Send Files from D365F&O [ER Module] to Azure Blob Storage

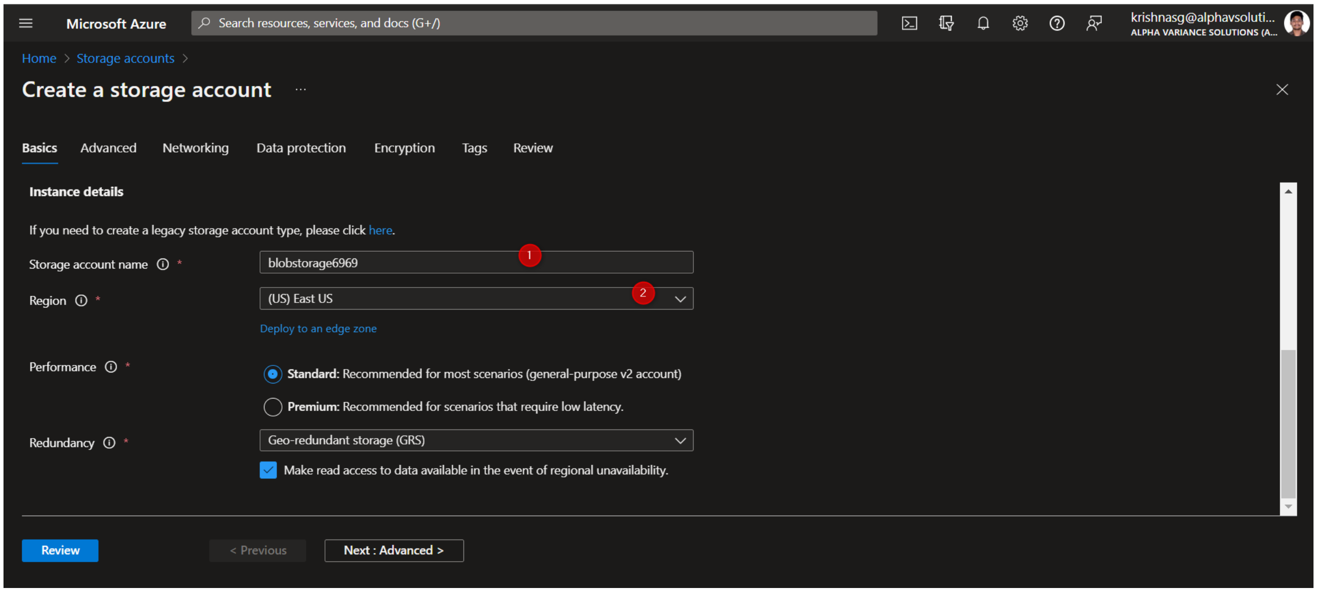
Confidentiality Notice:
This document is an internal working document of Alpha Variance Solutions, LLC. and the Client. It contains trade secrets and other proprietary information, which are the confidential property of Alpha Variance Solutions, LLC. and the Client. Neither this

This blog describes on how to upload files generated from D365FO [ER Module] to Azure Blob Storage container.

📋 Instructions

1. Create an Azure Storage Account

1. Login to portal.azure.com and sign into the preferred directory.
2. Under All Services search for **Storage Accounts** and open it.
3. Click on Create New and fill in the required details. [Make sure when creating the storage account, it resides in the particular resource group and the availability zone] as shown in the below picture.



Create a storage account

Instance details

If you need to create a legacy storage account type, please click here.

Storage account name ⓘ * blobstorage6969

Region ⓘ * (US) East US

Performance ⓘ *

● **Standard:** Recommended for most scenarios (general-purpose v2 account)

○ **Premium:** Recommended for scenarios that require low latency.

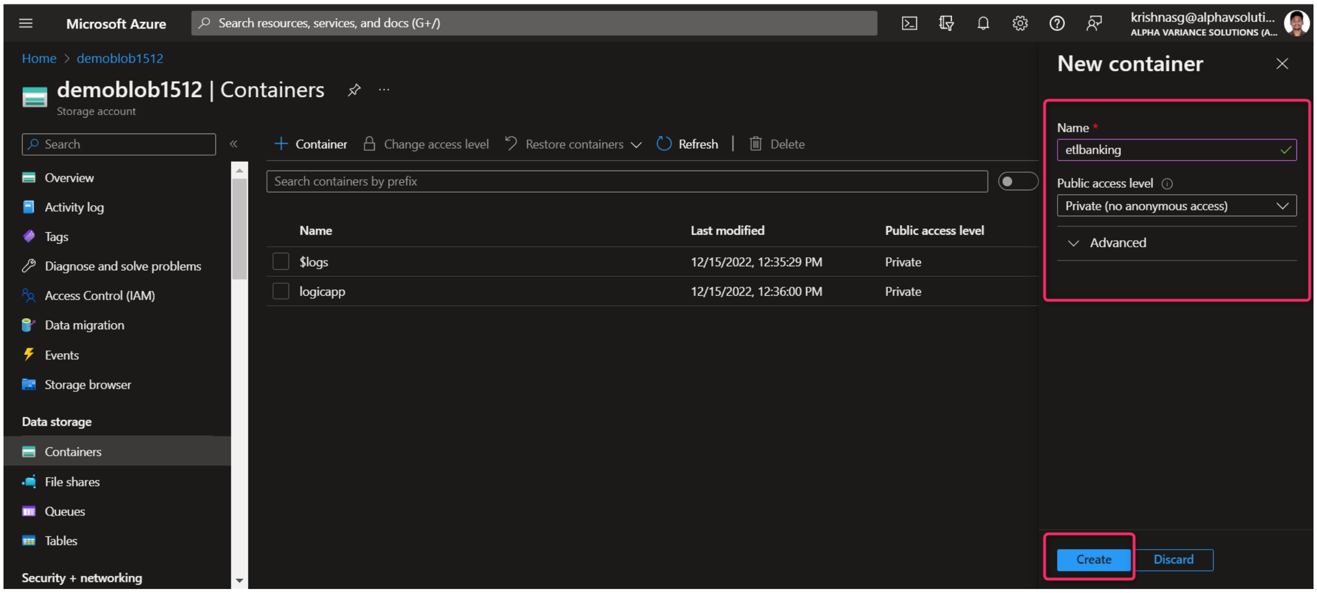
Redundancy ⓘ * Geo-redundant storage (GRS)

☑ Make read access to data available in the event of regional unavailability.

4. For all other settings/details choose the default ones and proceed with the creation.
5. Once the Storage account is created you can view it under storage accounts list.

Create a Container

1. Open the Storage Account which you have created.
2. Click on **Containers** > **New Container** as shown in the below picture.



Name	Last modified	Public access level
$logs	12/15/2022, 12:35:29 PM	Private
logicapp	12/15/2022, 12:36:00 PM	Private

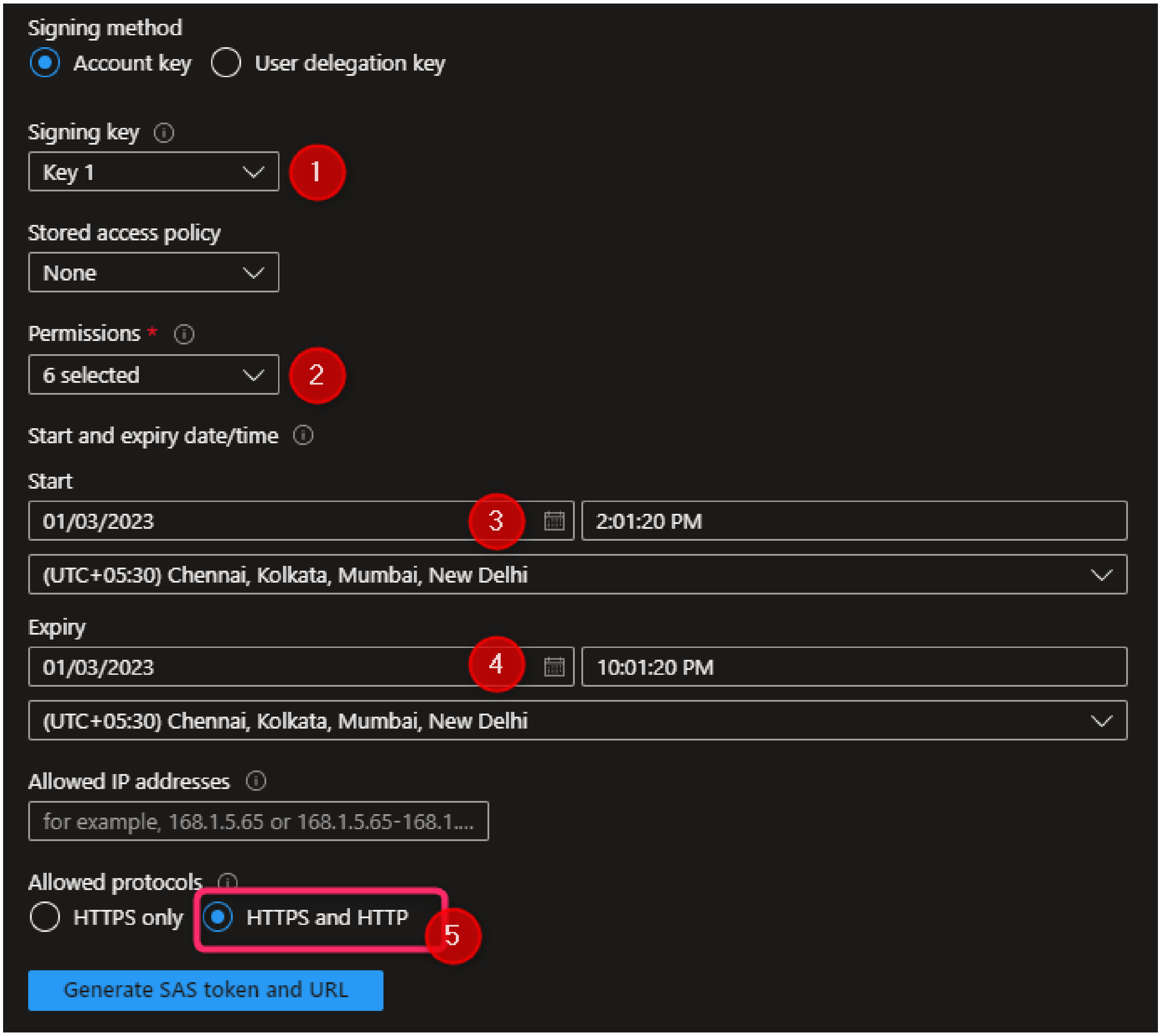
Name * etlbanking

Public access level: Private (no anonymous access)

3. Once the creation is done you can view the same under the containers list.

Generate SAS Token/Key

1. Open the container for which you want to generate the SAS Key/Token.
2. Under **Settings** click on **Shared Access Token**.
3. Create a SAS Token as shown in the below picture.



Signing method

● Account key ○ User delegation key

Signing key ⓘ

Key 1

Stored access policy

None

Permissions * ⓘ

6 selected

Start and expiry date/time ⓘ

Start

01/03/2023 2:01:20 PM

(UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi

Expiry

01/03/2023 10:01:20 PM

(UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi

Allowed IP addresses ⓘ

for example, 168.1.5.65 or 168.1.5.65-168.1....

Allowed protocols ⓘ

○ HTTPS only ● HTTPS and HTTP

4. Once Generated you will get an option to copy the same.

> Once SAS Key is generated you can view it only once. So kindly backup it up for future use.
>
> Make sure you put expiry date accordingly.

2. Write a logic in C sharp Library to Communicate with external sources.

- Create a C # Library with Required NuGet Dependencies.
- WindowsAzure.Storage (I have installed v9.3.3).
- Microsoft.IdentityModel.Clients.ActiveDirectory (I have installed v4.4.2).

```
 using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using System.Net;

namespace STE_AzureBlobStorageLibrary
{
    public class STE_AzureBlobStorageHelperLibrary
    {
        public string srcPath { get; set; }
        public string destinationPath { get; set; }
        public string storageAccountName { get; set; }
        public string containerName { get; set; }
        public string storageEndPoint { get; set; }
        public string blobName { get; set; }



    public string ReasonMesage { get; set; }

    public string mesg { get; set; }




    HttpClient httpClient = new HttpClient();

    static void Main()
    {

    }

    public Boolean saveFileInBlob(string _StorageAccountName, string 
_Container, string _SASKey = "", string _BlobName = "", System.IO.
Stream _stream = null)
    {

        string storageAddress = string.Format("<https://{0}.blob.core.
windows.net/",> _StorageAccountName);

        //preparing URI to drop the file to blob
        Uri containerUrl = new Uri(storageAddress + $"{_Container}/
{_BlobName}" + "?" + _SASKey);
```

```
        // which returns complete URI which is prepared in previous step
        string sasUri = containerUrl.AbsoluteUri;

        //x-ms-blob-type: <BlockBlob> - Returns the blob's type.
        httpClient.DefaultRequestHeaders.Add("x-ms-blob-type",
"BlockBlob");

        // Drops the specified Uri as an asynchronous operation to blob
        HttpResponseMessage response = httpClient.PutAsync(sasUri, new
StreamContent(_stream)).GetAwaiter().GetResult();
        if(response.IsSuccessStatusCode == false)
        {
            ReasonMesage = response.ReasonPhrase;
        bool mess = response.IsSuccessStatusCode;
        Console.WriteLine (mess.ToString());
           mesg = mess.ToString();
        }
        if (response.IsSuccessStatusCode)
            return true;
        else
            return false;


    }


}
```

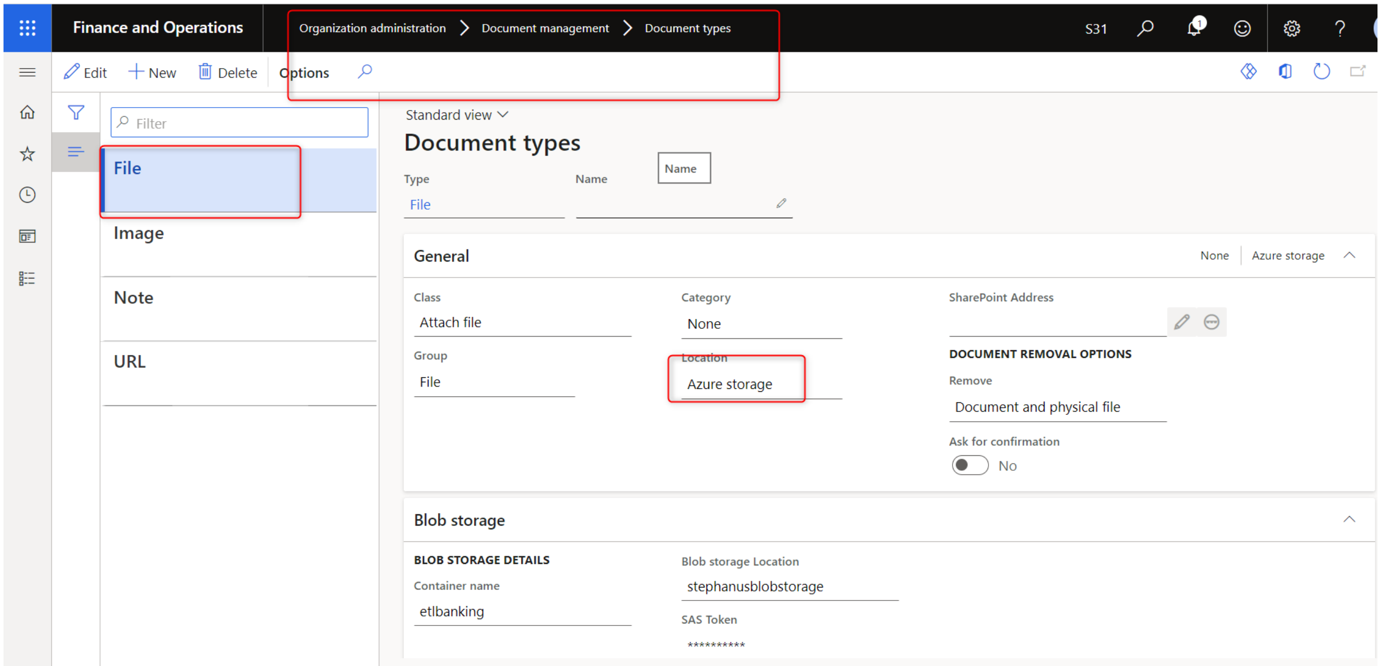
}

This class we have to call in X++ to send the files

3.Configure the File Type and Destination settings in F&O

Document Type Configuration:

1. Navigate to **Organization Admin** > **Document Management** > **Document Types**
2. Create a New or you can use an existing document type.
3. Configure the same by referring the below picture.



Document types

Type: File

Name:

General

Class: Attach file

Category: None

SharePoint Address:

Group: File

Location: Azure storage

DOCUMENT REMOVAL OPTIONS

Remove: Document and physical file

Ask for confirmation: No

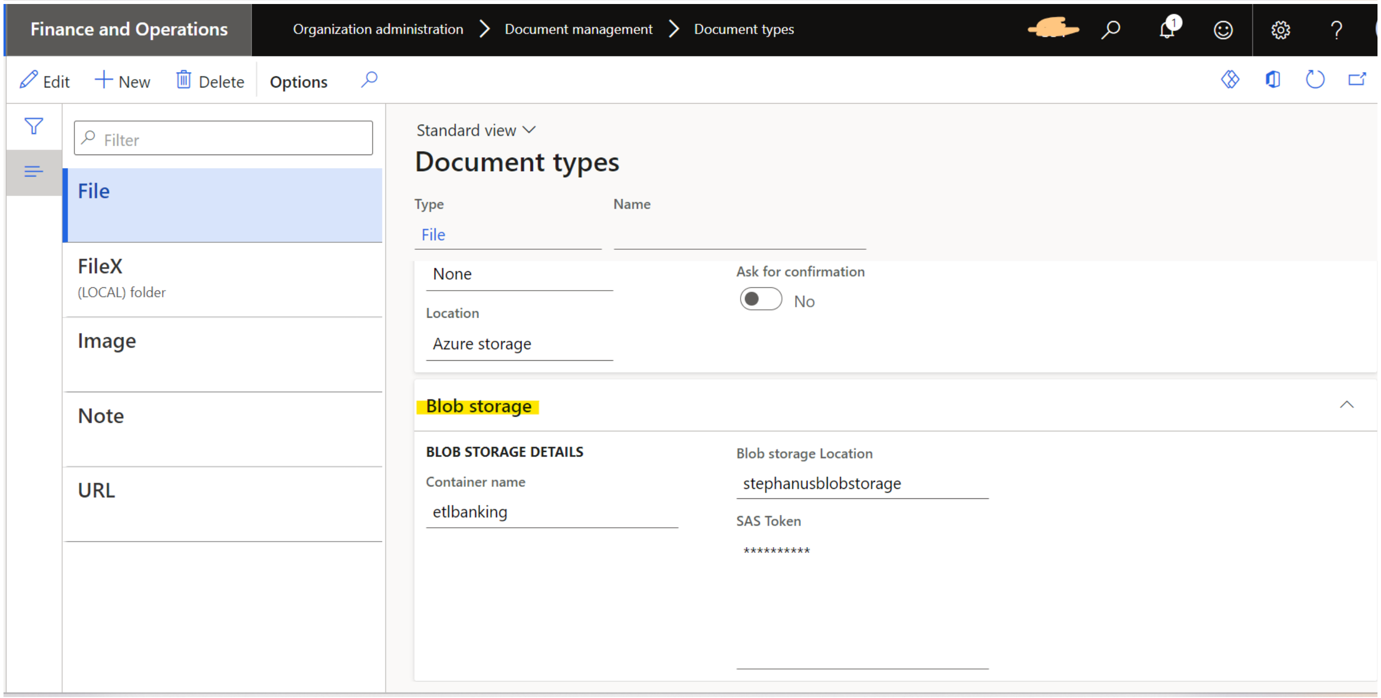
Blob storage

BLOB STORAGE DETAILS

Container name: etlbanking

Blob storage Location: stephanusblobstorage

SAS Token: *********



Document types

Type: File

Name:

None

Ask for confirmation: No

Location: Azure storage

Blob storage

BLOB STORAGE DETAILS

Container name: etlbanking

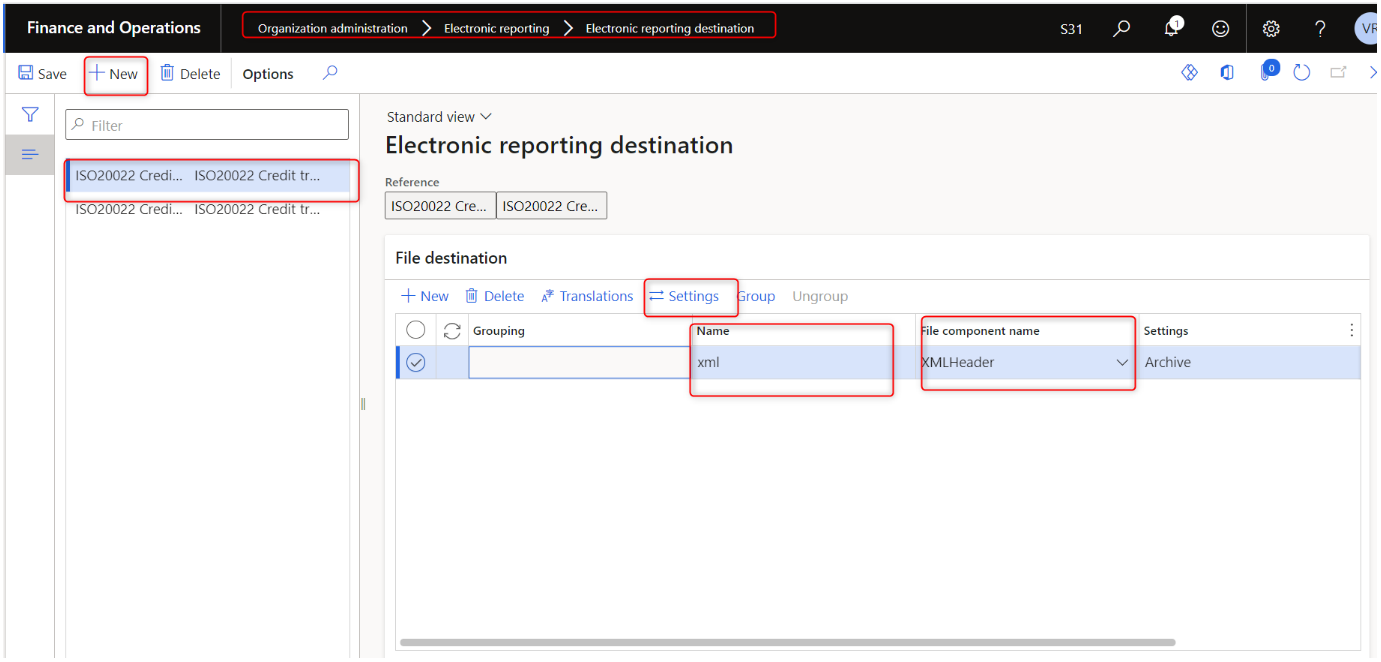
Blob storage Location: stephanusblobstorage

SAS Token: *********

Make sure you are configuring the blob storage details as well if you are using the blob storage to store the files.

ER Destination Configuration

- Electronic reporting destination will be setup to store the file in the archive as shown below.



Organization administration > Electronic reporting > Electronic reporting destination

ISO20022 Credi...	ISO20022 Credit tr...
ISO20022 Credi...	ISO20022 Credit tr...

Standard view

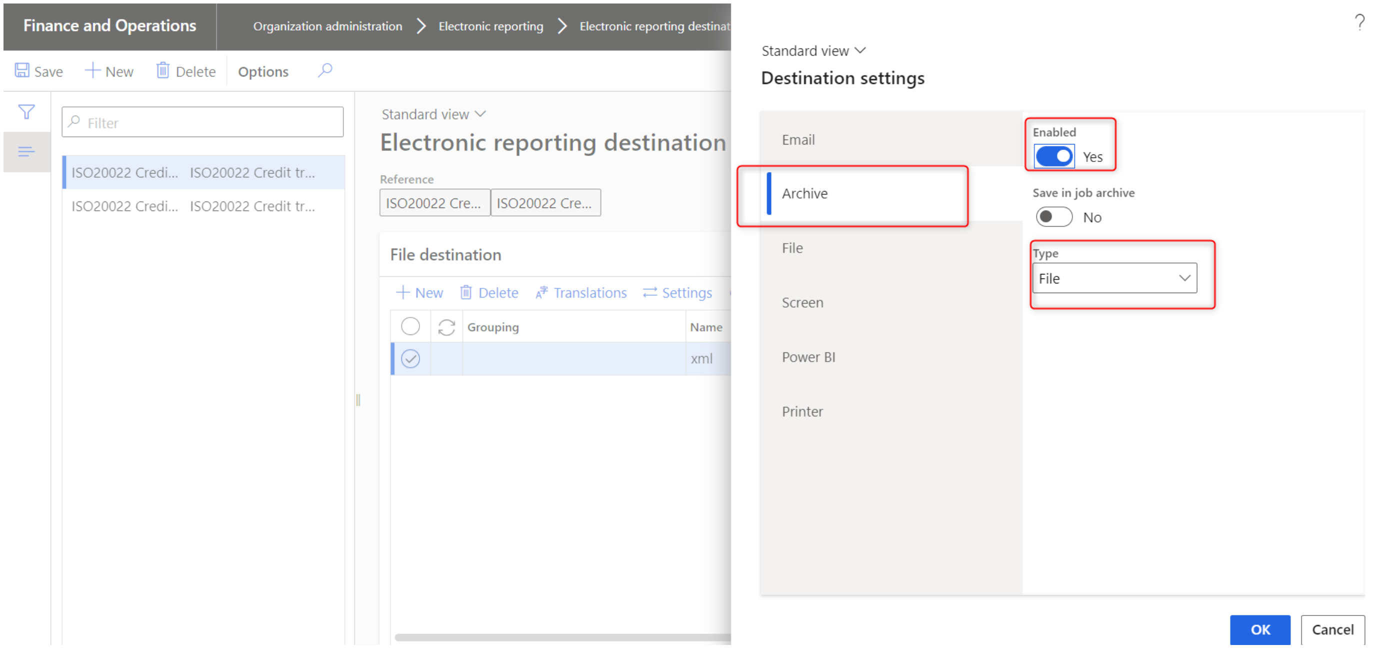
Electronic reporting destination

Reference: ISO20022 Cre... ISO20022 Cre...

File destination

New | Delete | Translations | Settings | Group | Ungroup

Grouping	Name	File component name	Settings
	xml	XMLHeader	Archive


Finance and Operations
Organization administration > Electronic reporting > Electronic reporting destinat
Save
New
Delete
Options
Filter
ISO20022 Credi... ISO20022 Credit tr...
ISO20022 Credi... ISO20022 Credit tr...
Standard view
Electronic reporting destination
Reference
ISO20022 Cre... ISO20022 Cre...
File destination
New
Delete
Translations
Settings
Grouping
Name
xml
?
Email
Archive
File
Screen
Power BI
Printer
OK
Cancel

Standard view

Destination settings

Archive

Enabled: Yes

Save in job archive: No

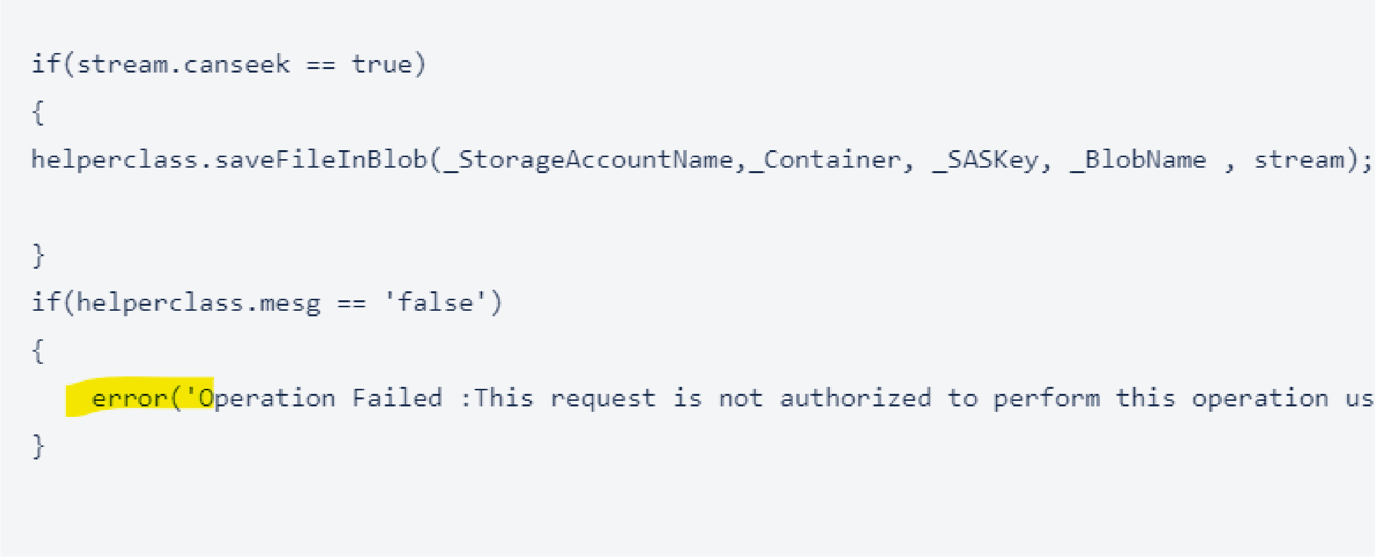
Type: File

4. Write a logic in X++ to call file and send to Blob storage with help of C Sharp Class.

Refer the below code snippet for the logic used.

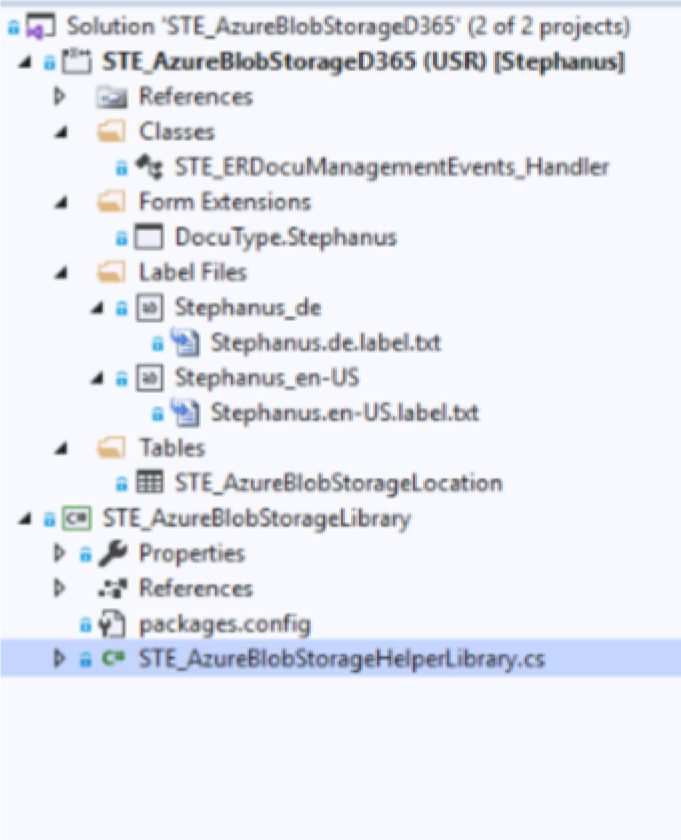
```
/// <summary>
/// This class is used to pass the files from F&O to Azure blob storage
/// </summary>
using Microsoft.Azure;
using Microsoft.WindowsAzure.Storage;
using Microsoft.WindowsAzure.Storage.Blob;
using STE_AzureBlobStorageLibrary;
internal final class STE_ERDocuManagementEvents_Handler_Old
{
    InteropPermission      Permission;
    System.Exception       Exception;


/// <summary>
/// This method is used to pass the files from F&O to blob storage
/// </summary>
/// <param name="_args">ERDocuManagementAttachingFileEventArgs</param>
[SubscribesTo(classStr(ERDocuManagementEvents), staticDelegateStr
(ERDocuManagementEvents, attachingFile))]
public static void ERDocuManagementEvents_attachingFile
(ERDocuManagementAttachingFileEventArgs _args)
{
    STE_AzureBlobStorageLocation    azureBlobStorageLocation,
azureBlobStorageLocationLoc;

    select firstonly azureBlobStorageLocation;
    str _StorageAccountName = azureBlobStorageLocation.
BlobStoragePath;
    str _Container =   azureBlobStorageLocation.
BlobContainer;
    str _SASKey =azureBlobStorageLocation.SasToken;
    str _BlobName = _args.getAttachmentName();
    System.IO.Stream _stream = _args.getStream();

    DocuType docuType = DocuType::find (_args.getDocuTypeId());
    _args.markAsHandled();
    var stream = _args.getStream();
    if (stream.CanSeek)
    {
        stream.Seek(0, System.IO.SeekOrigin::Begin);
    }

    select count(RecId) from azureBlobStorageLocationLoc;
    int countvar = azureBlobStorageLocationLoc.RecId;

    if(countvar >1)
    {
        delete_from azureBlobStorageLocation;
    }
    STE_AzureBlobStorageLibrary.STE_AzureBlobStorageHelperLibrary
helperclass = new   STE_AzureBlobStorageLibrary.
STE_AzureBlobStorageHelperLibrary();

    if(stream.canseek == true)
    {
    helperclass.saveFileInBlob(_StorageAccountName,_Container, _SASKey,
_BlobName , stream);

    }
    if(helperclass.mesg == 'false')
    {
        error('Operation Failed :This request is not authorized to
perform this operation using this permission ,Please make sure that
containername is correct ,Blob storage location is correct and before
genearting SAS token provide Add,Read,Write,Create permissions and SAS
token should generate for HTTPS and HTTP Allowed protocols in
blobstorage');
    }
}
}
```

ⓘ Make sure the SAS token expiry date, other wise it will expire the Token and throw an validation error because of logic given in X++ code



```
if(stream.canseek == true)
{
helperclass.saveFileInBlob(_StorageAccountName,_Container, _SASKey, _BlobName , stream);


}
if(helperclass.mesg == 'false')
{
   error('Operation Failed :This request is not authorized to perform this operation us
}
```

Then add the Both X++ project and C# project in same solution for easy access of both project objects.

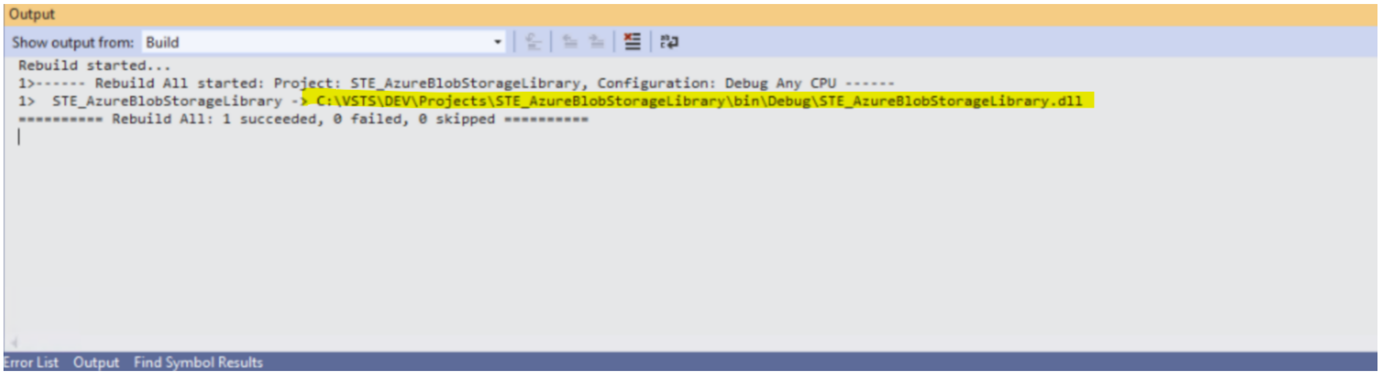


```
Solution 'STE_AzureBlobStorageD365' (2 of 2 projects)
  STE_AzureBlobStorageD365 (USR) [Stephanus]
    References
    Classes
      STE_ERDocuManagementEvents_Handler
    Form Extensions
      DocuType.Stephanus
    Label Files
      Stephanus_de
        Stephanus.de.label.txt
      Stephanus_en-US
        Stephanus.en-US.label.txt
    Tables
      STE_AzureBlobStorageLocation
  STE_AzureBlobStorageLibrary
    Properties
    References
    packages.config
    STE_AzureBlobStorageHelperLibrary.cs
```

5. Adding both C sharp and X++ projects in one solution.

Add the C# Project DLL reference in X++ Project

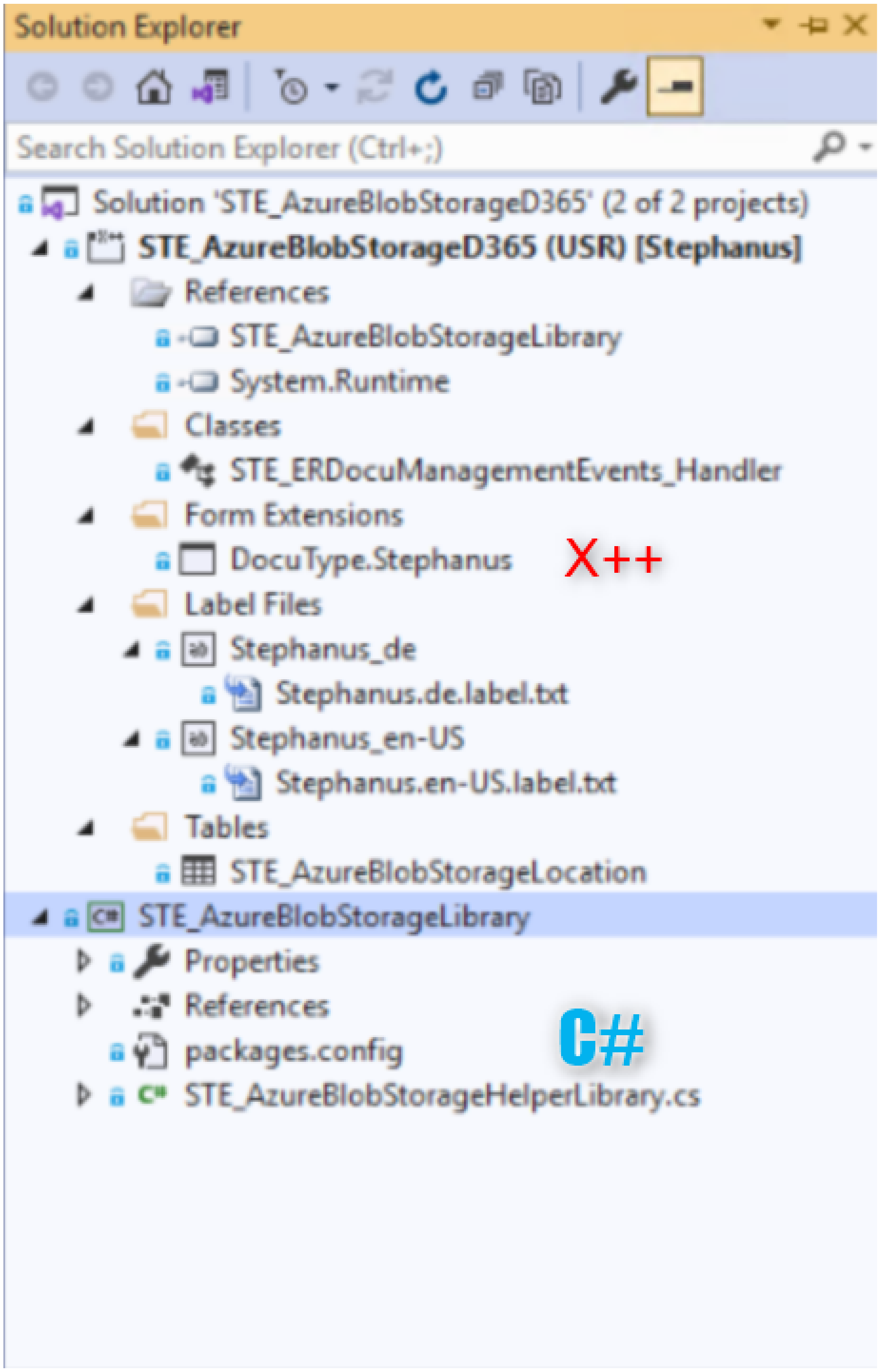
We have to add C # project DLL as reference to X++ project to call the classes from C#. When we build a C # library , it will show like this in output window



Show output from: Build

```
Rebuild started...
1>------ Rebuild All started: Project: STE_AzureBlobStorageLibrary, Configuration: Debug Any CPU ------
1>  STE_AzureBlobStorageLibrary -> C:\VSTS\DEV\Projects\STE_AzureBlobStorageLibrary\bin\Debug\STE_AzureBlobStorageLibrary.dll
========== Rebuild All: 1 succeeded, 0 failed, 0 skipped ==========
```

-



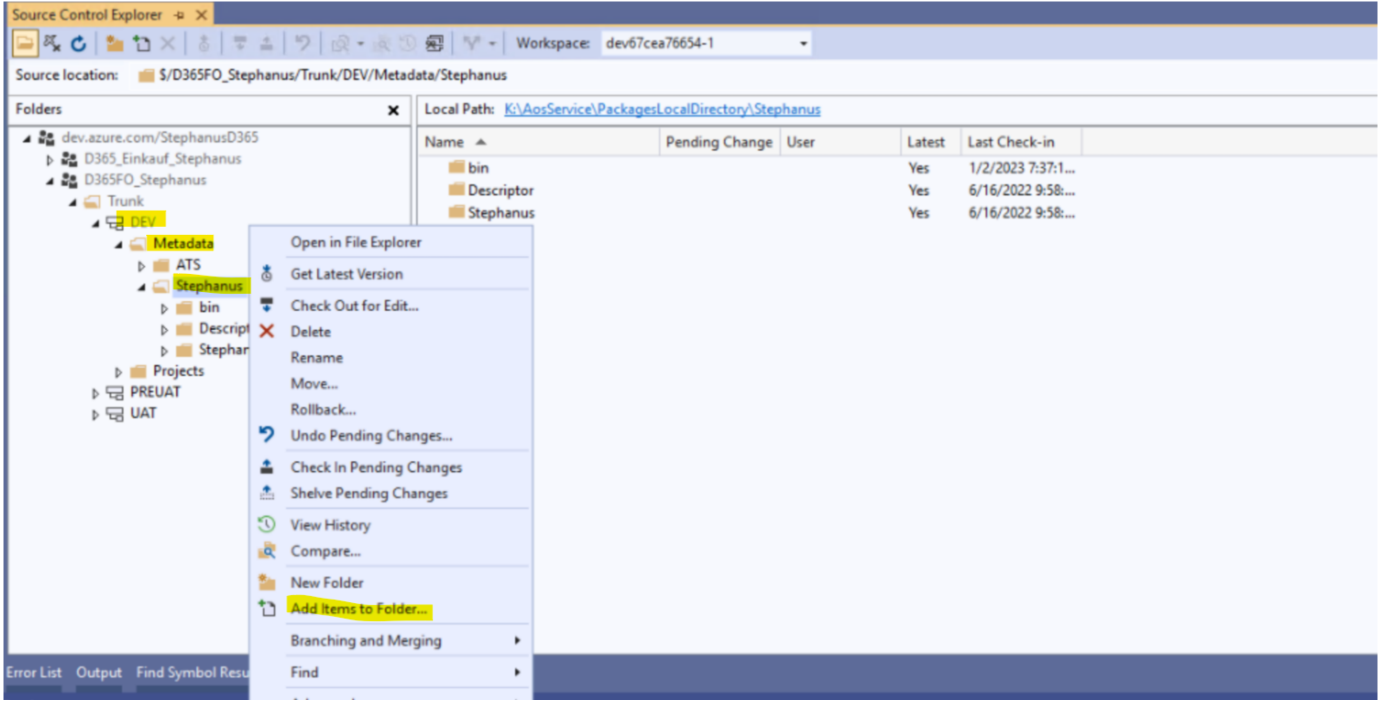
```
Solution 'STE_AzureBlobStorageD365' (2 of 2 projects)
  STE_AzureBlobStorageD365 (USR) [Stephanus]
    References
      STE_AzureBlobStorageLibrary
      System.Runtime
    Classes
      STE_ERDocuManagementEvents_Handler
    Form Extensions
      DocuType.Stephanus
    Label Files
      Stephanus_de
        Stephanus.de.label.txt
      Stephanus_en-US
        Stephanus.en-US.label.txt
    Tables
      STE_AzureBlobStorageLocation
  STE_AzureBlobStorageLibrary
    Properties
    References
    packages.config
    STE_AzureBlobStorageHelperLibrary.cs
```

- Once you have a reference then AxReference element is added to your model and the assembly (dll) is automatically copied to the bin folder of your package.
- File location:-
- K:\AosService\PackagesLocalDirectory\<Model Name>\<Model Name>\AxReference

By default, this dll's will not be part of version control. hence please add the same.

Add the DLL in the Metadata of Source Control

1. Go to Dev branch you connected and right click on the mode you are working and click on “Add items to folder”.



Workspace: dev67cea76654-1

Source location: $/D365FO_Stephanus/Trunk/DEV/Metadata/Stephanus

Local Path: K:\AosService\PackagesLocalDirectory\Stephanus

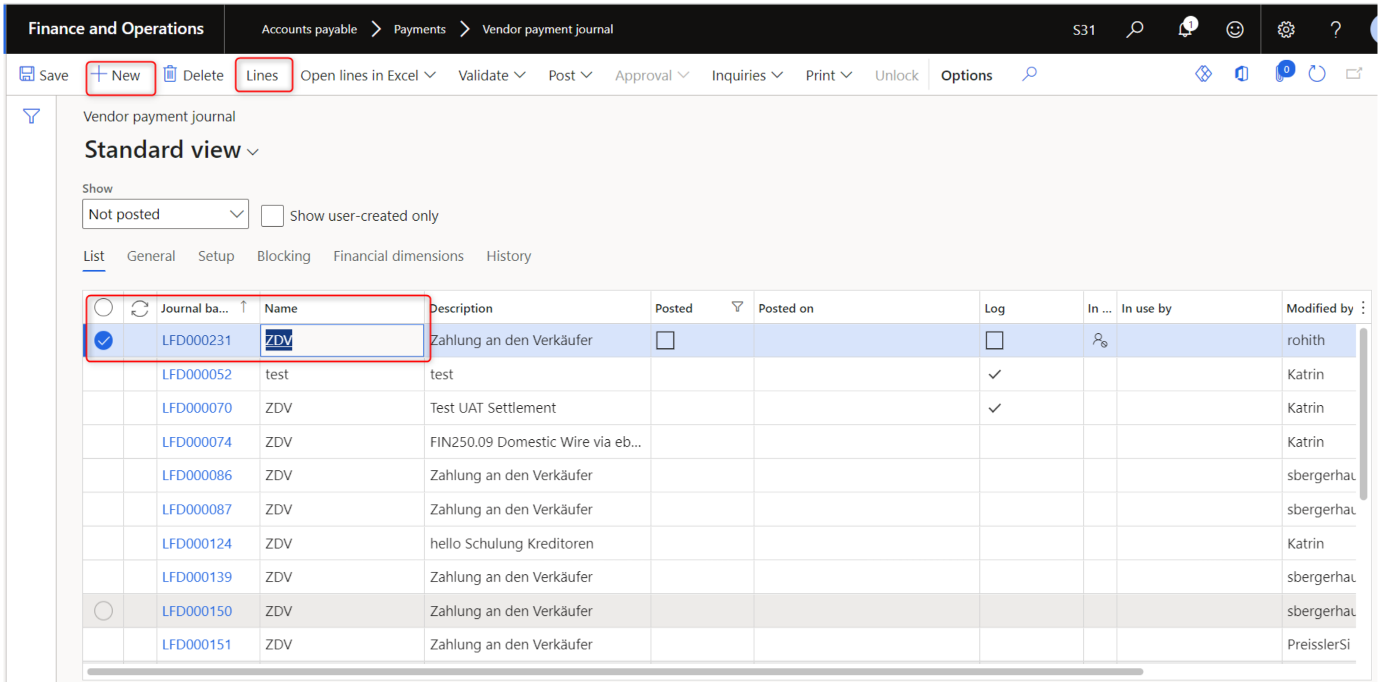
Name	Pending Change	User	Latest	Last Check-in
bin			Yes	1/2/2023 7:37:1...
Descriptor			Yes	6/16/2022 9:58:...
Stephanus			Yes	6/16/2022 9:58:...

Then navigate to C # project path like :

- **C:\VSTS\DEV\Projects\STE_AzureBlobStorageLibrary\bin\Debug\STE_AzureBlobStorageLibrary.dll**

Make sure that when Check-in the code, check in these DLL file as well.

Testing Scenario



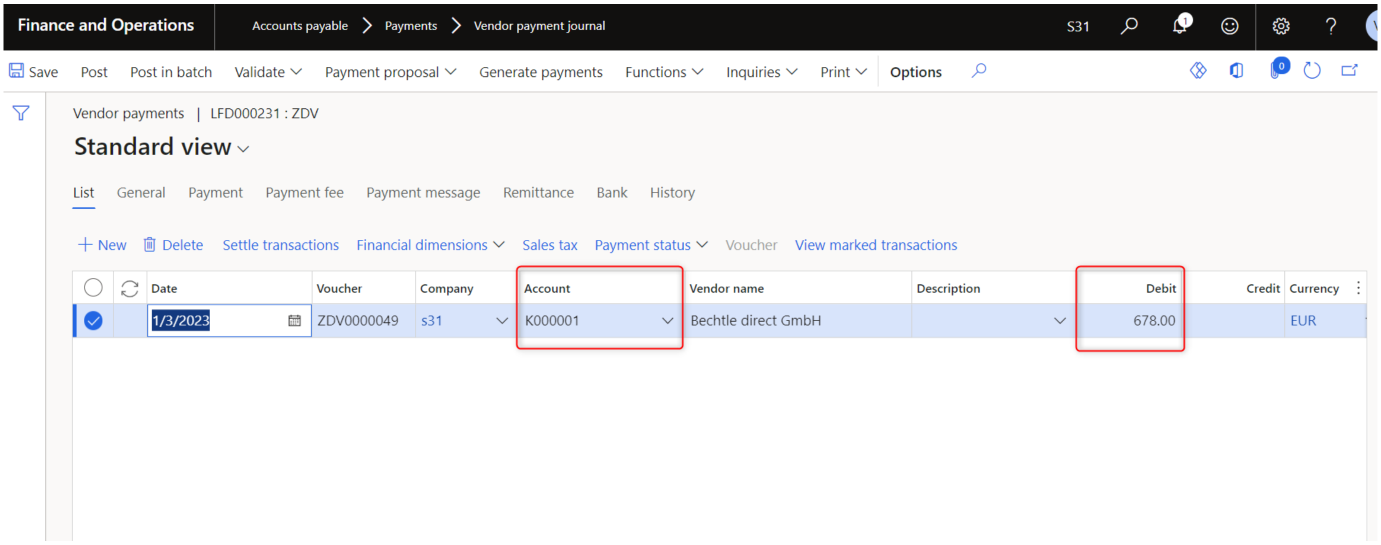
Vendor payment journal

Standard view

Show: Not posted

☐ Show user-created only

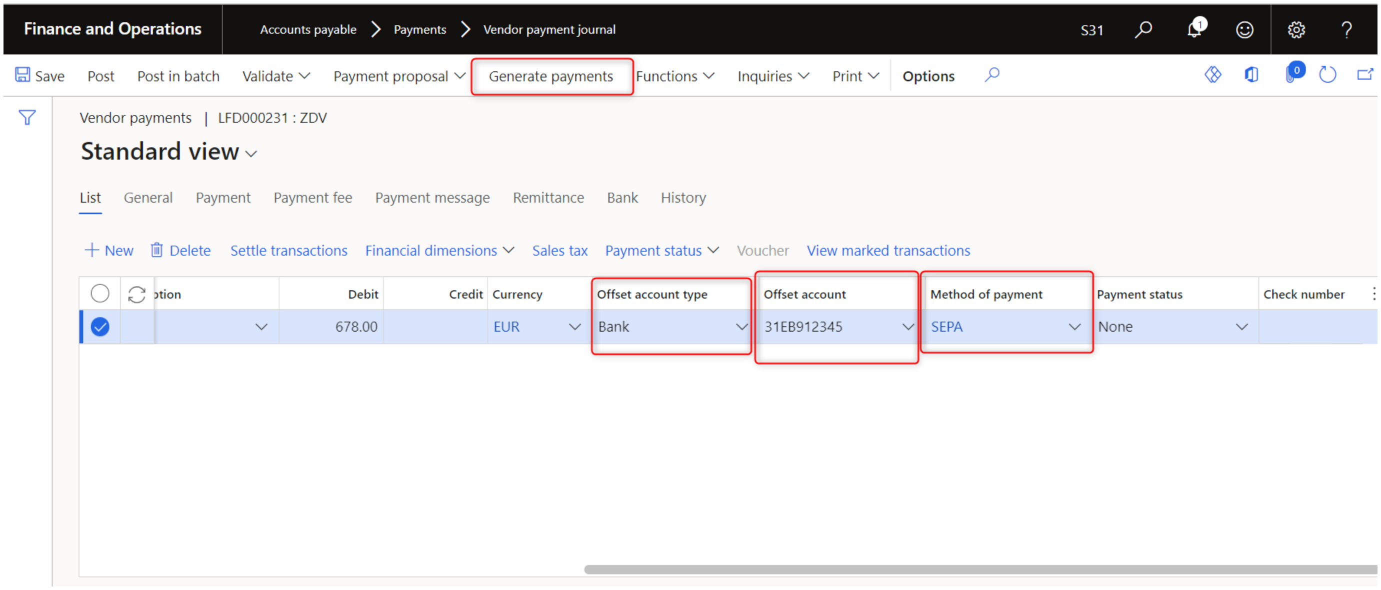
Journal ba...	Name	Description	Posted	Posted on	Log	In ...	In use by	Modified by
LFD000231	ZDV	Zahlung an den Verkäufer	☐		☐			rohith
LFD000052	test	test			✓			Katrin
LFD000070	ZDV	Test UAT Settlement			✓			Katrin
LFD000074	ZDV	FIN250.09 Domestic Wire via eb...						Katrin
LFD000086	ZDV	Zahlung an den Verkäufer						sbergerhau
LFD000087	ZDV	Zahlung an den Verkäufer						sbergerhau
LFD000124	ZDV	hello Schulung Kreditoren						Katrin
LFD000139	ZDV	Zahlung an den Verkäufer						sbergerhau
LFD000150	ZDV	Zahlung an den Verkäufer						sbergerhau
LFD000151	ZDV	Zahlung an den Verkäufer						PreisslerSi



Vendor payments | LFD000231 : ZDV

Standard view

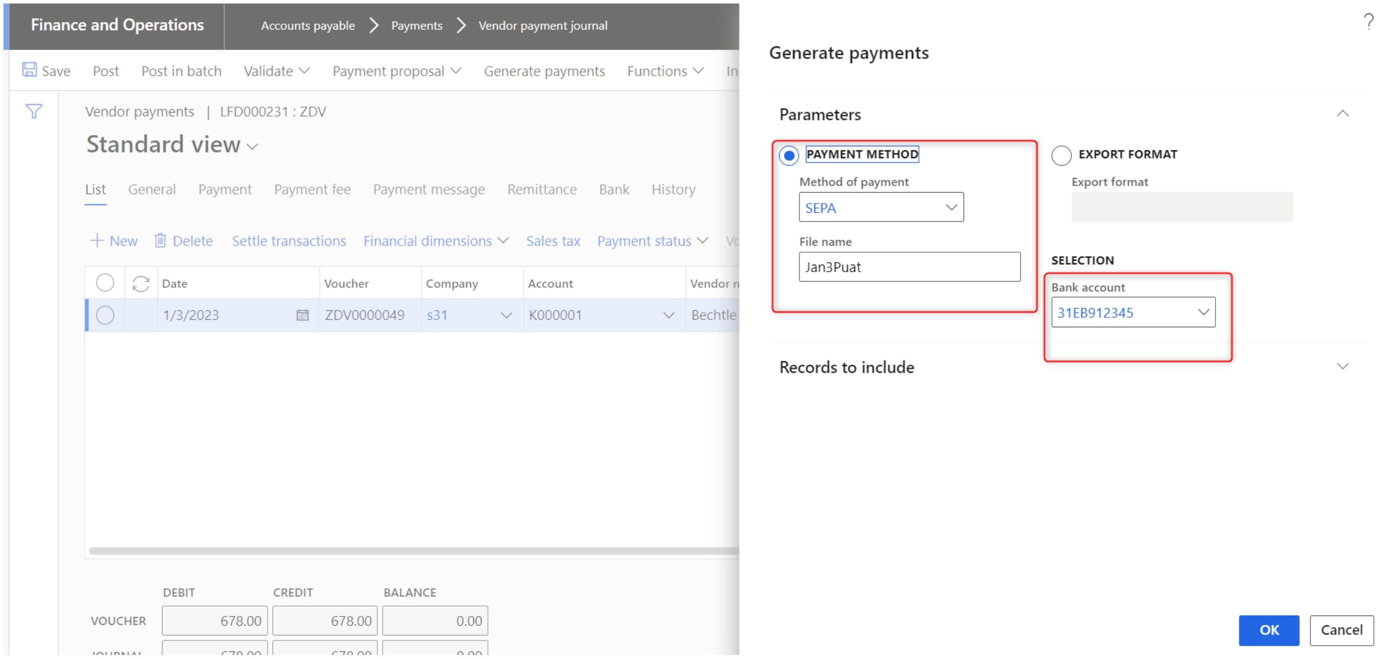
Date	Voucher	Company	Account	Vendor name	Description	Debit	Credit	Currency
1/3/2023	ZDV0000049	s31	K000001	Bechtle direct GmbH		678.00		EUR



Vendor payments | LFD000231 : ZDV

Standard view

ption	Debit	Credit	Currency	Offset account type	Offset account	Method of payment	Payment status	Check number
	678.00		EUR	Bank	31EB912345	SEPA	None	



Vendor payments | LFD000231 : ZDV

Standard view

Date	Voucher	Company	Account	Vendor n
1/3/2023	ZDV0000049	s31	K000001	Bechtle

	DEBIT	CREDIT	BALANCE
VOUCHER	678.00	678.00	0.00

Generate payments

Parameters

● PAYMENT METHOD

Method of payment: SEPA

File name: Jan3Puat

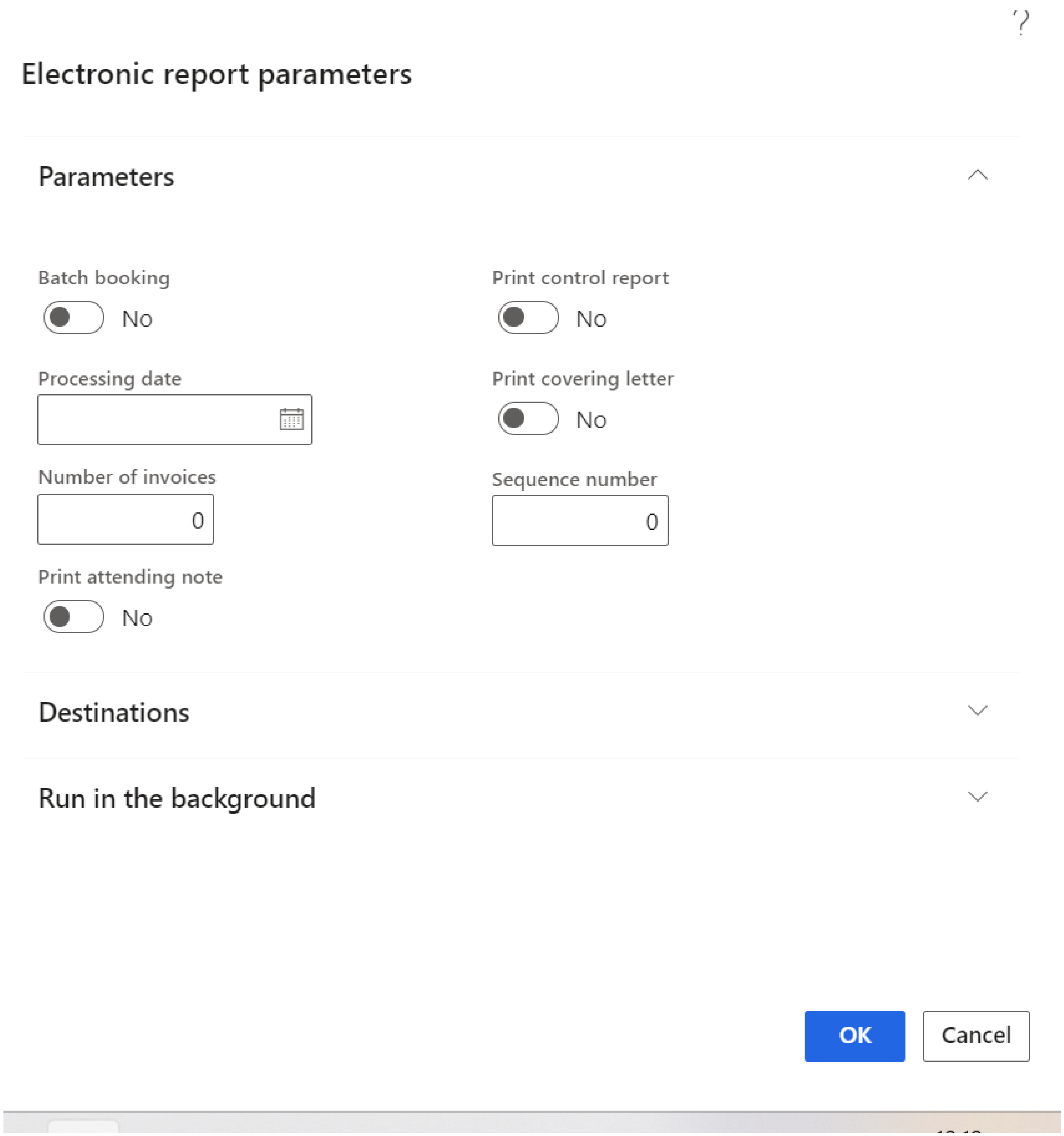
○ EXPORT FORMAT

Export format

SELECTION

Bank account: 31EB912345

Records to include



Electronic report parameters

Parameters

Batch booking
○ No

Print control report
○ No

Processing date

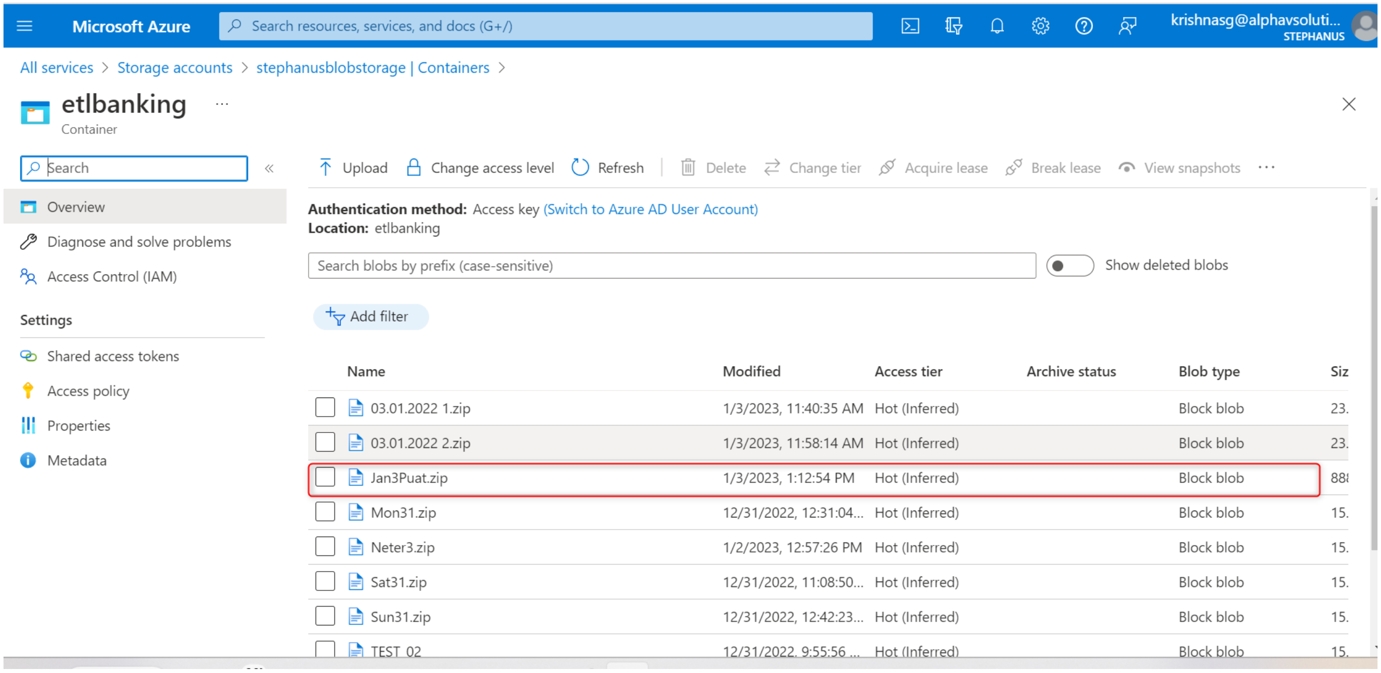
Print covering letter
○ No

Number of invoices
0

Sequence number
0

Print attending note
○ No

Finally, our file is in the blob storage 😀



Microsoft Azure
Search resources, services, and docs (G+/)
krishnasg@alphavsoluti...
STEPHANUS

All services > Storage accounts > stephanusblobstorage | Containers >

etlbanking ...

Container

Authentication method: Access key (Switch to Azure AD User Account)
Location: etlbanking

Search blobs by prefix (case-sensitive)

Show deleted blobs

	Name	Modified	Access tier	Archive status	Blob type	Siz
□	03.01.2022 1.zip	1/3/2023, 11:40:35 AM	Hot (Inferred)		Block blob	23.
□	03.01.2022 2.zip	1/3/2023, 11:58:14 AM	Hot (Inferred)		Block blob	23.
□	Jan3Puat.zip	1/3/2023, 1:12:54 PM	Hot (Inferred)		Block blob	88
□	Mon31.zip	12/31/2022, 12:31:04...	Hot (Inferred)		Block blob	15.
□	Neter3.zip	1/2/2023, 12:57:26 PM	Hot (Inferred)		Block blob	15.
□	Sat31.zip	12/31/2022, 11:08:50...	Hot (Inferred)		Block blob	15.
□	Sun31.zip	12/31/2022, 12:42:23...	Hot (Inferred)		Block blob	15.
□	TEST_02	12/31/2022, 9:55:56 ...	Hot (Inferred)		Block blob	15.