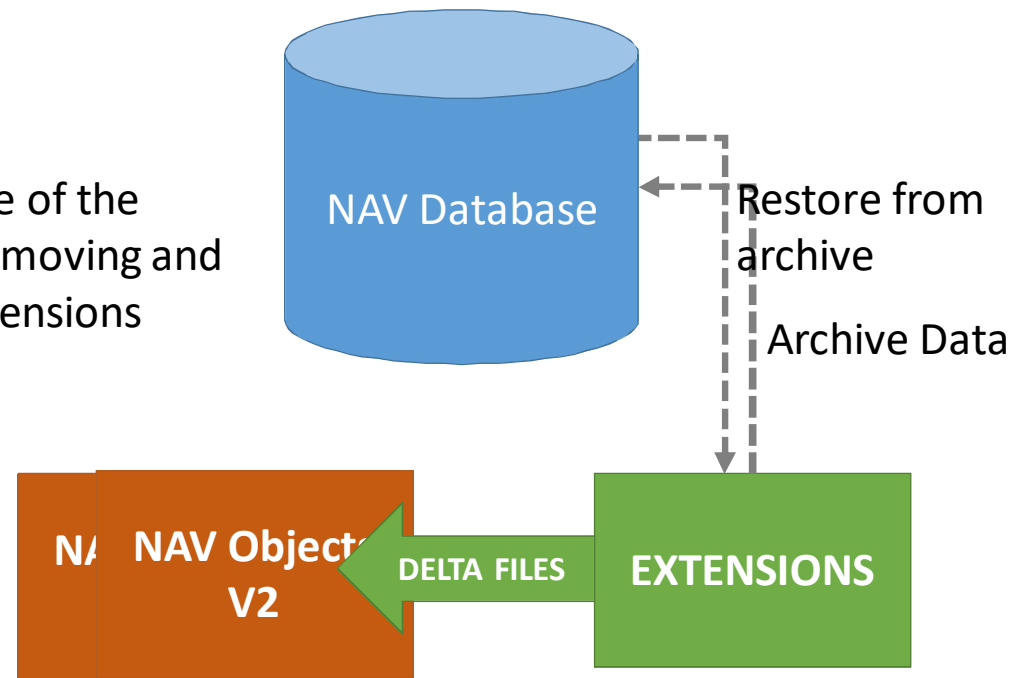
OnNavAppUpgradePerDatabase and PerCompany – fire and publish always on first install

There is no check for version

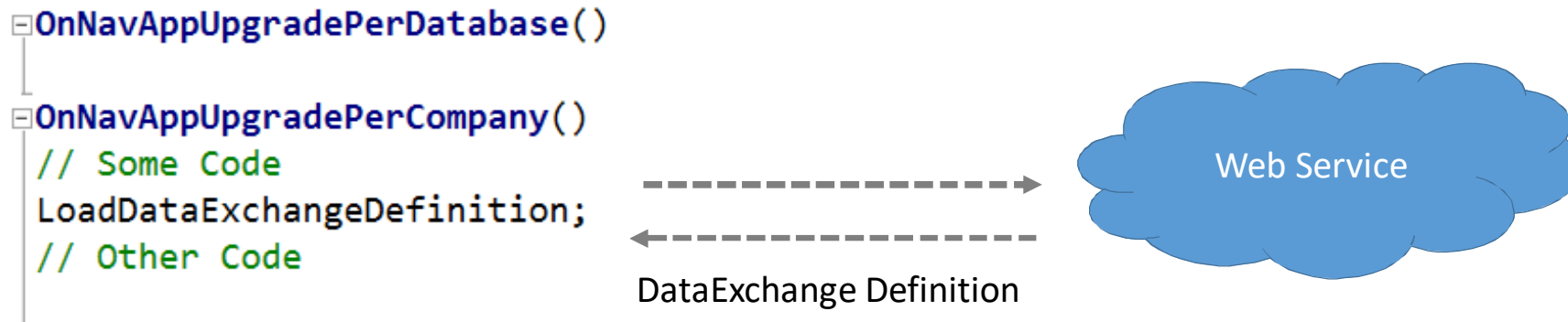
They should have been OnInstall instad of OnUpgrade

How Upgrade works

On every upgrade of the service we are removing and restoring The Extensions



How Upgrade works



Will slow down the upgrade speed

We are generating a lot of traffic for nothing - 100.000 users = 100.000 downloads

Someone will have to pay the cost
Every month

If you are hosting it your self - Could bring the service down

How to fix it – Cached Web Service Calls

1. Prepopulate data in the NAVX file, restore it OnUpgradePerDatabase
2. Check if new version is available
3. Load the definition at the convenient time, if not possible use cache
(Schedule a Job Queue, when User opens the page...)

Referenced Setup Table

Bank Account X

No. ...

Name

[Select Bank Statement Provider](#)

Discovery
Event +

Argument

Table

Select Statement Provider X

Yodlee

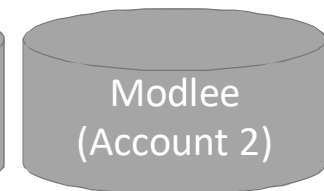
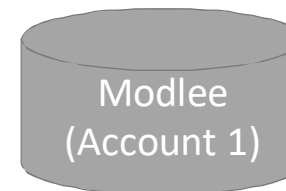
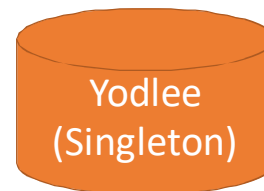
Modlee 1

Modlee 2

OK Cancel

How do we store

selection?



Referenced Setup Table

Problem

Different extensions can be using different setup tables.
Only one can be selected.

Solution

Reference an extension table through RecordID

RecordID Gymnastics

```
IF RecRef.GET(SetupRecordID) THEN // Check if it exists
```

```
RecRef := SetupRecordID.GETRECORD;
```

```
CASE SetupRecordID.TABLEN OF // Check the table no
```

```
BankAccount.SETRANGE("Bank Stmt. Service Record ID",SetupRecordID); // Set Range
```

```
IF BlankRecordID = SetupRecordID THEN // Check if it is blank
```

```
OnTargetedSubscriber(SetupRecordID) // Invoke Targeted Subscriber
```

Referenced Setup Table

Benefits

- Freedom when defining your own tables
- Easy lookup if the record exist
- Single field



Drawbacks

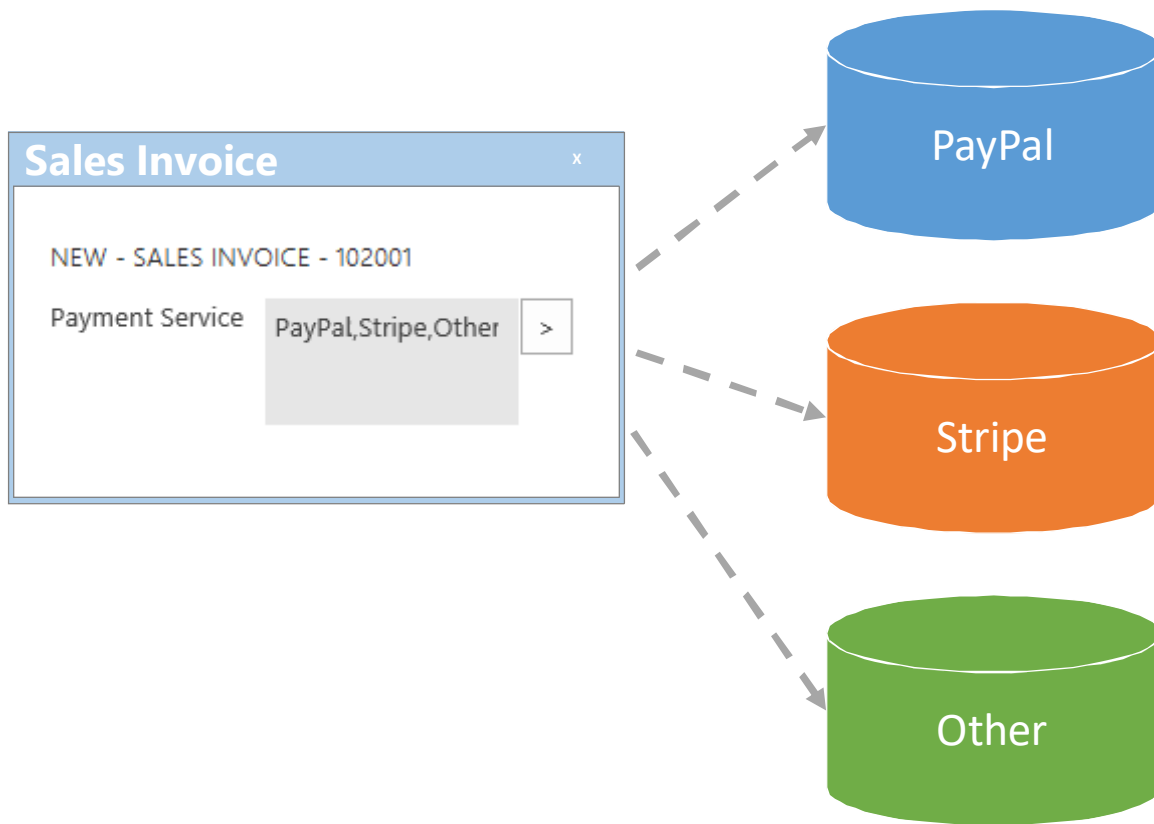
- RecordID takes too much byte space on the SQL row (448 bytes - 8000 limit)

- Could be replaced by the 2 fields - Table ID (int) and Code (10)

- Filtering on RecordID has rather bad performance on big data sets

Record Sets

Reference 3 different other tables from Sales Invoice
They are included by default



Add a comma separated field with values

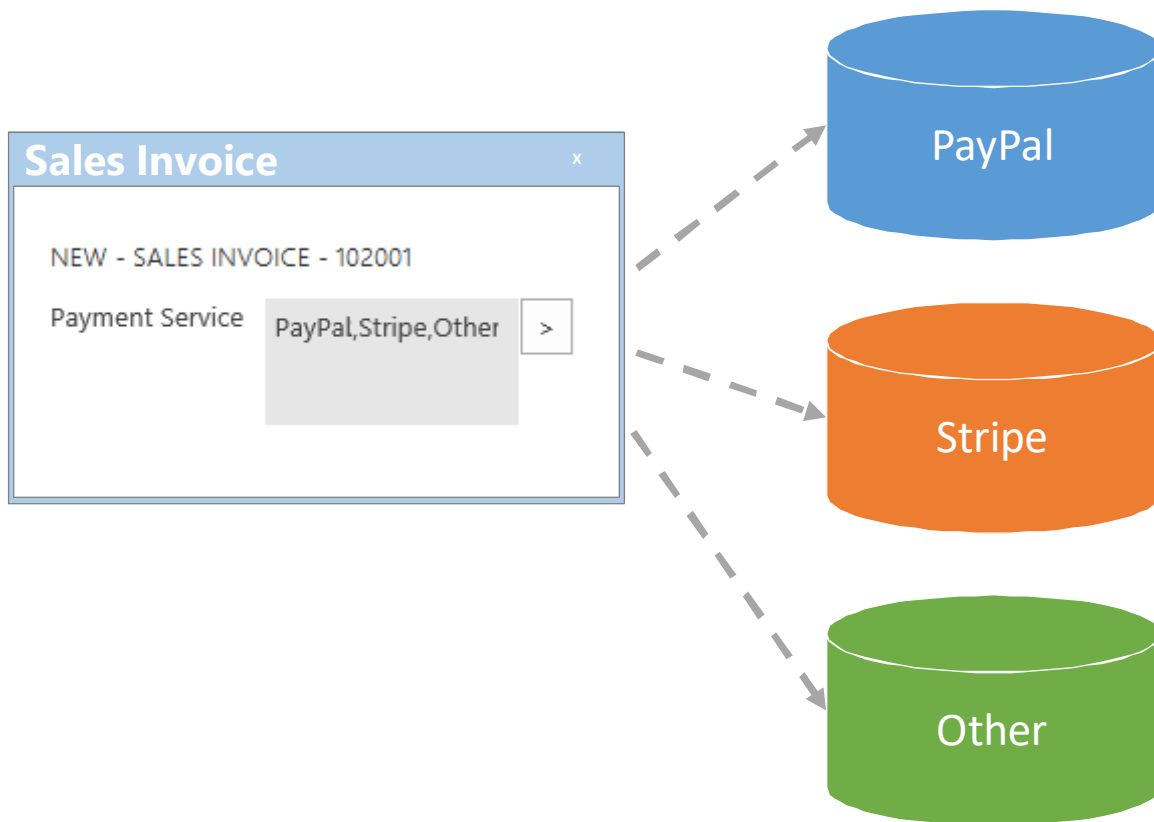
The screenshot shows a 'Table 36 Sales Header' window. A large speech bubble with the text 'Yuck!' is overlaid on the table, indicating a problem with the data. The table has columns for 'Field No.', 'Field Name', 'TableId', 'Code', and 'Value'. The following table represents the data visible in the screenshot:

Field No.	Field Name	TableId	Code	Value
5803	Last			
7001	Allow Line Disc.			
7200	Get Shipment Used	Boolean		
9000	Assigned User ID	Code	50	
30000	Payment Services	Text		250

TableId;Code;
80000;PayPal;90000;Stripe;10000;Other;

Record Sets

Reference 3 different other tables from Sales Invoice
They are included by default



Lets make it a blob so it does not have size limitation

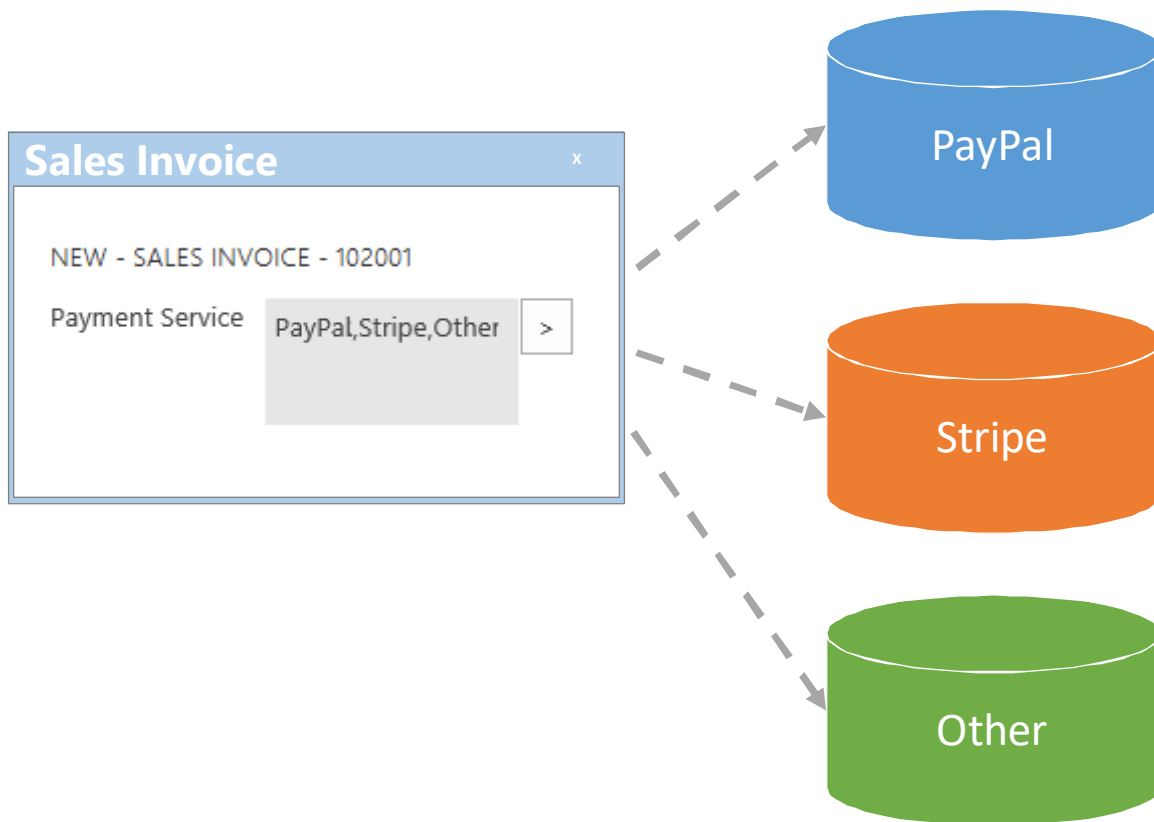
The screenshot shows a table editor window titled 'Table 36 Sales Header - Table 36'. A large speech bubble with the text 'Mega Yuck!' is overlaid on the table. The table has columns for Field No, Field Name, Data Type, and Code. The row for '30000 Payment Services' is highlighted, showing a 'BLOB' data type.

Field No	Field Name	Data Type	Code
5800	Last Modified	Boolean	
5803	Last Modified	Boolean	
7001	Allow Line Disc.	Boolean	
7200	Get Shipment Used	Boolean	
9000	Assigned User ID	Code	50
30000	Payment Services	BLOB	

TableId;Code;
80000;PayPal;90000;Stripe;10000;Other;

Record Sets

Reference 3 different other tables from Sales Invoice
They are included by default



Traditional implementation

Sales Invoice No.	Payment Service (Code/Table, RecordID)
102001	PayPal
102001	Stripe
102001	Some Other
102002	PayPal
102002	Stripe
102002	Some Other
....	

Record Sets

Problem

Grows with no of invoices

No of invoices x 3 in average – if we have 100.000 invoices, we will have max 300.000 records in this table

More than 90% of Duplicated data

Have to decorate every change to the Record (Post, Make Order from Quote, Archive...)

Maintenance of tables (Rename, modify, update, delete is costly due to the large number of records)

Solution

Use Record Sets

Record Sets

Sales Invoice X

Sales Invoice X

Sales Invoice X

NEW - SALES INVOICE - 102001

Payment Service

PayPal,Stripe,Other

 >

Set ID = 1

Sales Invoice X

NEW - SALES INVOICE - 102002

Payment Service

PayPal,Stripe

 >

Set ID = 2

Sales Invoice X

NEW - SALES INVOICE - 1020021

Payment Service

PayPal

 >

Set ID = 3

Set ID	Payment Service
1	PayPal
1	Stripe
1	Some Other
2	PayPal
2	Stripe
3	PayPal

For 3 Payment Services 8 sets max = 10 records
max (18 in total) VS 300.000 from old design


No of rows does not grow with no of invoices,
but with no of combinations used

Record Sets - Code

COD 8400 - Record Set Management

TAB 8402 - Record Set Buffer (Argument Table Pattern)

Generic - 4 objects in total

	8400	Record Set Definition
	8401	Record Set Tree
	8402	Record Set Buffer
	8400	Record Set Management

Dimensions – 3 tables, 3 pages

	480	Dimension Set Entry
	481	Dimension Set Tree Node
	482	Reclas. Dimension Set Buffer
	479	Dimension Set Entries
	480	Edit Dimension Set Entries
	699	Dimension Set Entries FactBox

Record Sets - Code

```
GetSet(VAR TempRecordSetBuffer : TEMPORARY Record "Record Set Buffer";SetID : Integer)
```

```
SaveSetSingleTable(RecordsVariant : Variant) : Integer
```

Different tables - Populate Record Set Buffer with Record ID

Call SaveSet function

```
SaveSet(VAR TempRecordSetBuffer : TEMPORARY Record "Record Set Buffer") SetID : Integer
```

```
RenameRecord(RecRef : RecordRef;xRecRef : RecordRef)
```

Decorate your rename trigger with:

```
RecordSetManagement.RenameRecord(RecRef,xRecRef)
```

Record Sets – Under the hood

Sales InvoiceX

NEW - SALES INVOICE - 102001

Payment Service PayPal,Stripe,Other >

Sales InvoiceX

NEW - SALES INVOICE - 102002

Payment Service PayPal,Stripe >

Sales InvoiceX

NEW - SALES INVOICE - 1020011

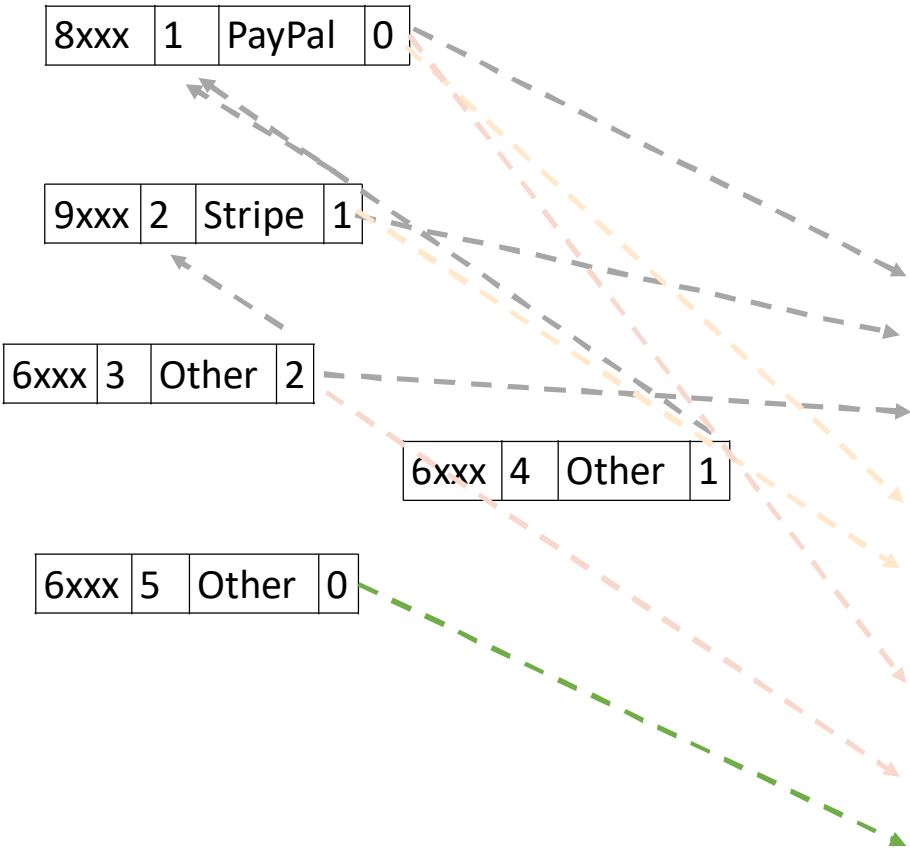
Payment Service PayPal,Other >

Sales InvoiceX

NEW - SALES INVOICE - 1020011

Payment Service Other >

Record Set Tree
(Table ID, Node ID, Record ID, Parent ID)



Record Set Definition

Set ID	Node ID	Record ID
1	1	PayPal
1	2	Stripe
1	3	Other
2	1	PayPal
2	2	Stripe
3	1	PayPal
3	4	Other
4	5	Other

Record Sets

Benefits

Saves space significantly

Easy pattern to use and apply

Set ID follows the record

(Post, Make Order, Delete...) – no need to change signatures or modify code

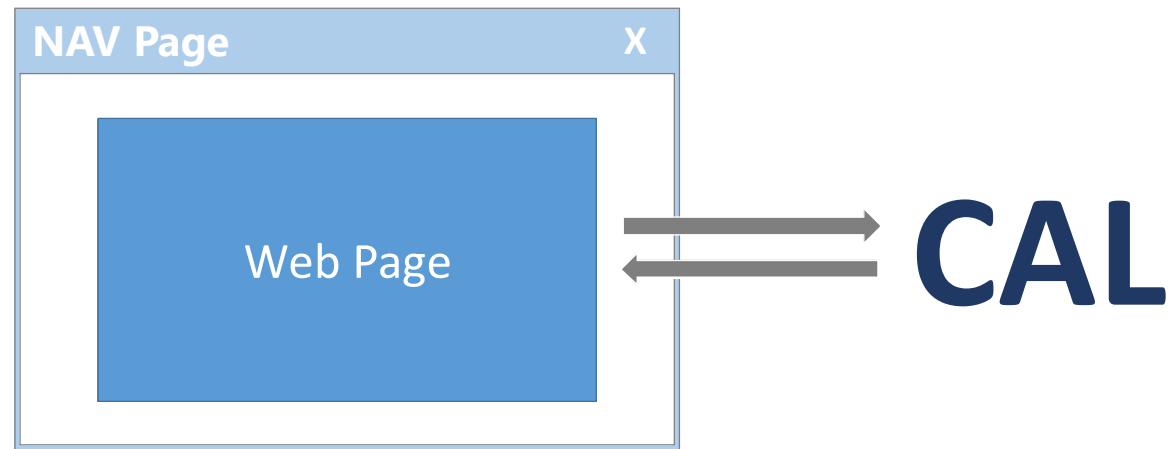


Drawbacks

Unless there is a common pattern, there is no reuse - we will end up using more space and add complexity

Order is not preserved, we will always sort ascending on the key

Cool Stuff – WebPageViewer



WebPageViewer – How to show a page

1. Add it as an add-in to a page

Add-in Name	Public Key Token	Description
Microsoft.Dynamics.Nav.Client.BusinessChart	31bf3856ad364e35	Microsoft Dynamics BusinessChart control add-in
Microsoft.Dynamics.Nav.Client.DynamicsOnlineConnect	31bf3856ad364e35	Microsoft Dynamics Online Connect control add-in
Microsoft.Dynamics.Nav.Client.OAuthIntegration	31bf3856ad364e35	Microsoft OAuth Integration control add-in
Microsoft.Dynamics.Nav.Client.PageReady	31bf3856ad364e35	Microsoft Dynamics PageReady control add-in
Microsoft.Dynamics.Nav.Client.PingPong	31bf3856ad364e35	Microsoft Dynamics PingPong control add-in
Microsoft.Dynamics.Nav.Client.SocialListening	31bf3856ad364e35	Microsoft Social Listening control add-in
Microsoft.Dynamics.Nav.Client.TimelineVisualization	31bf3856ad364e35	Interactive visualiztion for a timeline of events
Microsoft.Dynamics.Nav.Client.VideoPlayer	31bf3856ad364e35	Microsoft Dynamics VideoPlayer control add-in
Microsoft.Dynamics.Nav.Client.WebPageViewer	31bf3856ad364e35	Microsoft Web Page Viewer control add-in

2. (Optional) Set aspect ratio

```
CurrPage.WebPageViewer.InitializeIFrame(ratio);
```

Note - If the control is by itself then it will occupy all the space available

3. Navigate to online site or set HTML Content

```
CurrPage.WebPageViewer.Navigate(url,method,data);
```

```
CurrPage.WebPageViewer.SetContent(html)
```

WebPageViewer – Interact with the page

Listen to events from the page:

```
CurrPage.WebPageViewer.SubscribeToEvent(eventName,origin)
```

Invoke events on the page:

```
CurrPage.WebPageViewer.InvokeEvent(data)
```

Post message back to the page:

```
CurrPage.WebPageViewer.PostMessage(message,targetDomain,convertToJson)
```

Set links to open in a new window:

```
CurrPage.WebPageViewer.LinksOpenInNewWindow
```

We will show the busy dialog automatically if it takes too long time to load

WebPageViewer - Events

- [-] WebPageViewer::ControlAddInReady(callbackUrl : Text)
 - [-] WebPageViewer::DocumentReady()
 - [-] WebPageViewer::Callback(data : Text)
 - [-] WebPageViewer::Refresh(callbackUrl : Text)

WebPageViewer – Known Limitations

The content is hosted in an iframe, so browser security around iFrames will be enforced. For example:

- Websites can choose to not have their content displayed inside an iframe
- Limited scripting access to content inside/outside the iframe (it is subject to Same Origin Policy)

Kudos to the WebPageViewer Developer



Feedback from Last Time

Feedback from last time on Error Logging Pattern

Start a background session (STARTSESSION) for error login

Some of the partners are using it often to get a responsive UI



STARTSESSION is very expensive – equal to user logging in

2 users can be generating the load of 100

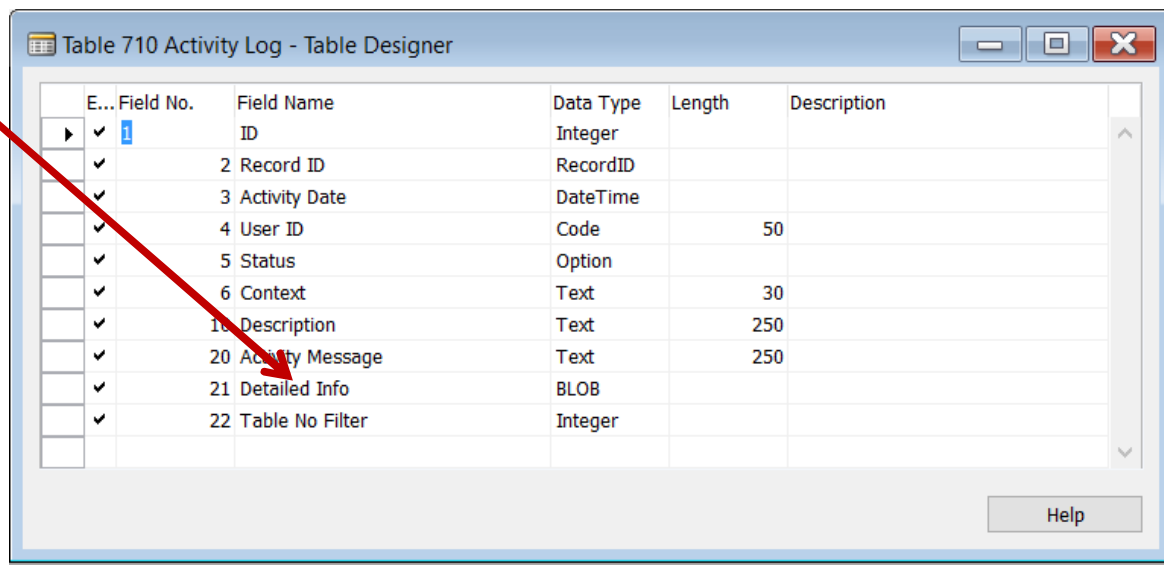
```
IF GLAccount.FINDSET THEN BEGIN
  REPEAT
    TempErrorMessage.LogIfEmpty (GLAccount, GLAccount.FIELDNO
(Name), TempErrorMessage."Message Type"::Error);

    XMLDOMManagement.AddElement (RootNode, 'Ctas', '', Namespace, Node
XMLDOMManagement.AddAttribute (Node, 'CodAgrup', GLAccount."SAT
Account Code");
    ...
    CASE GLAccount."Debit/Credit" OF
      GLAccount."Debit/Credit"::Debit:
        XMLDOMManagement.AddAttribute (Node, 'Natur', 'D');
      GLAccount."Debit/Credit"::Credit:
        XMLDOMManagement.AddAttribute (Node, 'Natur', 'A');
    ELSE
      TempErrorMessage.LogMessage (
        GLAccount, GLAccount.FIELDNO
("Debit/Credit"), TempErrorMessage."Message Type"::Error,
        STRSUBSTNO
(GLAccount.TvpeErr, GLAccount."Debit/Credit", GLAccount.RECORDID));
```

Feedback from Last Time

Implement large file support on the Activity Log

Done



	E...	Field No.	Field Name	Data Type	Length	Description
<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	ID	Integer		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	2	Record ID	RecordID		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	3	Activity Date	DateTime		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	4	User ID	Code	50	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	5	Status	Option		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	6	Context	Text	30	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	18	Description	Text	250	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	20	Activity Message	Text	250	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	21	Detailed Info	BLOB		
<input type="checkbox"/>	<input checked="" type="checkbox"/>	22	Table No Filter	Integer		

Help

Feedback from this time - Media Sets

Prefer media sets instead of images

It is cached – pages will not flicker. Blob gets loaded each time

You can access the Raw tables – but you shouldn't

Not all keywords are implemented yet

Not all controls client side are implemented yet

Summary

Hammer I Events

Observer Event Pattern
Discovery
Handled
Targeted Subscriber

Hammer II Extensions

Extension Blue Print
Be careful around upgrade
Referenced Setup Table
Record Sets

WebPageViewer

Try it out its cool



— Events

= Extensions

Hammer time

aka.ms/navpatterns
yammer

**NAV
TECH
DAYS
2016**

mibuso.com

QUESTIONS?

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

**NAV
TECH
DAYS
2016**

mibuso.com