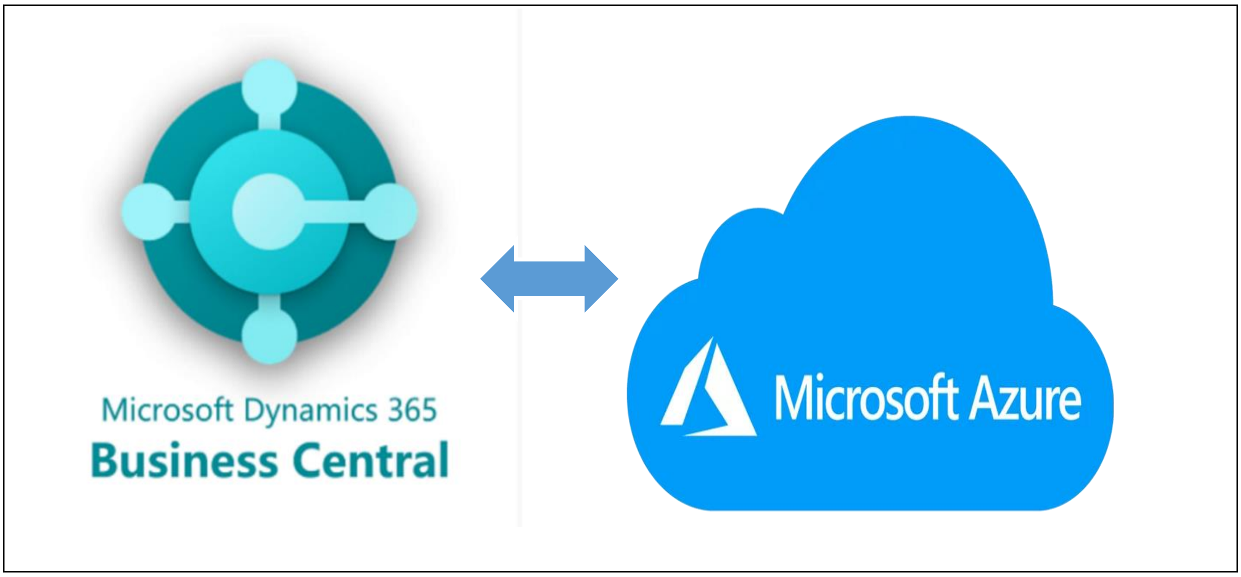
Azure Blob Storage Integration with Microsoft Dynamics 365 Business Central



Problem Statement:

In today's data-driven world data is everywhere but should it be stored in an old-fashioned Local data storage or a Cloud data storage, all depends on how that data is going to be used.

Businesses most likely rely on the data in their daily operations & it's important to store and manage large volumes of data related to Sales, Customers, Purchases & Vendors.

One of my clients used ERP Software called Microsoft Dynamics 365 Business Central (SAAS) and they are storing their daily operations product & customer details from Dynamics Business Central to Local On-premises storage that are long been the standard in the industry but managing them can be a complex and challenging task. But as days pass & data grows local storage options are hard to use due to high maintenance costs, limited scalability, low chances of data availability, No data level tiering, Manual Data backups & additional IT Support needs.

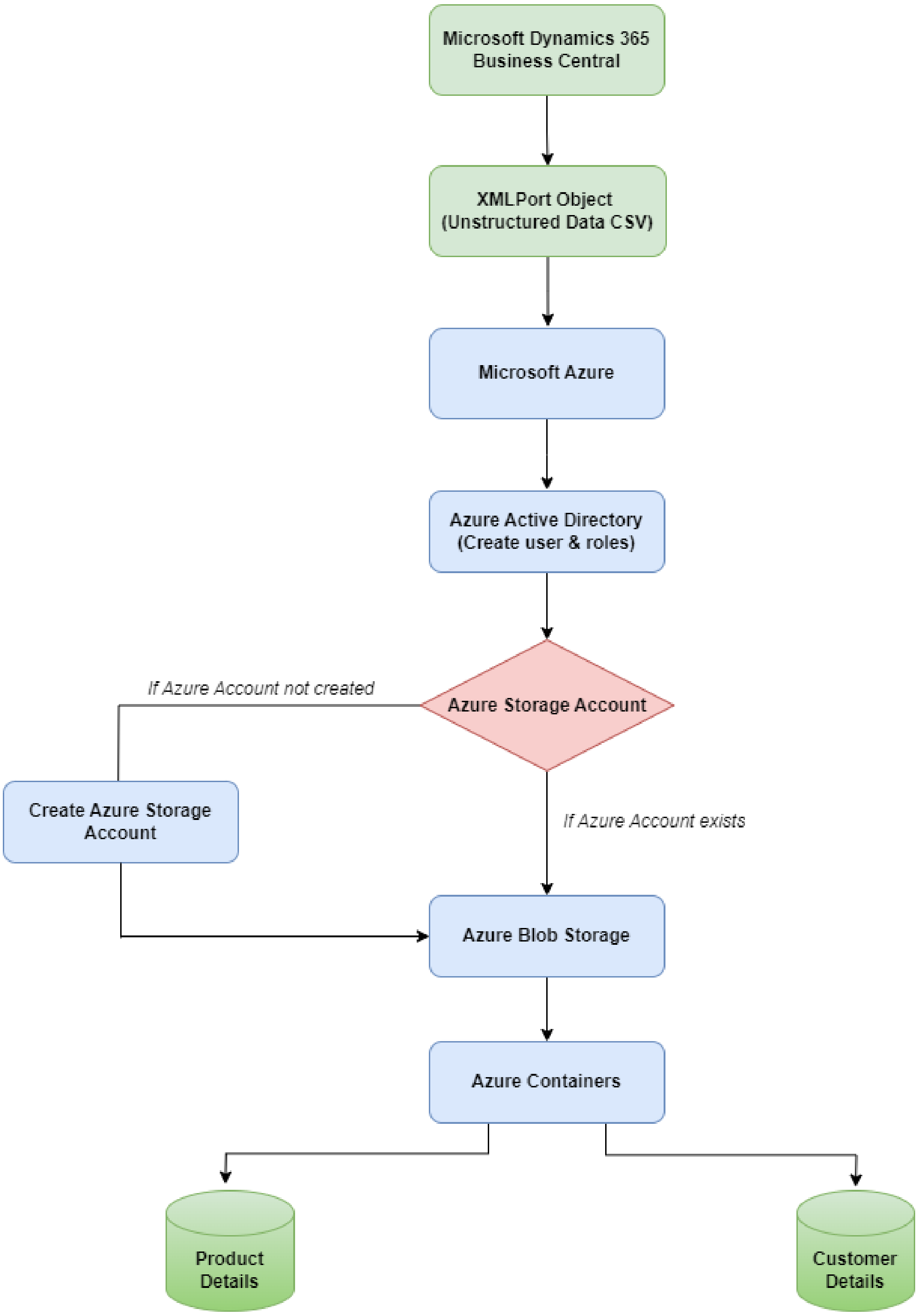
As Cloud Application Developer I have taken responsibility to solve data storage challenges and provide the best solution to my clients.

The goal of this blog is to offer best practices to store & manage daily business operations unstructured data in Azure storage to seamlessly store manage data to address any storage difficulties that businesses may face along the way.

Solution & Architecture:

I have chosen Microsoft Azure to use data storage, Azure is a cloud computing platform that enables you to use Microsoft cloud services including computing, data storage, networking, analytics, and data transfer as per your needs.

Whereas Azure Blob storage is Microsoft's object storage solution service to optimize for storing a large amount of unstructured data (binary & text data).


Microsoft Dynamics 365 Business Central
XMLPort Object (Unstructured Data CSV)
Microsoft Azure
Azure Active Directory (Create user & roles)
If Azure Account not created
Azure Storage Account
Create Azure Storage Account
If Azure Account exists
Azure Blob Storage
Azure Containers
Product Details
Customer Details

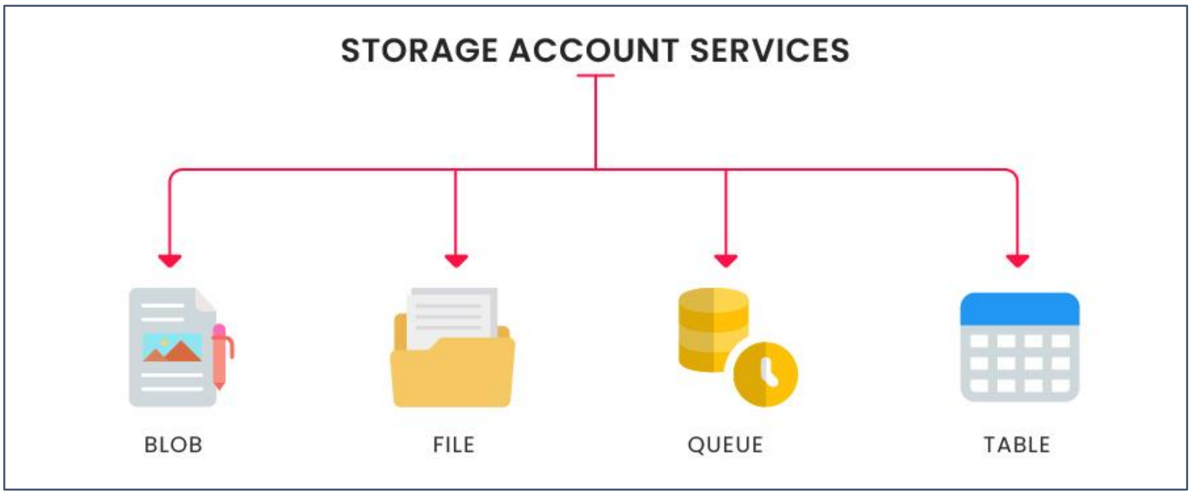
Microsoft Azure Services:

Microsoft Azure offers a wide range of cloud services to help organizations harness the power of the cloud computing infrastructure, Following is a list of different types of cloud services that Azure currently provides.

- Compute
- Storage
- Databases & Analytics
- Networking
- Development
- Security & Identity
- Internet of things
- Integration
- Machine Learning & AL
- Management & Governance
- Media
- Migration

Azure Storage Services:

Azure Storage offers different core services for storing and managing both structured and unstructured data.



Azure Blob Storage:

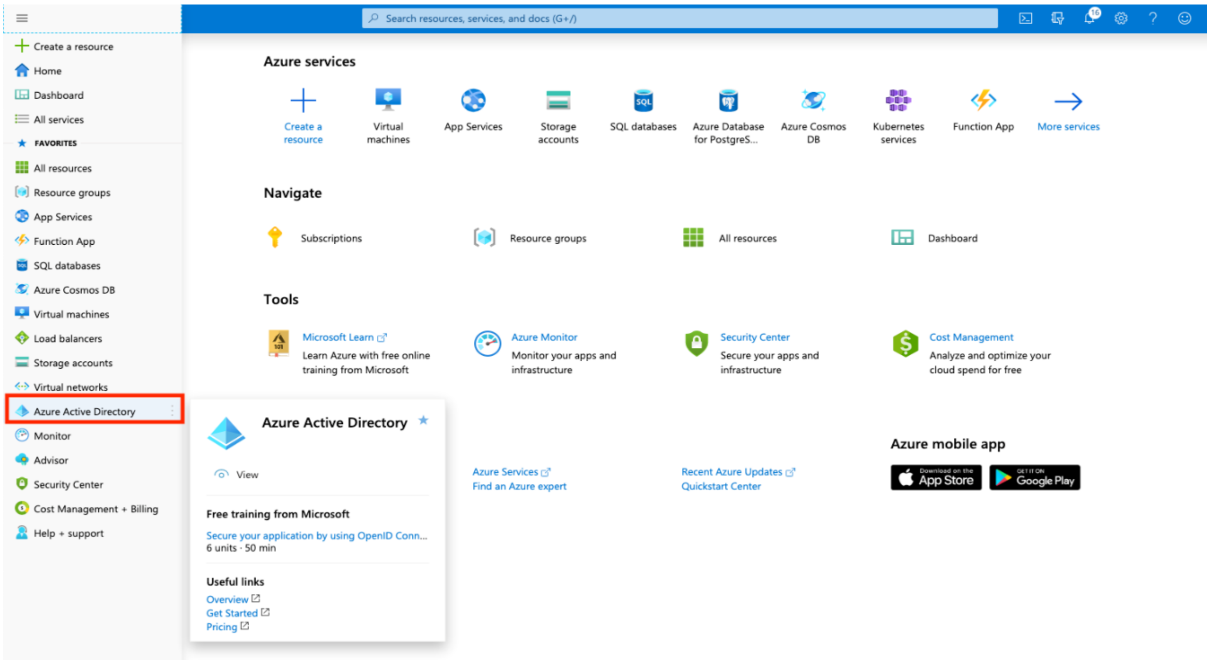
For data storage, I have used Azure Blob Storage Service in Azure, Blob Storage makes stored data available to businesses over the internet and particularly useful for storing media, such as audio and video, and frequently changing data, such as log files. A blob, or binary large object, is data in binary form.

Azure Blob Storage can also be used for data or device backup also you can store any number of blob files up to a total size of 500 TB. It is a cost-effective solution for managing and storing massive amounts of data and it allows you to pay only for the storage you need.

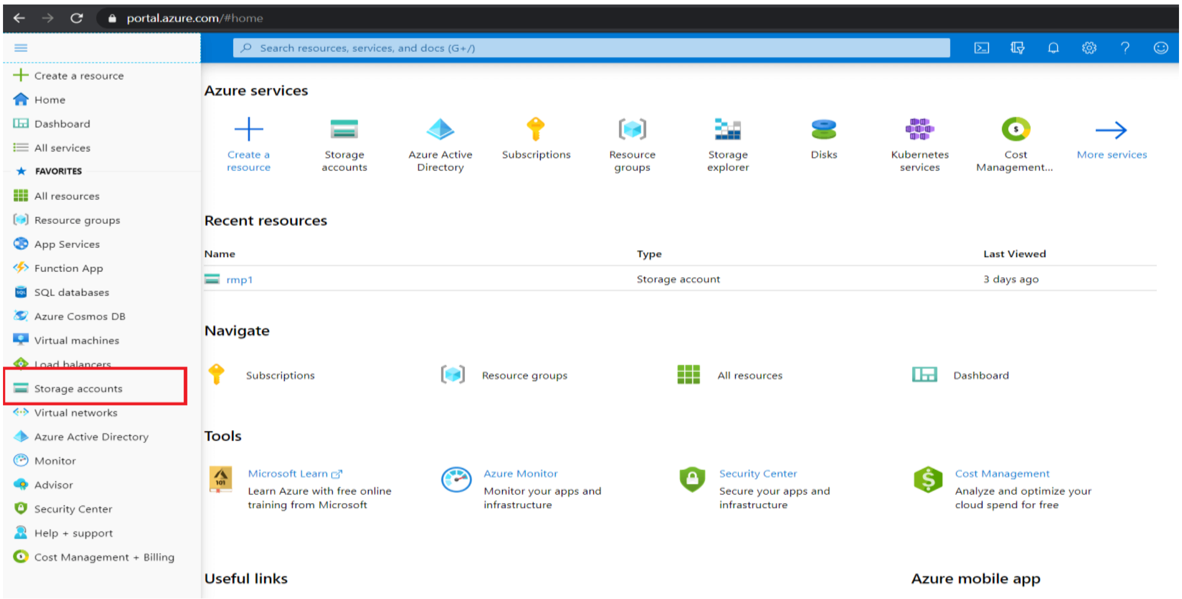
Technical Details & Implementation of solution:

- Programming Language Used: AL Programming
- Tools & Software: Microsoft Dynamics 365 Business Central, Visual Studio Code
- Azure Services Used: Azure Blob Storage, Storage Account, Azure Active Directory

Starting with Azure Active directory which is used to authenticate a user & permissions assigned to user. I have created a new user & Group relevant to the user, Groups will depend on what kind of access the new user will need to have (storage account & Blob Access).

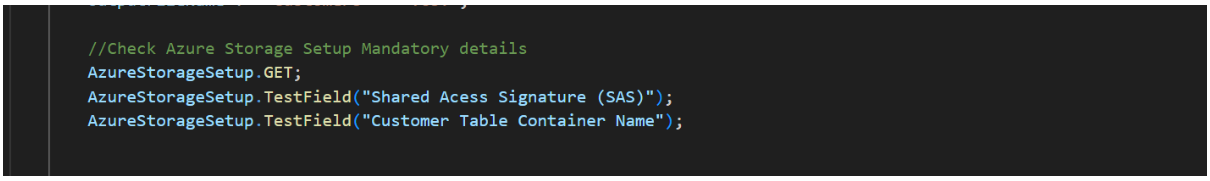


Created Azure Account by choosing subscription, resource group, storage label, location, performance, account type, replication, and access tier, I have chosen the Hot Access tier because I must frequently access files in blob storage containers.



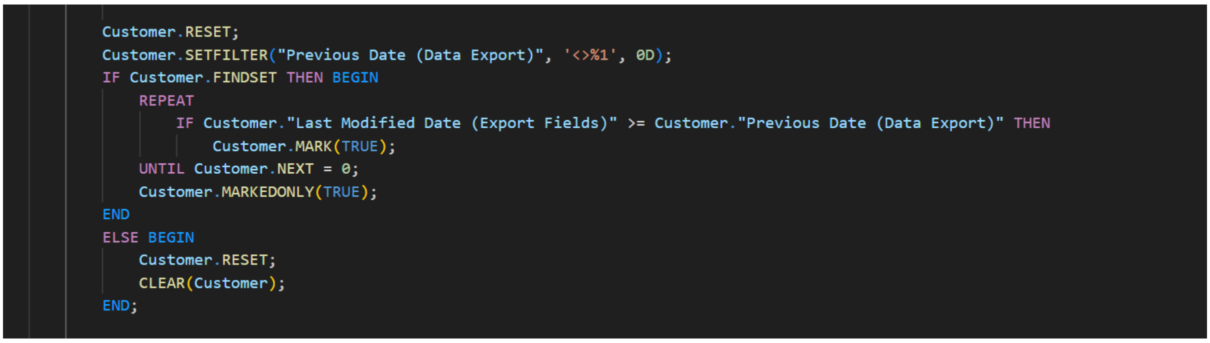
Name	Type	Last Viewed
rmp1	Storage account	3 days ago

- Azure storage setup table is created in Dynamics 365 Business Central to define storage Accounts mandatory details including Shared Access Signature (SAS Token), Blob Container labels which you must define first in Dynamics 365 Business Central



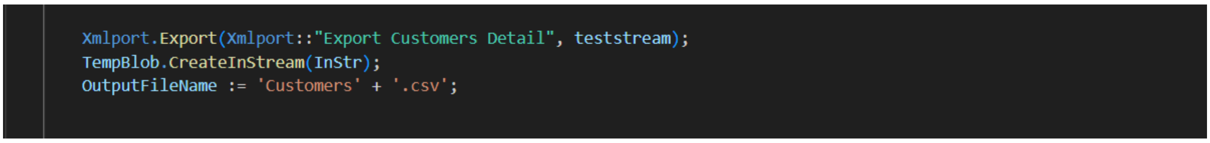
```
//Check Azure Storage Setup Mandatory details
AzureStorageSetup.GET;
AzureStorageSetup.TestField("Shared Acess Signature (SAS)");
AzureStorageSetup.TestField("Customer Table Container Name");
```

- Based on ‘Last Modified Date’ of data & ‘Previous Date of Data Export’, Customers data is to marked & ready to export to Azure blob storage.
- This step makes sure that there is not any data duplication & data redundancy occurring in azure storage. only new & modified data is exported after Previous Date of Data Export field.



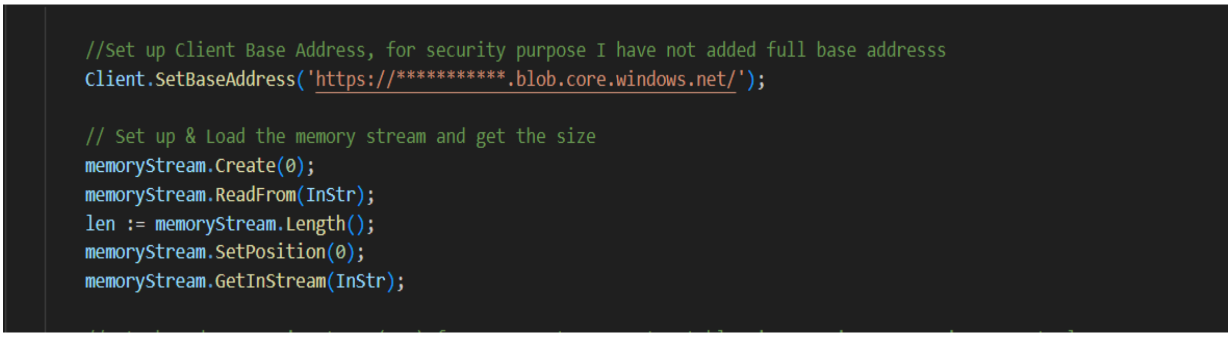
```
Customer.RESET;
Customer.SETFILTER("Previous Date (Data Export)", '<>%1', 0D);
IF Customer.FINDSET THEN BEGIN
    REPEAT
        IF Customer."Last Modified Date (Export Fields)" >= Customer."Previous Date (Data Export)" THEN
            Customer.MARK(TRUE);
    UNTIL Customer.NEXT = 0;
    Customer.MARKEDONLY(TRUE);
END
ELSE BEGIN
    Customer.RESET;
    CLEAR(Customer);
END;
```

- Filtered Customer details are exported in CSV file using XMLport Object which is used to export & import data between an external source and Dynamics 365 Business Central.



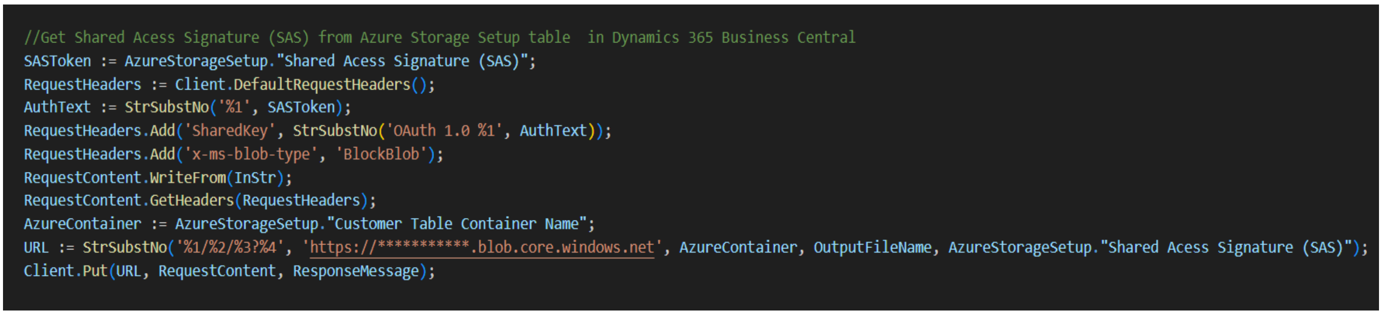
```
Xmlport.Export(Xmlport::"Export Customers Detail", teststream);
TempBlob.CreateInStream(InStr);
OutputFileName := 'Customers' + '.csv';
```

- Set up Client Base in this case Azure Storage Account URL will be base address for sending requests.
- Set up and load memoryStream and get the size & stream position.
- memoryStream encapsulates data stored as an unsigned byte array as this data directly accessible in memory.



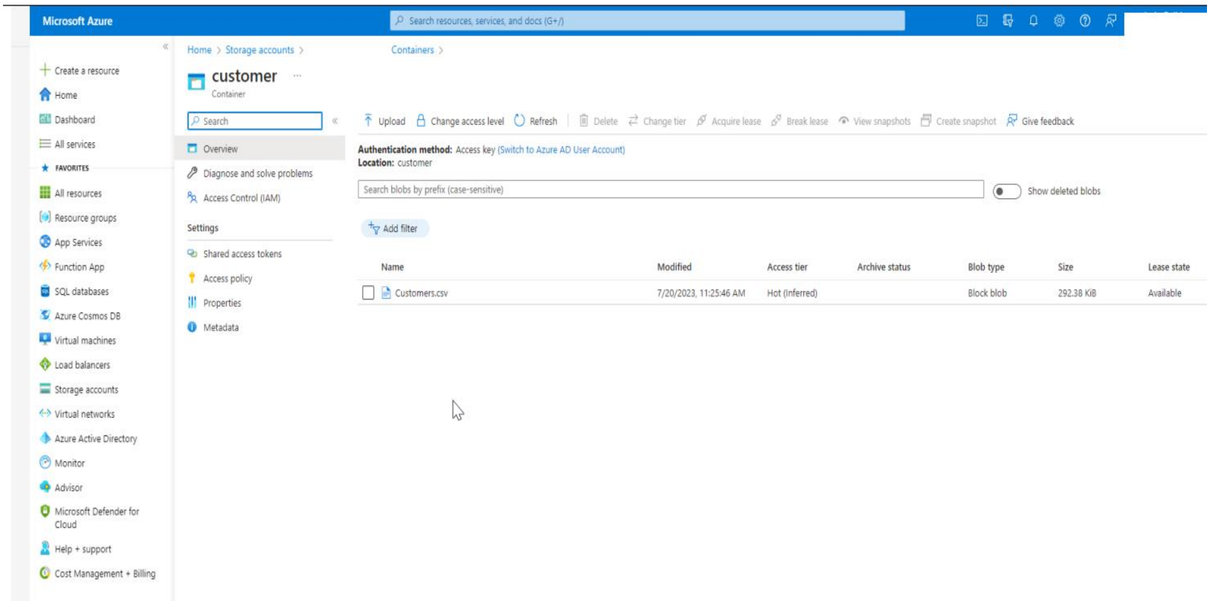
```
//Set up Client Base Address, for security purpose I have not added full base addresss
Client.SetBaseAddress('https://***********.blob.core.windows.net/');

// Set up & Load the memory stream and get the size
memoryStream.Create(0);
memoryStream.ReadFrom(InStr);
len := memoryStream.Length();
memoryStream.SetPosition(0);
memoryStream.GetInStream(InStr);
```

- Get Shared Access Signature (SAS) & Container Name from Azure Storage Setup Table in Dynamics Business Central to store data in target container.
- I have selected BlockBlob as a Blob type because it is primarily intended for storage of media files, documents, text files, and binary files.
- Target URI is created by reference Azure Container name, Output file name & Shared Access Signature (SAS)
- Sends a PUT request to the specified URI as an asynchronous operation in Azure Storage using Client Put Method.



```
//Get Shared Acess Signature (SAS) from Azure Storage Setup table  in Dynamics 365 Business Central
SASToken := AzureStorageSetup."Shared Acess Signature (SAS)";
RequestHeaders := Client.DefaultRequestHeaders();
AuthText := StrSubstNo('%1', SASToken);
RequestHeaders.Add('SharedKey', StrSubstNo('OAuth 1.0 %1', AuthText));
RequestHeaders.Add('x-ms-blob-type', 'BlockBlob');
RequestContent.WriteFrom(InStr);
RequestContent.GetHeaders(RequestHeaders);
AzureContainer := AzureStorageSetup."Customer Table Container Name";
URL := StrSubstNo('%1/%2/%3?%4', 'https://**********.blob.core.windows.net', AzureContainer, OutputFileName, AzureStorageSetup."Shared Acess Signature (SAS)");
Client.Put(URL, RequestContent, ResponseMessage);
```

- In Microsoft Azure Blob Storage Customers data is stored in target container Customer and target output file Customer.csv.
- During the initial extraction, the data will encompass all the relevant data from Customer table, Subsequent extractions will only include changes made to the customer table, allowing for efficient updates and synchronization with the latest Customer data from Dynamics Business Central.
- Once data is stored in Azure container file ‘Previous Date of Data Export’ is changed to Current date in Dynamics Business Central to target table (Customer).



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

	Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
□	Customers.csv	7/20/2023, 11:25:46 AM	Hot (Inferred)		Block blob	292.38 KiB	Available

- I have implemented the same process for Product detail & Sales Detail Masters to efficiently store newly created data in target Azure Blob Containers files.

Challenges in implementing the solution:

Microsoft Azure offers a wide range of storage services, the complexity of choosing the right Azure storage service to store data & use for solution was big challenge I faced as a developer. But through research and development I have chosen Azure blob storage which enables more flexibility and scalability & suitable for unstructured data.

Business Benefit:

- **Durability**: Using Azure Blob Storage my client kept multiple copies across various data centers to assure data availability and recovery.
- **Scalability**: Now any size of daily operation unstructured data is stored in containers which is simple to store and manage.
- **Cost-effective**: Microsoft Azure offers pay-as-you-go pricing, the client is now paying only for the storage they need with no upfront expenditures or termination fees.
- **Security**: Azure Blob Storage protects your data with numerous layers of protection, including role-based access control & shared access signatures.
- **Versioning**: Restore an earlier version of file in case of if my client mistakenly modifies or deletes file, so using blob version keeps the state of a blob at any given point in time.
- **Lifecycle Management Policy**: Set up policies to move data from frequent access to infrequent access, if after 4-5 years when the access of data will get lower for my client use.

Conclusion:

Using Microsoft Azure services in your business allows you to leverage the power of cloud computing by offering rich features, scalability, reliability, and security by enabling you to build robust feature rich applications.