



mibuso.com

# Give Business Central access to your data files

Gunnar Gestsson

When you are passionate about  
Microsoft Dynamics NAV/365 Business Central

# Me, myself and I



## GUNNAR ÞÓR GESTSSON

✉ [gunnar@navision.guru](mailto:gunnar@navision.guru)

🌐 <https://kappi.is>

🌐 <https://www.navision.guru>

## KAPPI EHF.

📍 Eyrartún 4, 550 Sauðárkrókur,  
Iceland | kt. 411203-2630

☎ +354 660 5565

✉ [kappi@kappi.is](mailto:kappi@kappi.is)



# Once upon a time

- File arrives
  - Automatic
  - Manual
- File is imported
  - Automatic with Job Queue
  - Automatic with DotNet on Windows Client
  - Manual
- File is processed with custom code

# File storage

- File Shares
  - Internal file servers
  - Various mounted file systems
- Local
  - My folder structure
  - I have control
- Business Central (NAV) is onPrem

# Today's Schedule

- Data Exchange Framework
- Job Queue Task Scheduler
- Incoming Documents
- File Services
- Dropbox, RushFiles
- Azure Blob, Azure Files
- Document Scanning



mibuso.com

# Data Exchange Framework

- Process Definition
  - Line Definition
  - Column Definition
  - Field Mapping
- Transformation Rules
- Extending

# Data Exchange Framework | Process Definition



←

DATA EXCHANGE DEFINITIONS | WORK DATE: 1/28/2021

🔍 Search

➕ New

✎ Edit List

🗑 Delete

✎ Edit

🔍 View

Import/Export

📄 Open in Excel

More options

🔍

☰

Code ↑	Name	Type	Data Handling Codeunit	Validation Codeunit	Reading/Writing Codeunit	Reading/Writing XMLport	Ext. Data Handling Codeunit	User Feedback Codeunit
→ <a href="#">ECB-EXCHA...</a> ⋮	European Central Bank Currenc...	Generic Import	0	0	1203	0	0	0
<a href="#">OCRCREDI...</a>	OCR - Credit Memo	Generic Import	1214	0	1203	0	0	0
<a href="#">OCRINVOICE</a>	OCR - Invoice	Generic Import	1214	0	1203	0	0	0
<a href="#">PEPPOLCRE...</a>	PEPPOL - Credit Memo	Generic Import	1214	0	1203	0	0	0
<a href="#">PEPPOLINV...</a>	PEPPOL - Invoice	Generic Import	1214	0	1203	0	0	0
<a href="#">SEPA CAMT</a>	SEPA CAMT	Bank Stateme...	0	0	1200	0	1240	0



# Data Exchange Framework | Process Definition

← DATA EXCHANGE DEFINITIONS | WORK DATE: 1/28/2021

Search + New Edit List Delete Edit View Import/Export Open in Excel More options

Code ↑	Name	Type	Data Handling Codeunit	Validation Codeunit	Reading/Writing Codeunit	Reading/Writing XMLport	Ext. Data Handling Codeunit	User Feedback Codeunit
→ <a href="#">ECB-EXCHA...</a>	European Central Bank Currenc...	Generic Import	0	0	1203	0	0	0
<a href="#">OCRCREDI...</a>	OCR - Credit Memo	Generic Import	1214	0	1203	0	0	0
<a href="#">OCRINVOICE</a>	OCR - Invoice	Generic Import	1214	0	1203	0	0	0
<a href="#">PEPPOLCRE...</a>	PEPPOL - Credit Memo	Generic Import	1214	0	1203	0	0	0
<a href="#">PEPPOLINV...</a>	PEPPOL - Invoice	Generic Import	1214	0	1203	0	0	0
<a href="#">SEPA CAMT</a>	SEPA CAMT	Bank Stateme...	0	0	1200	0	1240	0

- Confirmation/Authentication when writing data
- Confirmation/Notification when reading data

# Data Exchange Framework | Process Definition

← DATA EXCHANGE DEFINITIONS | WORK DATE: 1/28/2021

Search + New Edit List Delete Edit View Import/Export Open in Excel More options

Code ↑	Name	Type	Ext. Data Handling Codeunit	User Feedback Codeunit
→ ECB-EXCHA...	European Central Bank Currenc...	Generic Imp	0	0
OCRCREDI...	OCR - Credit Memo	Generic Imp	0	0
OCRINVOICE	OCR - Invoice	Generic Imp	0	0
PEPPOLCRE...	PEPPOL - Credit Memo	Generic Imp	0	0
PEPPOLINV...	PEPPOL - Invoice	Generic Imp	0	0
SEPA CAMT	SEPA CAMT	Bank Statem	1240	0

```
table 1220 "Data Exch."
{
    Caption = 'Data Exch.';
    Permissions = TableData "Data Exch." = i,
                  TableData "Data Exch. Field" = i;
    fields
    {
        field(1; "Entry No."; Integer)
        {
            AutoIncrement = true;
            Caption = 'Entry No.';
        }
        field(2; "File Name"; Text[250])
        {
            Caption = 'File Name';
        }
        field(3; "File Content"; BLOB)
        {
        }
    }
}
```

- Import/Download the file
- Export/Upload the file

# Data Exchange Framework | Process Definition

← DATA EXCHANGE DEFINITIONS | WORK DATE: 1/28/2021

Search + New Edit List Delete Edit View Import/Export Open in Excel More options

Code ↑	Name	Type	Data Handling Codeunit	Validation Codeunit	Reading/Writing Codeunit	Reading/Writing XMLport	Ext. Data Handling Codeunit	User Feedback Codeunit
→ ECB-EXCHA...	European Central Bank Currenc...	Generic Import	0	0	120			
OCRCREDI...	OCR - Credit Memo	Generic Import	1214	0	120			
OCRINVOICE	OCR - Invoice	Generic Import	1214	0	120			
PEPPOLCRE...	PEPPOL - Credit Memo	Generic Import	1214	0	120			
PEPPOLINV...	PEPPOL - Invoice	Generic Import	1214	0	120			
SEPA CAMT	SEPA CAMT	Bank Stateme...	0	0	120			

```
table 1221 "Data Exch. Field"
{
    Caption = 'Data Exch. Field';
    Permissions = TableData "Data Exch. Field" = rimd;

    fields
    {
        field(1; "Data Exch. No."; Integer)
        {
            Caption = 'Data Exch. No.';
            NotBlank = true;
            TableRelation = "Data Exch.";
        }
        field(2; "Line No."; Integer)
        {
            Caption = 'Line No.';
            NotBlank = true;
        }
        field(3; "Column No."; Integer)
        {
            Caption = 'Column No.';
            NotBlank = true;
        }
        field(4; Value; Text[250])
    }
}
```

- Read "File Content" the file into fields
- Write from fields into "File Content"

# Data Exchange Framework | Process Definition

← DATA EXCHANGE DEFINITIONS | WORK DATE: 1/28/2021



Search + New Edit List Delete Edit View Import/Export Open in Excel More options



Code ↑	Name	Type	Data Handling Codeunit	Validation Codeunit	Reading/Writing Codeunit	Reading/Writing XMLport	Ext. Data Handling Codeunit	User Feedback Codeunit
→ ECB-EXCHA...	European Central Bank Currenc...	Generic Import	0	0	1203	0	0	0
OCRCREDI...	OCR - Credit Memo	Generic Import	1214	0	1203	0	0	0

## Line Definitions

Manage



Line Type	Code ↑	Name	Column Count	Data Line Tag	Namespace
→ Detail	PEPPOLINV...	PEPPOL - Invoice Header	47	/Invoice	urn:oasis:na
Detail	PEPPOLINV...	PEPPOL - Invoice Lines	18	/Invoice/cac:InvoiceLine	

## Column Definitions

Manage



Column No. ↑	Name	Data Type	Data Format	Data Formatting Culture	Length
→ 1	ID	Text			
2	IssueDate	Text			

# Data Exchange Framework | Process Definition

← DATA EXCHANGE DEFINITIONS | WORK DATE: 1/28/2021

Search + New Edit List Delete Edit View Import/Export Open in Excel More options

Code ↑	Name	Type	Data Handling Codeunit	Validation Codeunit	Reading/Writing Codeunit	Reading/Writing XMLport	Ext. Data Handling Codeunit	User Feedback Codeunit
→ <a href="#">ECB-EXCHA...</a>	European Central Bank Currenc...	Generic Import	0	0	1203	0	0	0
<a href="#">OCRCREDI...</a>	OCR - Credit Memo	Generic Import	1214	0	1203	0	0	0
<a href="#">OCRINVOICE</a>	OCR - Invoice	Generic Import	1214	0	1203	0	0	0
<a href="#">PEPPOLCRE...</a>	PEPPOL - Credit Memo	Generic Import	1214	0	1203	0	0	0
<a href="#">PEPPOLINV...</a>	PEPPOL - Invoice	Generic Import	1214	0	1203	0	0	0
<a href="#">SEPA CAMT</a>	SEPA CAMT	Bank Stateme...	0	0	1200	0	1240	0

- Validate the "Data Exch. Field" data

# Data Exchange Framework | Process Definition

← DATA EXCHANGE DEFINITIONS | WORK DATE: 1/28/2021

Search + New Edit List Delete Edit View Import/Export Open in Excel

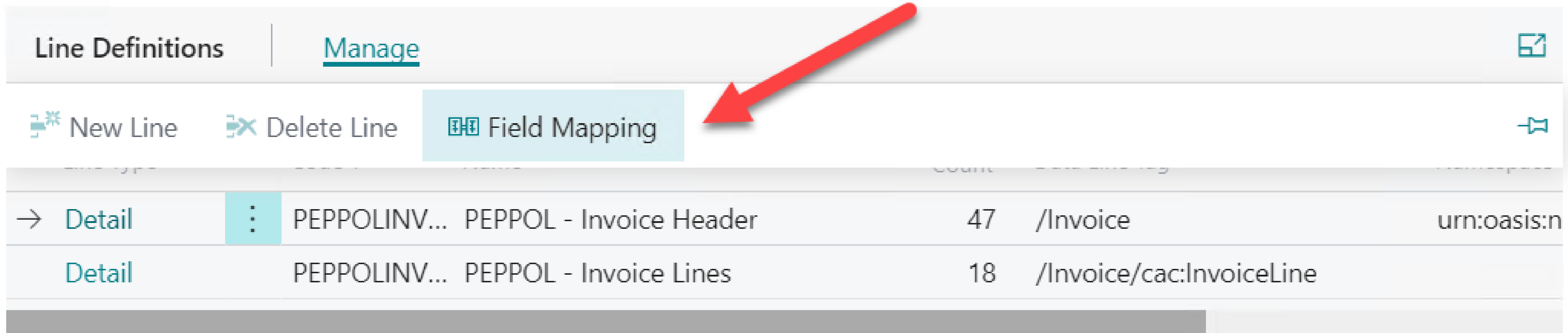
Code ↑	Name	Type	Data Handling Codeunit	Validation Codeunit	Reading/Writing Codeunit
→ ECB-EXCHA...	European Central Bank Currenc...	Generic Import	0	0	1
OCRCREDI...	OCR - Credit Memo	Generic Import	1214	0	1
OCRINVOICE	OCR - Invoice	Generic Import	1214	0	1
PEPPOLCRE...	PEPPOL - Credit Memo	Generic Import	1214	0	1
PEPPOLINV...	PEPPOL - Invoice	Generic Import	1214	0	1
SEPA CAMT	SEPA CAMT	Bank Stateme...	0	0	1

- Copy Field data to Intermediate Data Import

```
table 1214 "Intermediate Data Import"
{
    Caption = 'Intermediate Data Import';

    fields
    {
        field(1; ID; Integer)
        {
            AutoIncrement = true;
            Caption = 'ID';
        }
        field(2; "Data Exch. No."; Integer)
        {
            Caption = 'Data Exch. No.';
            NotBlank = true;
            TableRelation = "Data Exch.";
        }
        field(3; "Table ID"; Integer)
        {
            Caption = 'Table ID';
            NotBlank = true;
            TableRelation = AllObjWithCaption."Object ID" V
        }
        field(4; "Record No."; Integer)
        {
            Caption = 'Record No.';
        }
        field(5; "Field ID"; Integer)
        {
            Caption = 'Field ID';
            TableRelation = Field."No." WHERE(TableNo = FI
        }
        field(6; Value; Text[250])
    }
}
```

# Data Exchange Framework | Mapping Definition



The screenshot shows the 'Line Definitions' section with a 'Manage' link. Below this, there are three buttons: 'New Line', 'Delete Line', and 'Field Mapping'. The 'Field Mapping' button is highlighted with a light blue background and a red arrow pointing to it. Below the buttons, there is a table with two rows of data.

	Line Type	Table Name	Count	Path	URI
→ Detail	:	PEPPOLINV... PEPPOL - Invoice Header	47	/Invoice	urn:oasis:n
Detail		PEPPOLINV... PEPPOL - Invoice Lines	18	/Invoice/cac:InvoiceLine	

- Map "Data Exch. Field" to Table Data
- Map "Intermediate Data Import" to Table Data

# Data Exchange Framework | Mapping Definition



FIELD MAPPING | WORK DATE: 1/28/2021



✓ SAVED



ECB-EXCHANGE-RATES · CURREXCHANGE · 330

## General

Table ID	<input type="text" value="330"/>	Pre-Mapping Codeunit	<input type="text" value="0"/>
Use as Intermediate Table	<input type="checkbox"/>	Mapping Codeunit	<input type="text" value="1280"/>
Name	<input type="text" value="Mapping for Currency Exchange"/>	Post-Mapping Codeunit	<input type="text" value="0"/>

## Field Mapping

Manage



	Column No. ↑	Column Caption	Field ID ↑	Field Caption	Optio...	Transformation Rule	Overw... Value
→	90000	time	2	Starting Date	<input type="checkbox"/>		<input type="checkbox"/>
	110000	currency	1	Currency Code	<input type="checkbox"/>		<input type="checkbox"/>
	120000	rate	3	Exchange Rate Amount	<input type="checkbox"/>		<input type="checkbox"/>
	130000		6	Relational Exch. Rate Amount	<input type="checkbox"/>		<input type="checkbox"/>
					<input type="checkbox"/>		<input type="checkbox"/>

# Data Exchange Framework | Mapping Definition

```
repeat
  if DataExchFieldGroupByLineNo."Line No." <> CurrLineNo then begin
    CurrLineNo := DataExchFieldGroupByLineNo."Line No.";

    RecRef := RecRefTemplate.Duplicate;
    if (DataExchMapping."Data Exch. No. Field ID" <> 0) and (DataExchMapping."Data Exch. Line Field ID" <> 0) then begin
      SetFieldValue(RecRef, DataExchMapping."Data Exch. No. Field ID", DataExch."Entry No.");
      SetFieldValue(RecRef, DataExchMapping."Data Exch. Line Field ID", CurrLineNo);
    end;
    SetFieldValue(RecRef, LastKeyFieldId, CurrLineNo * 10000 + LineNoOffset);
    DataExchFieldMapping.FindSet;
    repeat
      DataExchField.SetRange("Line No.", CurrLineNo);
      DataExchField.SetRange("Column No.", DataExchFieldMapping."Column No.");
      if DataExchField.FindSet then
        repeat
          SetField(RecRef, DataExchFieldMapping, DataExchField, TempFieldIdsToNegate)
        until DataExchField.Next = 0
      else
        if not DataExchFieldMapping.Optional then
          Error(
            MissingValueErr, DataExch."File Name", GetType(DataExch."Data Exch. Def Code"),
            DataExch."Data Exch. Def Code", CurrLineNo, DataExchFieldMapping."Column No.");
    until DataExchFieldMapping.Next = 0;

    NegateAmounts(RecRef, TempFieldIdsToNegate);

    RecRef.Insert;
  end;
until DataExchFieldGroupByLineNo.Next = 0;
```

# Data Exchange Framework | Mapping Definition


← FIELD MAPPING | WORK DATE: 1/28/2021    ✓ SAVED  

## PEPPOLINVOICE · PEPPOLINVHEADER · 1214

### General

Table ID	<input type="text" value="1214"/>	Pre-Mapping Codeunit	<input type="text" value="1217"/>
Use as Intermediate Table	<input checked="" type="checkbox"/>	Mapping Codeunit	<input type="text" value="1218"/>
Name	<input type="text" value="PEPPOLSALESINVOICE to Purchase Invoice"/>	Post-Mapping Codeunit	<input type="text" value="0"/>

### Field Mapping

Manage 

Column No. ↑	Column Caption	Target Table ID	Table Caption	Target Field ID	Field Caption	Vali... Only	Transformation Rule
→ 1	ID	38	Purchase Header	68	Vendor Invoice No.	<input type="checkbox"/>	
2	IssueDate	38	Purchase Header	99	Document Date	<input type="checkbox"/>	
3	DocumentCurrencyCode	38	Purchase Header	32	Currency Code	<input type="checkbox"/>	
4	TaxCurrencyCode	38	Purchase Header	114	Tax Area Code	<input checked="" type="checkbox"/>	
5	OrderReferenceID	38	Purchase Header	66	Vendor Order No.	<input type="checkbox"/>	
6	SupplierEndpointGLNID	38	Purchase Header	2	Buy-from Vendor...	<input type="checkbox"/>	
7	SupplierEndpointVATID	23	Vendor	86	VAT Registration ...	<input type="checkbox"/>	
8	SupplierName	38	Purchase Header	79	Buy-from Vendor...	<input type="checkbox"/>	
11	CustomerEndpointIDGLN	79	Company Information	90	GLN	<input checked="" type="checkbox"/>	
12	CustomerPartyIdentificationIDG...	79	Company Information	90	GLN	<input checked="" type="checkbox"/>	
13	CustPartyTaxSchemeCompanyID	79	Company Information	19	VAT Registration ...	<input checked="" type="checkbox"/>	

# Data Exchange Framework | Mapping Definition

```
with IntermediateDataImport do begin
    SetRange("Data Exch. No.", DataExch."Entry No.");
    SetRange("Table ID", DATABASE::"Purchase Line");
    SetRange("Parent Record No.", ParentRecordNo);
    SetCurrentKey("Record No.");

    if not FindSet then
        exit;

    repeat
        if CurrRecordNo <> "Record No." then begin // new record
            if CurrRecordNo <> -1 then // if not start of loop then save current rec
                RecRef.Modify(true);

            CurrRecordNo := "Record No.";
            CreateNewPurchLine(PurchaseHeader, RecRef, DataExch."Entry No.", ParentRecordNo, CurrRecordNo);
        end;

        if not "Validate Only" then
            if not IsFieldProcessed(TempProcessedLineFldId, "Field ID") then
                if Value <> '' then
                    ProcessField(TempProcessedLineFldId, RecRef, "Field ID", Value);
    until Next = 0;

    // Save the last rec
    if CurrRecordNo <> -1 then
        RecRef.Modify(true);
end;
```

# Data Exchange Framework | Transformation

←

FIELD MAPPING | WORK DATE: 1/28/2021

+

✓ SAVED

PEPPOLINVOICE · PEPPOLINVHEADER · 1214

General

Table ID

1214

Pre-Mapping Codeunit

1217

Use as Intermediate Table

Mapping Codeunit

1218

Name

PEPPOLSALESINVOICE to Purchase Invoice

Post-Mapping Codeunit

0

Field Mapping

Manage

Column No. ↑	Column Caption	Target Table ID	Table Caption	Target Field ID	Field Caption	Vali... Only	Transformation
→ 1	ID	38	Purchase Header	68	Vendor Invoice No	<input type="checkbox"/>	
2	IssueDate	38	Purchase Header	99			
3	DocumentCurrencyCode	38	Purchase Header	32			
4	TaxCurrencyCode	38	Purchase Header	114			
5	OrderReferenceID	38	Purchase Header	66			
6	SupplierEndpointGLNID	38	Purchase Header	2			
7	SupplierEndpointVATID	23	Vendor	86			
8	SupplierName	38	Purchase Header	79			
11	CustomerEndpointIDGLN	79	Company Information	90			
12	CustomerPartyIdentificationIDG...	79	Company Information	90			
13	CustPartyTaxSchemeCompanyID	79	Company Information	19			

←

TRANSFORMATION RULES | WORK DATE: 1/28/2021

Search

+

New

Manage

Open in Excel

Code ↑	Description	Transforma... Type	Find Value	Replace '
→ IMERIC_ONLY	Alphanumeric Text Only	Remove Non-		
DELETE_NO...	Delete NOTPROVIDED value	Regular Exp...	NOTPROVIDED	
DK_DECIM...	Danish Decimal Format	Decimal For...		
FOURTH_T...	Fourth to Sixth Characters Text	Substring		
LOWERCASE	Lower Case Text	Lowercase		
TITLECASE	Title Case Text	Title Case		
TRIM	Trim Text	Trim		
UNIXTIMES...	Transforming UNIX timestamp ...	Custom		

# Data Exchange Framework | Extending



DATA EXCHANGE DEFINITIONS | WORK DATE: 1/28/2021



Search	+ New	Edit List	Delete	Edit	View	Import/Export	Open in Excel	More options		
Code ↑	Name	Type	Data Handling Codeunit	Validation Codeunit	Reading/Writing Codeunit	Reading/Writing XMLport	Ext. Data Handling Codeunit	User Feedback Codeunit		
<a href="#">ECB-EXCHA...</a>	European Central Bank Currenc...	Generic Import	0	0	1203	0	0	0		
<a href="#">OCRCREDI...</a>	OCR - Credit Memo	Generic Import	1214	0	1203	0	0	0		
<a href="#">OCRINVOICE</a>	OCR - Invoice	Generic Import	1214	0	1203	0	0	0		
<a href="#">PEPPOLCRE...</a>	PEPPOL - Credit Memo	Generic Import	1214	0	1203	0	0	0		
<a href="#">PEPPOLINV...</a>	PEPPOL - Invoice	Generic Import	1214	0	1203	0	0	0		
→ <a href="#">SEPA CAMT</a>	: SEPA CAMT	Bank Stateme...	0	0	1200	0	1240	0		

# Data Exchange Framework | Extending

```
codeunit 1240 "Read Data Exch. from File"
{
    TableNo = "Data Exch.";

    trigger OnRun()
    var
        TempBlob: Codeunit "Temp Blob";
        FileMgt: Codeunit "File Management";
        RecordRef: RecordRef;
    begin
        OnBeforeFileImport(TempBlob, "File Name");

        if not TempBlob.HasValue then
            "File Name" := CopyStr(
                FileMgt.BLOBImportWithFilter(TempBlob, ImportBankStmtTxt, ' ', FileFilterTxt, FileFilterExtensionTxt), 1, 250);

        if "File Name" <> '' then begin
            RecordRef.GetTable(Rec);
            TempBlob.ToRecordRef(RecordRef, FieldNo("File Content"));
            RecordRef.SetTable(Rec);
        end;
    end;

    var
        ImportBankStmtTxt: Label 'Select a file to import';
        FileFilterTxt: Label 'All Files(*.*)|*.xml|XML Files(*.xml)|*.xml|Text Files(*.txt;*.csv;*.asc)|*.txt;*.csv;*.asc;*.nda';
        FileFilterExtensionTxt: Label 'txt,csv,asc,xml,nda', Locked = true;

    [IntegrationEvent(false, false)]
    local procedure OnBeforeFileImport(var TempBlob: Codeunit "Temp Blob"; var FileName: Text)
    begin
    end;
}
```

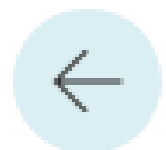
# Data Exchange Framework | Extending

```
codeunit 50135 "O4N Blob Content"
{
    EventSubscriberInstance = Manual;
    1 reference
    procedure Set(var InStr: InStream; FileName: Text)
    var
        OutStr: OutStream;
    begin
        Clear(BlobStorage);
        BlobStorage.CreateOutStream(OutStr);
        CopyStream(OutStr, InStr);
        FileNameStorage := FileName;
    end;

    [EventSubscriber(ObjectType::Codeunit, Codeunit::"Read Data Exch. from File", 'OnBeforeFileImport', '', false, false)]
    0 references
    local procedure SetBlob(var TempBlob: Codeunit "Temp Blob"; var FileName: Text)
    var
        InStr: InStream;
        OutStr: OutStream;
    begin
        if not BlobStorage.HasValue() then exit;
        BlobStorage.CreateInStream(InStr);
        TempBlob.CreateOutStream(OutStr);
        CopyStream(OutStr, InStr);
        FileName := FileNameStorage;
    end;

    var
        4 references
        BlobStorage: Codeunit "Temp Blob";
        2 references
        FileNameStorage: Text;
}
```

# Data Exchange Framework | Extending



BANK EXPORT/IMPORT SETUP | WORK DATE: 1/28/2021

Search + New Edit List Delete Open in Excel					
Code ↑		Name	Direction	Processing Codeunit ID	Processing Codeunit Name
→ SEPA CAMT	:	SEPA CAMT	Import	1270	Exp. Launcher Gen. Jnl.
SEPACT		SEPA Credit Transfer	Export	1220	SEPA CT-Export File
SEPADD		SEPA Direct Debit	Export	1230	SEPA DD-Export File

# Data Exchange Framework | Extending

```
codeunit 50140 "O4N Exp. Launcher Gen. Jnl."  
{  
    TableNo = "Gen. Journal Line";  
  
    trigger OnRun()  
    var  
        BlobContent: Codeunit "O4N Blob Content";  
        TempBlob: Codeunit "Temp Blob";  
        InStr: InStream;  
        FileName: Text;  
    begin  
        TempBlob.CreateInStream(InStr);  
        FileName := GetDownloadUrl();  
        DownloadBlob(FileName, InStr);  
        BlobContent.Set(InStr, FileName);  
        BindSubscription(BlobContent);  
        Codeunit.Run(Codeunit::"Exp. Launcher Gen. Jnl.", Rec);  
        UnbindSubscription(BlobContent);  
    end;  
  
    1 reference  
    local procedure GetDownloadUrl() Url: Text;  
    begin  
        Url := 'https://365links.blob.core.windows.net/azureblobdemo/ANNA_20190215_0003-thumb.png';  
    end;  
  
    1 reference  
    procedure DownloadBlob(Url: Text; var InStr: InStream)  
    var  
        WebResponse: HttpResponseMessage;  
        WebClient: HttpClient;  
    begin  
        WebClient.Get(url, WebResponse);  
        if not WebResponse.IsSuccessStatusCode() then  
            error(WebResponse.ReasonPhrase());  
        WebResponse.Content().ReadAs(InStr);  
    end;  
}
```



mibuso.com

# Job Queue Task Scheduler

- Automates tasks
- Free up the User Interface for interactions
- Asynchronous APIs
- Navision Application Server (NAS)
  - Session running a timer that reads the database to search for tasks on time to start.
  - New session created for every task
- Task Scheduler ( $\geq 2017$ )
  - Creates a scheduled task via the service tier
  - Service tier creates new session when the schedule starts
  - Service configuration for maximum concurrent tasks

# Job Queue Task Scheduler | Configuration

Task Scheduler

Enable Task Scheduler:

☒

Maximum Concurrent Running Tasks:

System Task End Time:

System Task Start Time:

# Job Queue Task Scheduler

- Problems
  - Error handling
  - Tasks not starting
  - Tasks not completing
- Workarounds
  - Put Job Queue details on the Role Center
  - Create dedicated handler task
  - Create dedicated error monitor task

# Job Queue Task Scheduler | Role Center



by Advania

Business Central

Kappi ehf.

Finance

Cash Management

Sales

Purchasing

Approvals

Self-Service

Setup & Extensions

Payroll & Employees

Customers

Vendors

Items

Bank Accounts

Chart of Accounts

Subscription Invoices

Ófrágengin

Customer Application

Vendor Application

More

ONGOING SALES

SALES QUOTES

0

>

SALES ORDERS

0

>

SALES INVOICES

0

>

ONGOING PURCHASES

PURCHASE ORDERS

0

>

ONGOING PUR... INVOICES

0

>

PURCH. INVOI... NEXT WEEK

2

>

PAYMENTS

UNPROCESSED PAYMENTS

0

>

AVERAGE COL... DAYS

16,2

OUTSTANDING... INVOICES

9

>

CAMERA

Create Incomin... from Camera

INCOMING DOCUMENTS

MY INCOMING DOCUMENTS

12

>

INC. DOC. AWA... VERIFICATION

0

>

MY USER TASKS

PENDING USER TASKS

0

>

JOB QUEUE

JOB QUEUE E... WITH ERROR

1

>

START

Sales Quote

Sales Order

Sales Invoice

Purchase Invoice

Sales Return Order

PRODUCT VIDEOS

Product Videos

Insights

Business Assistance

Self-Service

PENDING TIME SHEETS

PENDING TIME SHEETS

# Job Queue Task Scheduler | Handler

```
codeunit 50122 "O4N Import Job Queue Handler"
{
    TableNo = "Job Queue Entry";

    trigger OnRun()
    var
        ImportSetup: Record "O4N Import Setup";
    begin
        TestField("Record ID to Process");
        ImportSetup.Get("Record ID to Process");
        ScheduleImport(ImportSetup);
        MonitorErrorTasks(ImportSetup);
    end;

    1 reference
    procedure ScheduleImport(ImportSetup: Record "O4N Import Setup")
    var
        JobQueueMgt: Codeunit "O4N Import Scheduler Mgt";
    begin
        JobQueueMgt.EnqueueTask(Codeunit::"O4N Import Job Queue Task", ImportSetup.RecordId(), 30, ImportSetup."Provider Description");
    end;

    1 reference
    procedure MonitorErrorTasks(ImportSetup: Record "O4N Import Setup")
    var
        JobQueueMgt: Codeunit "O4N Import Scheduler Mgt";
    begin
        JobQueueMgt.EnqueueTask(Codeunit::"O4N Import Job Queue Monitor", ImportSetup.RecordId(), 0, ImportSetup."Provider Description");
    end;
}
```

# Job Queue Task Scheduler | Monitor

```
codeunit 50121 "O4N Import Job Queue Monitor"
{
    TableNo = "Job Queue Entry";

    You, 4 days ago • Server Import on it's way

    trigger OnRun()
    begin
        RestartErrorTasks();
    end;

    1 reference
    local procedure RestartErrorTasks()
    var
        JobQueueEntry: Record "Job Queue Entry";
        ActivityLog: Record "Activity Log";
    begin
        with JobQueueEntry do begin
            SetRange("Object Type to Run", "Object Type to Run"::Codeunit);
            SetFilter("Object ID to Run", '%1|%2|%3|%4',
                Codeunit::"O4N Import Job Queue Handler",
                Codeunit::"O4N Import Job Queue Task",
                Codeunit::"O4N Import To DataExch. Job",
                Codeunit::"O4N Import Process DataExch.");
            SetRange(Status, Status::Error);
            SetAutoCalcFields("Object Caption to Run");
            if FindSet() then
                repeat
                    SetImportEntryError(JobQueueEntry);
                    ActivityLog.LogActivity("Record ID to Process", ActivityLog.Status::Failed, RestartLbl, "Object Caption to Run", "Error Message");
                    SetStatus(Status::Ready);
                until Next() = 0;
            end;
        end;
    end;
}
```

# Job Queue Task Scheduler | Start Tasks

1 reference

```
local procedure JobQueueEntryExists(CodeunitId: Integer; RelatedRecordId: RecordID): Boolean
var
    JobQueueEntry: Record "Job Queue Entry";
begin
    JobQueueEntry.SetRange("Object Type to Run", JobQueueEntry."Object Type to Run"::Codeunit);
    JobQueueEntry.SetRange("Object ID to Run", CodeunitId);
    JobQueueEntry.SetRange("Record ID to Process", RelatedRecordId);
    exit(not JobQueueEntry.IsEmpty());
end;
```



mibuso.com

# Incoming Documents

- Entry point for data that is to create Business Central document
  - Purchase
  - Sales
  - G/L Journal
- Attachments
  - One main attachment, used for the Data Exchange process
  - Binary attachment can be sent to OCR that results in the main attachment (Xml)
  - Supplemental attachments
- Navigation
  - Linked to posted documents and entries

# Incoming Documents | Business Manager



by Advania

Business Central

- Kappi ehf.
- Finance
- Cash Management
- Sales
- Purchasing
- Approvals
- Self-Service
- Setup & Extensions
- Payroll & Employees
- Customers
- Vendors
- Items
- Bank Accounts
- Chart of Accounts
- Subscription Invoices
- Ófrágengin
- Customer Application
- Vendor Application
- More

ONGOING SALES

SALES QUOTES

0

>

SALES ORDERS

0

>

SALES INVOICES

0

>

ONGOING PURCHASES

PURCHASE ORDERS

0

>

ONGOING PUR... INVOICES

0

>

PURCH. INVOI... NEXT WEEK

2

>

PAYMENTS

UNPROCESSED PAYMENTS

0

>

AVERAGE COL... DAYS

16,2

OUTSTANDING... INVOICES

9

>

CAMERA

Create Incomin... from Camera

INCOMING DOCUMENTS

MY INCOMING DOCUMENTS

12

>

INC. DOC. AWA... VERIFICATION

0

>

MY USER TASKS

PENDING USER TASKS

0

>

JOB QUEUE

JOB QUEUE E... WITH ERROR

1

>

START

+

Sales Quote

+

Sales Order

+

Sales Invoice

+

Purchase Invoice

+

Sales Return Order

PRODUCT VIDEOS

Product Videos

Insights

Business Assistance

Self-Service

PENDING TIME SHEETS

# Incoming Documents | Creation

- Manually create an incoming document

The screenshot displays the 'MY INCOMING DOCUMENTS | WORK DATE: 1/28/2021' interface. A red arrow points to the 'Create from File' button in the 'New' dropdown menu. The interface includes a search bar, a 'New' button, and a list of actions: Manage, Process, Release, Status, Show, Open in Excel, and More options. A table titled 'Incoming Document Files' shows a document named 'London Postmaster Invoice W1' with a PDF type.

Name	Type
London Postmaster Invoice W1	PDF

- Automatically create an incoming document
  - Document Exchange Service (TradeShift)
  - Custom Data Exchange Flow
  - Document Scanning/From Camera

# Incoming Documents | Processing

	Code ↑	Description	Data Exch. Def. Code	Entity Type
	OCRCREDITMEMO	OCR - Credit Memo	OCRCREDITMEMO	Credit Memo
	OCRINVOICE	OCR - Invoice	OCRINVOICE	Invoice
	PEPPOLCREDITMEMO	PEPPOL - Credit Memo	PEPPOLCREDITMEMO	Credit Memo
	PEPPOLINVOICE	PEPPOL - Invoice	PEPPOLINVOICE	Invoice
→	* <input type="text"/>			Invoice

# Incoming Documents | Processing

- Selected by counting the no. of hits

```
codeunit 1215 "Data Exch. Type Selector"
{
    Permissions = TableData "Data Exch." = imd;
    TableNo = "Incoming Document Attachment";

    trigger OnRun()
    var
        BestDataExchCode: Code[20];
    begin
        if Type <> Type::XML then
            Error(InvalidTypeErr);

        CheckContentHasValue(Rec);

        BestDataExchCode := FindDataExchType(Rec);

        SetResult("Incoming Document Entry No.", BestDataExchCode);
    end;

    var
        InvalidTypeErr: Label 'The attachment is not an XML document.';
        AttachmentEmptyErr: Label 'The attachment does not contain any data.';

    local procedure FindDataExchType(IncomingDocumentAttachment: Record "Incoming Document Attachment"): Code[20]
```

# Incoming Documents | Processing

- Your own document types
  - Fixed Format Text Files (txt)
  - Variable Format Text Files (csv)
  - Excel Files (xlsx)
  - Xml Files (xml)
  - Json Files (json)
- Bring the files into Incoming Documents in Xml format
- Process with Data Exchange to create the target document type

# Incoming Documents | Processing

- Processing the best fit

← MY INCOMING DOCUMENTS | WORK DATE: 1/28/2021

Search New Manage Process Release Status Show Open in Excel Actions Navigate Fewer options

New Document ▾ Process ▾ Release ▾ Status ▾ Request Approval ▾ Incoming Documents ▾ Set View ▾

Create Document  
Create Journal Line  
Create Manually  
Attach File  
Map Text to Account

Incoming Document Files ▾

Name	Type
London Postmaster Invoice ...	PDF



mibuso.com

# File Services

- Make your files accessible by Business Central
- File Service needs to have an API
- Examples
  - Dropbox
  - RushFiles
  - OneDrive for Business
  - Azure Blob
  - Azure Files

# File Services | Dropbox

- Creates a folder on any of your local drives
- Home
  - <https://www.dropbox.com/>
- Download
  - <https://www.dropbox.com/downloading>
- API information
  - <https://www.dropbox.com/developers/documentation/http/overview>
- App Console
  - [https://www.dropbox.com/developers/apps?\\_tk=pilot\\_lp&\\_ad=topbar4&\\_camp=myapps](https://www.dropbox.com/developers/apps?_tk=pilot_lp&_ad=topbar4&_camp=myapps)
- Authentication
  - Authentication Token(s) created for every App

# File Services | RushFiles

- Creates a new remote drive
- Home
  - <https://rushfiles.net/>
- Download
  - <https://rushfiles.com/download-client-applications>
- API information
  - <https://developers.rushfiles.com>
- Management
  - <https://rushfiles.net/reseller/Default.aspx>
- Authentication
  - Reseller ApiKey, Company ApiKey, Device DomainToken

# File Services | OneDrive

- Creates a folder on any of your local drives
- Home
  - [https://<tenant>-my.sharepoint.com/personal/gunnar\\_kappi\\_is/\\_layouts/15/onedrive.aspx](https://<tenant>-my.sharepoint.com/personal/gunnar_kappi_is/_layouts/15/onedrive.aspx)
- Download
  - <https://onedrive.live.com/about/en-us/download/>
- API information
  - <https://docs.microsoft.com/en-us/graph/api/overview?view=graph-rest-1.0>
- Playground
  - <https://developer.microsoft.com/en-us/graph/graph-explorer/preview>
- Authentication
  - Azure AD OAuth

# File Services | Azure Blob

- Needs 3<sup>rd</sup> party app or Powershell to explore/sync the storage
- Home
  - <https://portal.azure.com/>
- Download
  - <https://azure.microsoft.com/en-us/features/storage-explorer/>,  
<https://www.goodsync.com/download/GoodSync-v10-WindowsServer-Setup.exe>
- API information
  - <https://docs.microsoft.com/en-us/rest/api/storageservices/blob-service-rest-api>
- Management
  - Azure Storage Account
- Authentication
  - Access Key

# File Services | Azure Files

- Syncs a folder on any of your local drives
- Home
  - <https://portal.azure.com/>
- Download
  - <https://docs.microsoft.com/en-us/azure/storage/files/storage-sync-files-deployment-guide?tabs=azure-portal>
- API information
  - <https://docs.microsoft.com/en-us/rest/api/storageservices/file-service-rest-api>
- Management
  - Azure Storage Account
- Authentication
  - Access Key

# File Services | Azure Files

- Creates a new remote drive with PowerShell

```
$resourceGroupName = "<your-resource-group-name>"
$storageAccountName = "<your-storage-account-name>"
$fileShareName = "<your-file-share-name>"

$storageAccount = Get-AzStorageAccount -ResourceGroupName $resourceGroupName -Name
$storageAccountName
$storageAccountKeys = Get-AzStorageAccountKey -ResourceGroupName $resourceGroupName -Name
$storageAccountName
$fileShare = Get-AzStorageShare -Context $storageAccount.Context | Where-Object {
    $_.Name -eq $fileShareName -and $_.IsSnapshot -eq $false}
if ($fileShare -eq $null) {
    throw [System.Exception]::new("Azure file share not found")}
$password = ConvertTo-SecureString -String $storageAccountKeys[0].Value -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential -ArgumentList
"AZURE\$($storageAccount.StorageAccountName)", $password
New-PSDrive -Name <desired-drive-letter> -PSProvider FileSystem -Root
"\$($fileShare.StorageUri.PrimaryUri.Host)\$($fileShare.Name)" -Credential $credential -Persist
```

# File Services | Demo

## DEMO



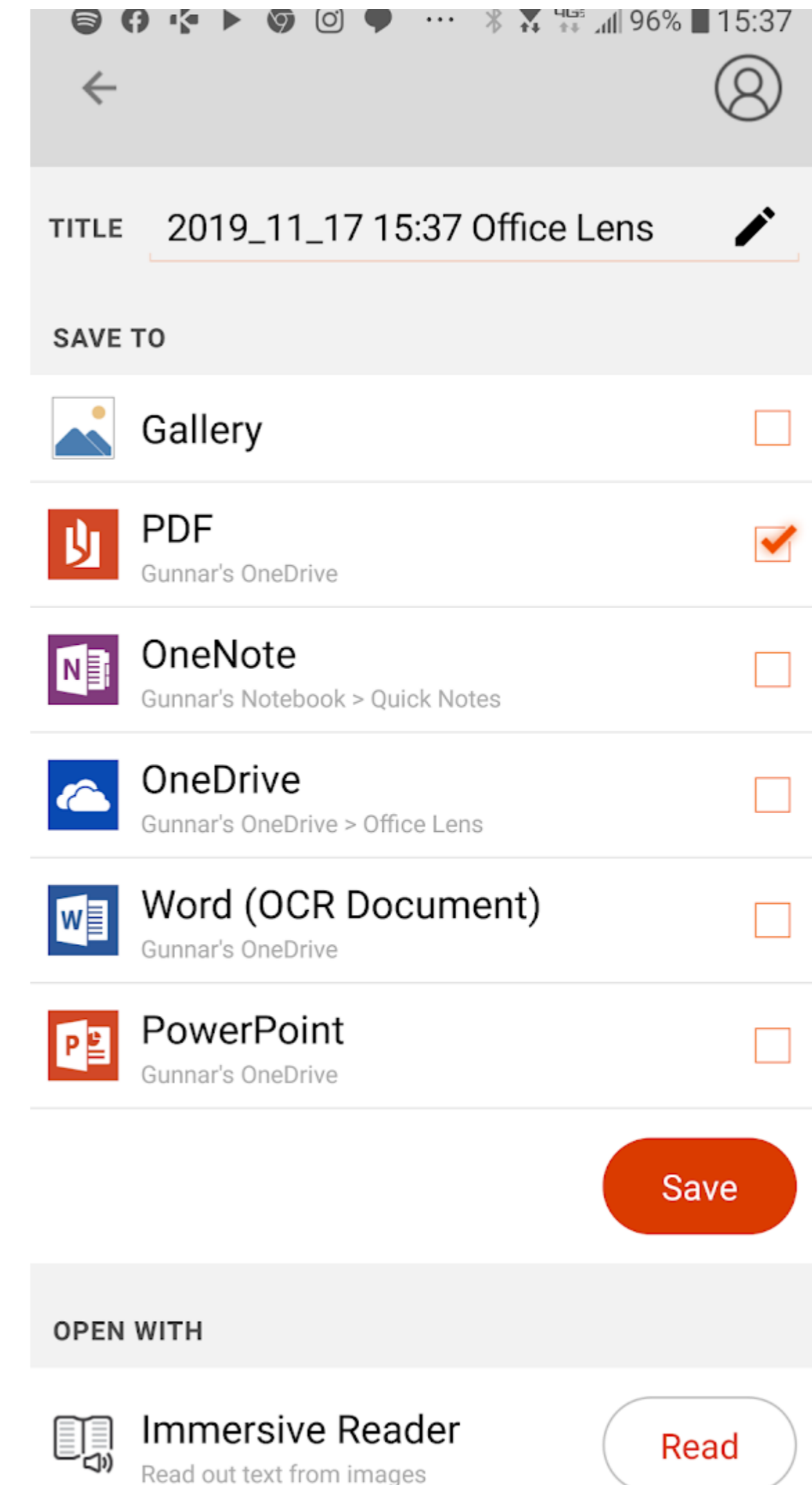
mibuso.com

# Document Scanning

- Having File Service Sync enables multiple document scanning options

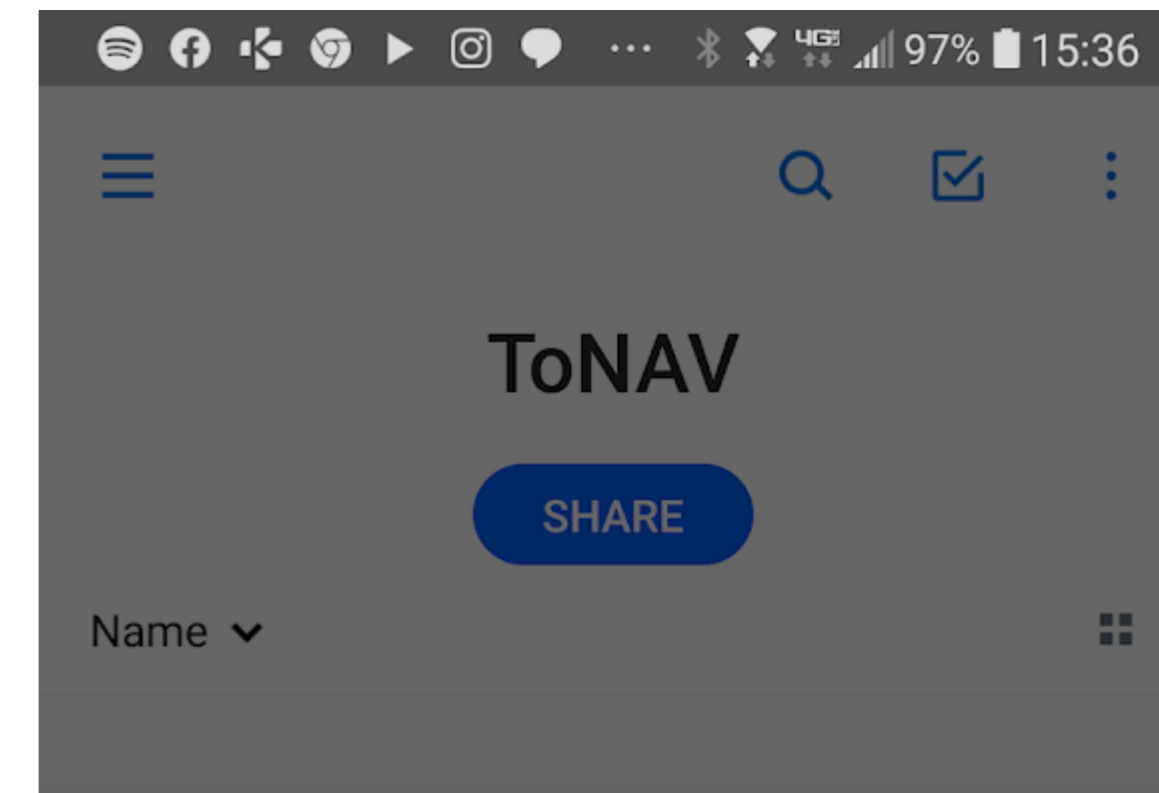
# Document Scanning | OneDrive

- Office Lens on you smartphone










# Document Scanning | Dropbox

- Dropbox on you smartphone



Add to ToNAV

-  Scan document
-  Upload photos or videos
-  Upload files
-  Create new folder
-  Create new Paper doc
-  Create new text file
-  Take a photo

# Document Scanning | In the Office

- Configure your scanner to batch scan to a folder
  - Share the folder with a File Service
  - User Business Central Job Queue to import to Incoming Documents
- Configure your scanner to scan to Email
  - Use Microsoft Flow to save attachment to a File Service
- Email attachments
  - Save your attachments directly to your File Service folder
  - Print to PDF and save in your File Service Folder

# Document Scanning | Importing

- Attachment sent to OCR if configured

```
local procedure NewIncomingDocument(var DataExch: Record "Data Exch.")
var
    IncomingDocument: Record "Incoming Document";
    SendIncomingDocumenttoOCR: Codeunit "Send Incoming Document to OCR";
begin
    GetImportEntry(DataExch);
    Window.Open(StrSubstNo(CreatingIncommingDocTxt, ImportEntry."File Path"));

    with IncomingDocument do begin
        CreateIncomingDocument(FileManagement.GetFileNameWithoutExtension(DataExch."File Name"), '');
        NewIncomingDocumentAttachment(DataExch, IncomingDocument);
        DataExch."Incoming Entry No." := "Entry No.";
        DataExch.Modify();
        ImportEntry."Processed To Record Id" := IncomingDocument.RecordId();
        ImportEntry.Modify();
    end;
    Window.Close();

    if IncomingDocument."OCR Service Doc. Template Code" <> '' then begin
        SendIncomingDocumenttoOCR.SetShowMessages(false);
        SendIncomingDocumenttoOCR.SendToJobQueue(IncomingDocument);
    end;
end;

1 reference
local procedure NewIncomingDocumentAttachment(DataExch: Record "Data Exch."; IncomingDocument: Record "Incoming Document")
```



mibuso.com

# Business Central Extension | Demo

DEMO

<https://dev.azure.com/objects4nav/TechDays>



mibuso.com

# Q&A

## Any Questions?

*Thank  
You!*