Blue Prism Cloud Data Vault and Partitioner

Blue Prism Cloud offers a combined SQL Server and Azure Data Factory solution that is designed to mitigate common issues encountered with session log data accumulation within the Blue Prism database. The Blue Prism Cloud Data Vault and Partitioner process primarily focuses on maintaining the session logs and work queue items. Data from tables related to the session log and work queue items is copied over to be used for reporting purposes. This data is archived to a cost-effective columnar file format (Apache Parquet) in Azure Data Lake.

> Apache Parquet is an open source, column-oriented data file format designed for efficient data storage and retrieval. It provides efficient data compression and encoding schemes with enhanced performance to handle complex data in bulk. For more information, see https://parquet.apache.org/.

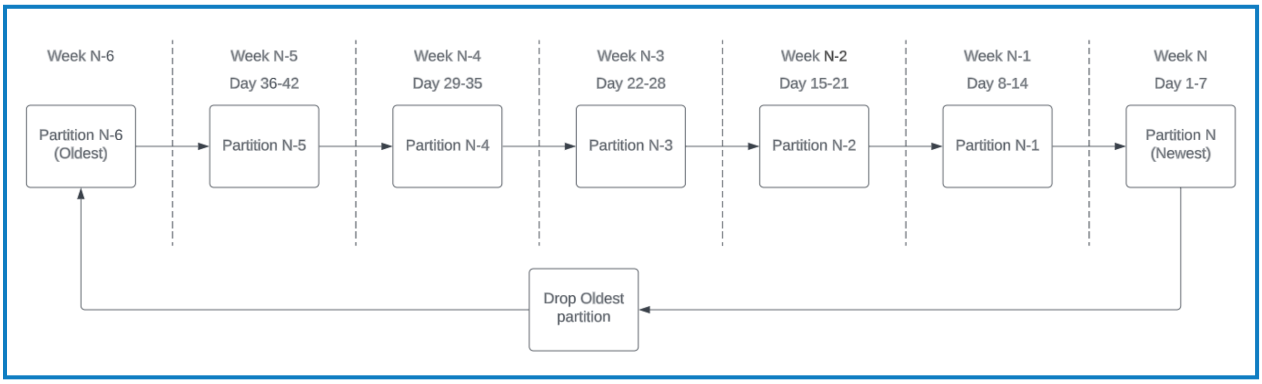
To maintain the database size, session log entries are regularly purged from the production database.

The solution is deployed in alignment with Blue Prism Cloud's SOC 2 verified data segregation policies. No resources are shared between customer subscriptions or accounts.

Overall Strategy

The goal of the Blue Prism Cloud Data Vault and Partitioner (Data Vault) solution is to keep the production database lean by moving the data from several key tables to a cost-effective and accessible archive. The configurable Data Vault solution syncs production data to the Azure Data Lake on a daily basis. Archived data older than one year is permanently purged from the Azure Data Lake. If required, you may download and store this data. For more information, see Data Security and Access on page 4.

The SQL Partitioner solution groups and purges the session log data by week, according to the pre-scheduled Weekly Reboot Automation. This process retains six weeks of data in the Production Databases session log table. The diagram below summarizes the process:



This solution requires a pre-planned database maintenance window to add, configure, and start the SQL Server partitioning (grouping) process. The Data Vault solution has little to no impact on performance, however Blue Prism Cloud will deploy and trigger the solution based on the Blue Prism Production Database's least busy time.

This Archiving solution focuses on the following tables:

- BPAAuditEvents
- BPAEnvironmentVar
- BPAProcess
- BPAProcessEnvironmentVarDependency
- BPARelease
- BPAResource
- BPASession
- BPASession log_NonUnicode
- BPASession log_Unicode
- BPAWorkQueue
- BPAWorkQueueItem
- BPAWorkQueueItemTag
- BPATag
- BPAUser

Components

Azure Data Factory

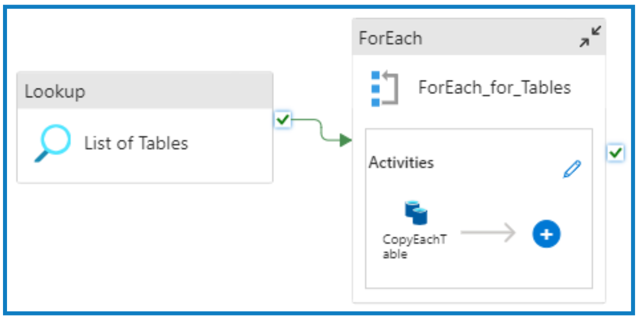
Azure Data Factory (ADF) provides Extract-Transform-Load (ETL) capability. A series of parameterized pipelines have been created which are deployed via Azure Resource Manager (ARM) templates and require minimal setup per subscription and database.

An ADF pipeline is a series of operations organized into a workflow. Pipelines can execute other pipelines and are scheduled to run an ADF trigger with configurable parameters. Some pipelines are configured to use a lookup table to store copied activity and this can be referenced in future pipeline runs.

Blue Prism Cloud has configured a Master pipeline that runs once daily. The Master pipeline contains four additional pipelines embedded within it. The pipelines are summarized below.

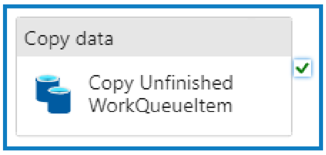
PIPELINE: LOAD MULTIPLE TABLES

This pipeline copies the BPAWorkQueueItemTag, BPATag, BPASession, BPAAuditEvents, BPAProcess, BPAProcessEnvironmentVarDependency, BPAEnvironmentVar, BPARelease, BPAResource, BPAUser, and BPAWorkQueue tables from the RPA Production database and loads them into Azure Data Lake storage once a day. This pipeline overwrites the existing file and only one instance of the data is kept.



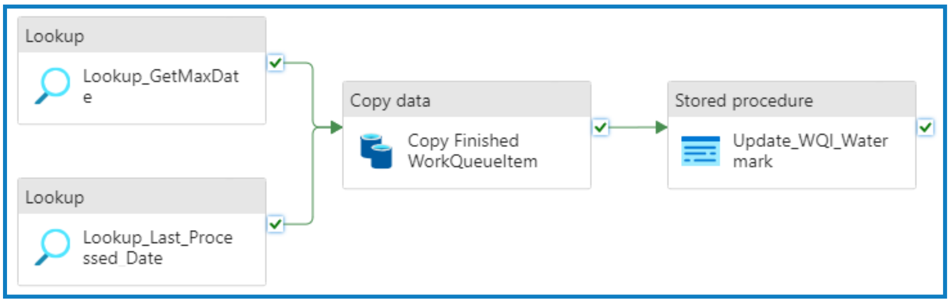
PIPELINE: WORKQUEUEITEM UNFINISHED

This pipeline copies unfinished Work Queue Item data into Azure Data Lake storage. This information is overwritten daily to reflect changes to the unfinished Work Queue Items, whilst finished items are captured by the WorkQueueItem Finished Incremental pipeline.



PIPELINE: WORKQUEUEITEM FINISHED INCREMENTAL

This pipeline copies Work Queue Item finished data into Azure Data Lake storage. This pipeline only copies incremental or changed (delta) data.



It is comprised of two lookup activities:

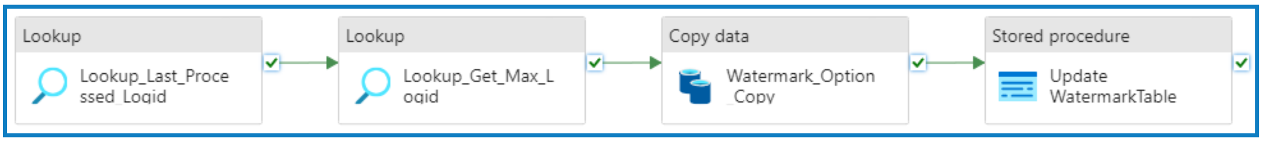
- Lookup_GetMaxDate gets the maximum date of the finished date.
- Lookup_Last_Processed_Date gets the last processed date of the finished date.

The copy activity (Copy Finished WorkQueueItem) only copies the data that is greater than both the last processed date and the maximum finished date. When the copy activity is complete, the watermark table is updated to keep track of the last processed date.

PIPELINE: SESSION LOG INCREMENTAL

This pipeline copies the session log data into Azure Data Lake storage based on the *logid* column. This pipeline is designed to only copy new data since the previous run.

logid is a unique id for each session log row and increments sequentially.

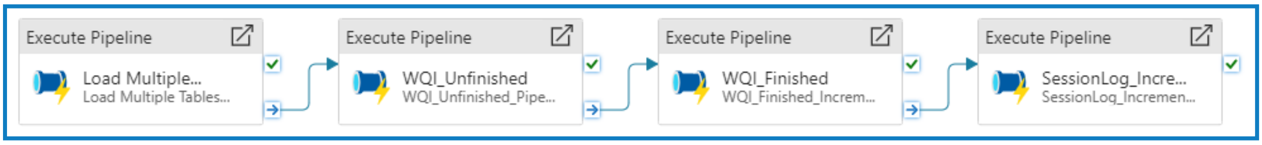


This pipeline looks up the minimum *logid* (via the lookup table) not previously copied into Azure Data Lake and the maximum *logid* that needs to be copied.

After the number of rows to be copied is determined, the data is then copied into Azure Data Lake storage. Once the data has been copied, the lookup table is updated with the maximum copied *logid*. This then becomes the minimum *logid* during the lookup activity the next time the pipeline is run.

PIPELINE: MASTER

This is the Master pipeline (PL_Master) that processes four other pipelines sequentially as shown below.



The pipeline is designed so that the next pipeline will not run until the first or preceding pipeline has been completed. This is to prevent two forEach activities in any of the pipelines running simultaneously.

Triggers

The ADF scheduler consists of configurable Triggers that execute ADF pipelines. A single Trigger named “triggerMasterPipeline” is deployed with the Data Vault solution which aims to execute the wrapper pipeline every day. This is scheduled based on the Blue Prism Database's least busy time at time of deployment. The trigger time is evaluated regularly and adjusted as necessary.

SQL Server Partitioning

Microsoft SQL Server provides a high-level method to maintain large table maintenance known as 'partitioning', which groups (or partitions) large amount of data to make it easier to handle. By implementing this method along with the Data Vault, the Blue Prism session log table’s growth can be strategically maintained while at the same time allowing the old data to be accessed through Azure Data Lake storage. A series of SQL Stored Procedures have been created and are deployed via ARM template. The initial partitioning activity requires a one-time negotiated database maintenance. The weekly partitioning process is executed during the normal scheduled weekly platform reboot.

Azure Data Lake

By default, the Data Lake includes one BLOB container designed to store the Parquet files generated by the archive process. Data Lakes are large repositories of structured, semi-structured, unstructured, and binary data stored in its natural/raw format. Azures’ Data Lake is built on Azure Blob storage (Binary Large Object) and exists as a v2 storage account.

The container takes advantage of a datetime-based namespace hierarchy to enable efficient querying of the contained BLOBs. The hierarchy has been applied in the following way:

- BPASessionLog: /CONTAINERNAME /TABLENAME/YEAR/MONTH/DAY
- All Other Tables: /CONTAINERNAME /TABLENAME

Data Security and Access

The current design relies on the familiar subscription-based segregation of Azure resources and data. Each client is isolated to their individual subscription, having its own set of resources that enable the Data Vault solution.

The Azure Data Factory utilizes its Managed Identity account with explicit permissions on a specified number of tables in the Blue Prism production database. The Azure Data Lake, which is effectively a storage account, also utilizes the Data Factory’s specified Managed Identity for authorizing access.

Customer access to the data is available using a SAS token to the Data Lake. Data BLOBs can also be accessed through the storage account endpoints using SAS authorization, and the Parquet BLOB files can be transformed using the customer’s chosen ETL tooling.

Blue Prism Cloud can also target your own Azure Data Lake storage.

To request access or to use your own Azure Data Lake, please submit a service request using the SS&C | Blue Prism support portal**.**

If you wish to use your own Azure Data Lake, additional details and documentation will be provided upon request.