Configure Offloading Large Data to External Storage

Last Modified on 07/24/2026 7:55 am EDT

V11.3.1

Overview

As a system administrator, you can configure to have large data persisted on external storage instead of database storage. Enabling this configuration reduces database costs, improves query performance, and scales data retention without re-architecting the solution.

The configuration is applied on the application level and is relevant for server-side activities that generate residual internal data, that is generally used only during workflow execution and is rarely read thereafter. For example, the *request* and *response* data that gets generated by consumer activities.

How it works?

Server-side activities may generate large data as part of their execution. The current (and default) system behavior is to save this data in the operational database. The default behavior does not change and is fully backwards compatible.

You can configure, during system deployment, a new behavior mode that saves the data in an external storage. The system attempts to write the data first to the external storage and then add a record in the relevant database table and put NULL in the fields where the data would normally go. The NULL value is used later as a flag to indicate the *read* API where to get the data from. The system reads the data from the database table first, and if the data fields are NULL, then it looks for the data in the external storage.

If saving the data in the external storage fails, the system falls back to the database to save the data in the relevant table. This ensures resilience in case of transient errors in the external storage, while keeping backwards compatibility with existing implementations.

Affected activities

Following are the Consumer activities generating request/response XML data:

- AiAgentActivity
- AzureServiceBusConsumerActivity
- BuiltInCommandActivity
- BuiltInCommandListenerActivity
- CheckInDocumentActivity
- CheckOutDocumentActivity
- CreateListItemActivity
- DeleteListItemActivity
- DynamicInputActivity
- ExecuteRequestActivity
- ExternalServiceConsumerActivity
- HttpConsumerActivity
- HttpFaultActivity
- HttpInputActivity
- HttpOutputActivity
- InProcessServiceConsumerActivity

- KafkaProducerActivity
- MessageBusProducerActivity
- RestServiceConsumerActivity
- RfcConsumerActivity
- SKActivity
- SPConsumerActivity
- UpdateListItemActivity
- UploadDocumentActivity
- WcfServiceConsumerActivity
- WebServiceConsumerActivity
- WebServiceInputActivity
- WebServiceListenerActivity
- WebServiceOutputActivity

How to configure?

The recommended storage type is Azure BLOB or AWS S3, and the approach is like the configuration for File Storage. You can use the same storage that is configured for the File Storge or use a separate storage.

Default configuration, out of the box

```
<configuration>
 ...
 <sequence.engine>
  ...
  <largeDataStorage defaultStorageName="default"  />
  ...
 </sequence.engine>
</configuration>
```

Configuration for external storage

```
<largeDataStorage defaultStorageName="myConsumersStorage">
    <storageProviders>
        <add name="myConsumersStorage"
           connectionType="AzureBlobStorage"
           relativePath="myLargeData"
           connection="%%sequence:consumersExtStorage:fileConnection%%"
           <!--
           connection="BlobEndpoint=https://********.blob.core.windows.net/tester;
                    DefaultEndpointsProtocol=https;
                    AccountName=myaccount;
                    AccountKey=*****;
                    EndpointSuffix=core.windows.net"
           -->
        />
    </storageProviders>
</ largeDataStorage >
```

Configuration for external storage, using *external config file*

```
<configuration>
 ...
 <sequence.engine>
   ...
   <largeDataStorage configSource="Configuration\largeDataExternalStorage.config" />
   ...
 </sequence.engine>
</configuration>
```

In the ~\Configuration\largeDataExternalStorage.config file, the default is:

```
<largeDataStorage defaultStorageName="Default">
    <storageProviders />
</ largeDataStorage >
```