

# Merging Attachments

Attachments are stored in ServiceNow in two separate tables, **sys\_attachment** and **sys\_attachment\_doc**. The **sys\_attachment** record holds information about the attachment (such as the file name and file size), while the **sys\_attachment\_doc** table contains the actual byte data of the attachment separated into different chunks so as to not have one record take up too much space in a table row. Thus, sending attachment (s) to your database with the DataSync Agent will store the attachment (s) in the two tables just as they are stored in ServiceNow.

What's on this page?

- Merging Attachments in the Database
- Saving Locally or to a Cloud
  - Saving Locally
  - Saving to a Cloud

## Merging Attachments in the Database

For [edge encrypted](#) attachments, ServiceNow will encrypt the attachment byte data when the attachment is in one piece and then split up the encrypted content into chunks for saving in the **sys\_attachment\_doc** table. As a result, this content is not usable since each **sys\_attachment\_doc** record is only one portion of the encrypted content.

By including a **SysAttachmentHandler** in the DataSync Agent configuration, the Agent will query the **sys\_attachment** and **sys\_attachment\_doc** tables and merge the attachment records back together into one record. The records are removed from these two tables and the merged attachment records can then be stored in whole in the database (in a new table called **attachments**) or shared out as messages back to the Integration Mesh.

**NOTE:** Storing the attachments as complete records in the database will lead to large table rows. Ensure your database has the necessary storage space to save these large records.

When saving attachments into the database, an **attachments** table gets created to store the fully built attachments and has the following structure:

Column	SQL Data Type*	Description
sys_id	varchar(40)	sys_id of the sys_attachment record
table_name	varchar(128)	Name of the table that the file is attached to
file_name	varchar(128)	Name of the file
content_type	varchar(128)	Type of file
table_sys_id	varchar(32)	sys_id of the record the file is attached to
position	bigint	Will have a value of 0 to represent a full attachment starting at the beginning (0) position
attachment	varchar(max)	Attachment's complete byte data. This column's data can be used to get the whole attachment as one viewable file.

\*Data types shown are for MySQL/SQL Server. Other databases will create columns in their equivalent data types (for example, Oracle will create the position field as a number field data type, its equivalent of bigint).

1

Navigate to the directory where you saved your **agent.xml** file when installing your DataSync Agent.

2

Open your **agent.xml** file in a text editing application. Then, add the following directives under `<share><task>` :

Directive	Description	Required?	Example Value
-----------	-------------	-----------	---------------

<handler>	<p>Name of the Java handler class. In this case, the value for this directive will be <b>com.perspectium.replicator.sql.subscriber.edge.SysAttachmentHandler</b>.</p> <pre>&lt;config&gt;   &lt;agent&gt;     &lt;share&gt;       &lt;task&gt;         ...         &lt;handler&gt;com.perspectium.replicator.sql.subscriber. edge.SysAttachmentHandler&lt;/handler&gt;       &lt;/task&gt;     &lt;/share&gt;   &lt;/agent&gt; &lt;/config&gt;</pre>	Yes	com.perspectium.replicator.sql.subscriber.edge.SysAttachmentHandler																														
<skip_queue />	<p>By default merged attachment files will be published as messages into the Integration Mesh queue specified in the <b>&lt;message_connection&gt;</b> directive in this sharing task.</p> <p>To insert the merged attachment into the <b>attachments</b> table in the database instead, include the <b>&lt;skip_queue/&gt;</b> (no value required). If using <b>&lt;skip_queue/&gt;</b>, then <b>&lt;message_connection&gt;</b> is not required since records will not be published into a queue.</p>	No	If using <b>&lt;message_connection&gt;</b> , then <b>&lt;skip_queue/&gt;</b> is NOT required.																														
<edge_encryption>	<p>Decrypts messages that have been published from a ServiceNow instance that is leveraging <b>Edge Encryption</b>.</p> <p>See <a href="#">Edge Encryption Support</a> for more information.</p> <p>The Agent will need access to the <a href="#">keystore</a> containing the encryption key used for <b>Edge Encryption</b>. This keystore can be stored in an <a href="#">Azure Key Vault</a> cloud key management or stored locally on a filesystem the Agent has access to.</p> <p>For Azure Key Vault, add the following:</p> <table><thead><tr><th>Parameter</th><th>Description</th></tr></thead><tbody><tr><td>keystore</td><td>Specifies where the keystore is located. Value: <b>azure</b></td></tr><tr><td>vault_tenant</td><td>tenant_id for the Azure Key Vault containing the keystore*</td></tr><tr><td>vault_url</td><td>URL to the Azure Key Vault*</td></tr><tr><td>vault_principal</td><td>principal_id for the Azure Key Vault*</td></tr><tr><td>principal_secret</td><td>Password for the Azure Key Vault*</td></tr><tr><td>secret_name</td><td>Name of the keystore*</td></tr><tr><td>keystore_password</td><td>Password for the keystore^</td></tr><tr><td>keystore_alias</td><td>Name of the key alias^</td></tr><tr><td>alias_password</td><td>Password for the key^</td></tr></tbody></table> <p>*See <a href="#">Authentication in Azure Key Vault</a> for more information on these configurations.</p> <p>^See <a href="#">Edge Encryption properties</a> for more information on these configurations.</p> <p><b>NOTE:</b> The Azure KeyVault uses its own HTTP client, which means Azure is making network connections separate from Perspectium messages and instance connections. Thus, to access Azure KeyVault through a proxy, add the following:</p> <table><thead><tr><th>Parameter</th><th>Description</th></tr></thead><tbody><tr><td>proxy</td><td>The URL of the proxy server to connect to.  proxy="http://example.com/"</td></tr><tr><td>proxy_port</td><td>Port number this client will use.  proxy_port="8080"</td></tr><tr><td>proxy_user</td><td>The username to authenticate with the proxy server  proxy_user="proxyConfiguredUser"</td></tr><tr><td>proxy_password</td><td>The password to authenticate with the proxy server  proxy_password="proxyConfiguredUserPassword"</td></tr></tbody></table>	Parameter	Description	keystore	Specifies where the keystore is located. Value: <b>azure</b>	vault_tenant	tenant_id for the Azure Key Vault containing the keystore*	vault_url	URL to the Azure Key Vault*	vault_principal	principal_id for the Azure Key Vault*	principal_secret	Password for the Azure Key Vault*	secret_name	Name of the keystore*	keystore_password	Password for the keystore^	keystore_alias	Name of the key alias^	alias_password	Password for the key^	Parameter	Description	proxy	The URL of the proxy server to connect to.  proxy="http://example.com/"	proxy_port	Port number this client will use.  proxy_port="8080"	proxy_user	The username to authenticate with the proxy server  proxy_user="proxyConfiguredUser"	proxy_password	The password to authenticate with the proxy server  proxy_password="proxyConfiguredUserPassword"	Yes, if the attachments are <b>edge encrypted</b> .  Otherwise, NOT required.	true
Parameter	Description																																
keystore	Specifies where the keystore is located. Value: <b>azure</b>																																
vault_tenant	tenant_id for the Azure Key Vault containing the keystore*																																
vault_url	URL to the Azure Key Vault*																																
vault_principal	principal_id for the Azure Key Vault*																																
principal_secret	Password for the Azure Key Vault*																																
secret_name	Name of the keystore*																																
keystore_password	Password for the keystore^																																
keystore_alias	Name of the key alias^																																
alias_password	Password for the key^																																
Parameter	Description																																
proxy	The URL of the proxy server to connect to.  proxy="http://example.com/"																																
proxy_port	Port number this client will use.  proxy_port="8080"																																
proxy_user	The username to authenticate with the proxy server  proxy_user="proxyConfiguredUser"																																
proxy_password	The password to authenticate with the proxy server  proxy_password="proxyConfiguredUserPassword"																																

```

<config>
  <agent>
    <share>
      <task>
        ...
        <edge_encryption
keystore="azure"
                                proxy="
http://example.com"
                                proxy_port="
8080"
                                vault_tenant="
VAULT_TENANT_GOES_HERE"
                                vault_url="
VAULT_URL_GOES_HERE"
                                vault_principal="
VAULT_PRINCIPAL_GOES_HERE"
                                principal_secret="
PRINCIPAL_SECRET_GOES_HERE"
                                secret_name="
SECRET_NAME_GOES_HERE"
                                keystore_password="
KEYSTORE_PASSWORD_GOES_HERE"
                                keystore_alias="
KEYSTORE_ALIAS_GOES_HERE"
alias_password="ALIAS_PASSWORD_GOES_HERE">
                                true
                                </edge_encryption>
      </task>
    </share>
  </agent>
</config>

```

For a keystore stored in the local file system, add the following:

Parameter	Description
keystore	Specifies where the keystore is located. Value: <b>local</b>
keystore_path	File path to the keystore^
keystore_password	Password for the keystore^
keystore_alias	Name of the key alias^
alias_password	Password for the key^

^See [Edge Encryption properties](#) for more information on these configurations.

```

<config>
  <agent>
    <share>
      <task>
        ...
        <edge_encryption
keystore="local"
  keystore_path="
KEYSTORE_PATH_GOES_HERE"
  keystore_password="
KEYSTORE_PASSWORD_GOES_HERE"
  keystore_alias="
KEYSTORE_ALIAS_GOES_HERE">true</edge_encryption>
      </task>
    </share>
  </agent>
</config>

```

<database\_\*> Since the Agent will need to connect to the database to read records from the **sys\_attachment** and **sys\_attachment\_doc** tables to merge attachments, you will need to configure all <database\_> directives. See [Database configurations](#) for more information.

Yes

<attachment_retries>	Specifies how many times an attachment will be tried to be built until it's skipped and removed from the sys_attachment table. If tag is not include, the default value is 3.	No	3
----------------------	---	----	---

```

<config>
  <agent>
    <share>
      <task>
        ...
        <attachment_retries>3<
      /attachment_retries>
    </task>
  </share>
</agent>
</config>

```

If you are using **<message\_connection>** so the merged attachments are shared to a queue in the Integration Mesh, messages will be created using the directives configured in the share task:

Directive	Description
message_connection	The <b>&lt;message_connection&gt;</b> is used to configure the Integration Mesh queue where messages will be shared to. This configuration is not needed if using <b>&lt;skip_queue&gt;</b> .
encryption_key	A 24+ character key to encrypt the merged attachment's byte data. This is the same approach as entering an encryption key in the Perspectium ServiceNow app for encrypting data shared out of ServiceNow.
topic	The topic for each message shared. If no value specified, <b>replicator</b> will be used.
type	The type for each message shared. If no value specified, <b>sql</b> will be used.
key	The key for each message shared. If no value specified, the task's name will be used.
name	The action for each message shared to append to <b>sys_attachment</b> .  For example, if the value entered is <b>.insert</b> , messages will have a name of <b>sys_attachment.insert</b> .  If no value specified, the name field will be <b>sys_attachment</b> without an action name at the end.
value	<Encrypted Merged Attachment Byte Data>  The data will be encrypted using the encryption key specified in the <b>&lt;encryption_key&gt;</b> directive in the share task

For example, a configuration like this

```

<config>
  <agent>
    <share>
      <task>
        ...
        <encryption_key>some_encryption_key_here</encryption_key>
        <message_connection user="USER" password="PASSWORD" queue="
psp.out.replicator.dev1234">https://URL.perspectium.net</message_connection>
        <topic>replicator</topic>
        <type>agent</type>
        <key>dev1234</key>
        <name>.insert</name>
      </task>
    </share>
  </agent>
</config>

```

Will create messages shared to the Integration Mesh that are as follows:

```

{
  "topic": "replicator",
  "type": "agent",
  "key": "dev1234",
  "name": "sys_attachment.insert",
  "value": "<Encrypted Merged Attachment Data>"
}

```

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<config>
  <agent>
    <max_reads_per_connect>1</max_reads_per_connect>
    <polling_interval>5</polling_interval>
    <test_mode/>
    <subscribe>
      <task>
        <polling_interval>5</polling_interval>
        <task_name>oracle_subscriber_automated_test</task_name>
        <handler>com.perspectium.replicator.sql.SQLSubscriber</handler>
        <decryption_key>some_decryption_key_here</decryption_key>
        <message_connection user="USER" password="PASSWORD" queue="psp.out.replicator.
dev1234">https://URL.perspectium.net</message_connection>
        <use_cache/>
        <instance_connection user="USER" password="PASSWORD">https://dev1234.service-now.
com</instance_connection>
        <database_type>sqlserver</database_type>
        <database_port>1234</database_port>
        <database_server>SERVER_URL</database_server>
        <database_user>USER</database_user>
        <database_password>PASSWORD</database_password>
        <database_parms>lockTimeout=15000;queryTimeout=15</database_parms>
        <database>DATABASE_NAME</database>
        <skip_columns_log_interval>200</skip_columns_log_interval>
        <plugins>
          <plugin keystore="azure"
            vault_tenant="12345678-ab12-ab12-ab12-123456789ab"
            vault_url="https://url.vault.azure.net/"
            vault_principal="12345678-ab12-ab12-ab12-123456789ab"
            principal_secret="3213156165-adasdads_als5d6a5s1d6a"
            secret_name="some_secret_name"
            keystore_password="efg123"
            keystore_alias="128bitkey"
            alias_password="abc123">com.perspectium.replicator.sql.plugin.
SQLSubscriberDecryptColumnPlugin
          </plugin>
        </plugins>
      </task>
    </subscribe>
    <share>
      <task>
        <task_name>attachment_processor</task_name>
        <handler>com.perspectium.replicator.sql.subscriber.edge.SysAttachmentHandler<
/handler>
        <encryption_key>some_encryption_key_here</encryption_key>
        <message_connection user="USER" password="PASSWORD" queue="psp.out.replicator.
dev5678">https://URL.perspectium.net</message_connection>
        <polling_interval>60</polling_interval>
        <max_writes_per_connect>1</max_writes_per_connect>
        <skip_queue/>
        <skip_report/>
        <topic>replicator</topic>
        <type>agent</type>
        <key>dev1234</key>
        <name>.insert</name>
        <database_type>sqlserver</database_type>
        <database_port>1234</database_port>
        <database_server>SERVER_URL</database_server>
        <database_user>USER</database_user>
        <database_password>PASSWORD</database_password>
        <database_parms>lockTimeout=15000;queryTimeout=15</database_parms>
        <database>DATABASE_NAME</database>

        <edge_encryption
          keystore="azure"
          vault_tenant="12345678-ab12-ab12-ab12-123456789ab"
          vault_url="https://url.vault.azure.net/"
          vault_principal="12345678-ab12-ab12-ab12-123456789ab"
          principal_secret="3213156165-adasdads_als5d6a5s1d6a"

```

```

        secret_name="some_secret_name"
        keystore_password="efg123"
        keystore_alias="128bitkey"
        alias_password="abc123">true
    </edge_encryption>

</task>
</share>
</agent>
</config>

```

3

Optionally, you can encrypt the attachments using Voltage API. See [Encryption using Voltage](#) for more information on the plugin.

In the agent.xml, add the <voltage> directive within <share>:

```

<voltage url="https://example.com/api"
  trace_id="TRACE01"
  app_name="PAgent"
  host_name="psp"
  identity=""
  shared_secret="secret"
  encryption_type="3"
  encrypt_all="true">true</voltage>

```

Directive	Description	Value	Attributes		
			Attribute	Description	Required?
<voltage>	Encrypts the fully built attachments using Voltage.	true/false	url	REST url endpoint	Yes
			trace_id	Identifier for each request made to the API	No
			app_name	Name of the application, i.e. Perspectium Agent	Yes
			host_name	Name of the machine running the application	Yes
			identity	Identifier for application using the API	No
			shared_secret	Secret phrase shared between clients using API	Yes
			encryption_type	Encryption method, i.e. AES = 3	Yes
			encrypt_all	If set to true, all attachments will be encrypted with Voltage. Otherwise only those previously encrypted by edge will be encrypted with Voltage	No

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<config>
  <agent>
    <max_reads_per_connect>1</max_reads_per_connect>
    <polling_interval>5</polling_interval>
    <test_mode/>
    <subscribe>
      <task>
        <polling_interval>5</polling_interval>
        <task_name>oracle_subscriber_automated_test</task_name>
        <handler>com.perspectium.replicator.sql.SQLSubscriber</handler>
        <decryption_key>some_decryption_key_here</decryption_key>
        <message_connection user="USER" password="PASSWORD" queue="psp.out.replicator.
dev1234">https://URL.perspectium.net</message_connection>
        <use_cache/>
        <instance_connection user="USER" password="PASSWORD">https://dev1234.service-now.
com</instance_connection>
        <database_type>sqlserver</database_type>
        <database_port>1234</database_port>
        <database_server>SERVER_URL</database_server>
        <database_user>USER</database_user>
        <database_password>PASSWORD</database_password>
        <database_parms>lockTimeout=15000;queryTimeout=15</database_parms>
        <database>DATABASE_NAME</database>
        <skip_columns_log_interval>200</skip_columns_log_interval>
        <plugins>
          <plugin keystore="azure"
            vault_tenant="12345678-ab12-ab12-ab12-123456789ab"

```

```

        vault_url="https://url.vault.azure.net/"
        vault_principal="12345678-ab12-ab12-ab12-123456789ab"
        principal_secret="3213156165-adasdads_als5d6a5s1d6a"
        secret_name="some_secret_name"
                                keystore_password="efgl23"
        keystore_alias="128bitkey"
        alias_password="abc123">com.perspectium.replicator.sql.plugin.
SQLSubscriberDecryptColumnPlugin
                                </plugin>
        </plugins>

    </task>
</subscribe>
<share>
    <task>
        <task_name>attachment_processor</task_name>
        <handler>com.perspectium.replicator.sql.subscriber.edge.SysAttachmentHandler<
/handler>
        <encryption_key>some_encryption_key_here</encryption_key>
        <message_connection user="USER" password="PASSWORD" queue="psp.out.replicator.
dev5678">https://URL.perspectium.net</message_connection>
        <polling_interval>60</polling_interval>
        <max_writes_per_connect>1</max_writes_per_connect>
        <skip_queue/>
        <skip_report/>
        <topic>replicator</topic>
        <type>agent</type>
        <key>dev1234</key>
        <name>.insert</name>
        <database_type>sqlserver</database_type>
        <database_port>1234</database_port>
        <database_server>SERVER_URL</database_server>
        <database_user>USER</database_user>
        <database_password>PASSWORD</database_password>
        <database_parms>lockTimeout=15000;queryTimeout=15</database_parms>
        <database>DATABASE_NAME</database>
        <edge_encryption
                                keystore="azure"
                                vault_tenant="12345678-ab12-ab12-ab12-123456789ab"
                                vault_url="https://url.vault.azure.net/"
                                vault_principal="12345678-ab12-ab12-ab12-123456789ab"
                                principal_secret="3213156165-adasdads_als5d6a5s1d6a"
                                secret_name="some_secret_name"
                                keystore_password="efgl23"
                                keystore_alias="128bitkey"
                                alias_password="abc123">true
        </edge_encryption>
        <voltage url="https://example.com/api"
                                trace_id="TRACENUM"
                                app_name="PAgent"
                                host_name="psp"
                                identity=""
                                shared_secret="secret"
                                encryption_type="3"
                                encrypt_all="true">true
        </voltage>
    </task>
</share>
</agent>
</config>

```

# Saving Locally or to a Cloud

By including a **SysAttachmentHandler** in the DataSync Agent configuration, the Agent will query the **sys\_attachment** and **sys\_attachment\_doc** tables and construct the full attachment to be saved locally or to a cloud storage service such as Amazon S3, Azure Blob Storage, or Google Cloud Storage. The records are then removed from these two tables.

When adding a **SysAttachmentHandler**, all database configurations must match the database configurations from the subscribe task to ensure the correct `sys_attachment` and `sys_attachment_doc` tables are being queried and processed

## Saving Locally

When saving locally, you will need the following directives under `<share><task>`:

Directive	Description	Required?
<code>&lt;destination&gt;</code>	Final location of the attachment (local, S3, GCS, ABS)	yes
<code>&lt;directory&gt;</code>	Where the attachment will be saved locally	yes



```

<?xml version="1.0" encoding="UTF-8"?>
<config>
  <agent>
    <max_reads_per_connect>1</max_reads_per_connect>
    <polling_interval>5</polling_interval>
    <skip_message_set_processing />
    <test_mode />
    <subscribe>
      <task>
        <task_name>oracle_subscriber_automated_test</task_name>
        <handler>com.perspectium.replicator.sql.SQLSubscriber</handler>
        <encryption_key>The cow jumped over the moon</encryption_key>
        <message_connection user="demo" password="example_pass" queue="psp.out.replicator.agent_example"
>https://example.perspectium.net</message_connection>
        <use_cache />
        <instance_connection user="admin" password="example_pass">https://example.service-now.com<
/instance_connection>
        <database_server />
        <database_user>psptest</database_user>
        <database_password />
        <database_port>1433</database_port>
        <database_type>sqlserver</database_type>
        <database>psp_repl</database>
        <skip_columns_log_interval>200</skip_columns_log_interval>
      </task>
    </subscribe>
    <share>
      <task>
        <task_name>attachment_processor</task_name>
        <handler>com.perspectium.replicator.sql.subscriber.edge.SysAttachmentHandler</handler>
        <encryption_key>The cow jumped over the moon twice in 2021</encryption_key>
        <message_connection user="demo" password="example_pass" queue="psp.out.replicator.
agent_attachments">https://example.perspectium.net</message_connection>
        <polling_interval>60</polling_interval>
        <max_writes_per_connect>1</max_writes_per_connect>
        <skip_queue />
        <skip_report />
        <topic>test</topic>
        <type>test</type>
        <key>mac</key>
        <name />
        <database_server />
        <database_user>psptest</database_user>
        <database_password />
        <database_port>1433</database_port>
        <database_type>sqlserver</database_type>
        <database>psp_repl</database>
        <attachment_storage>
          <directory>/Users/demo/Desktop/TestAttachments</directory>
          <destination>local</destination>
        </attachment_storage>
      </task>
    </share>
  </agent>
</config>

```

## Saving to a Cloud

When saving to a cloud service, you will need the following directives under **<share><task>**:

### Amazon S3

Directive	Description	Required?
-----------	-------------	-----------

<b>&lt;destination&gt;</b>	Final location of the attachment (local, S3, GCS, ABS)	Yes
<b>&lt;directory&gt;</b>	Where the attachment will be saved locally	Yes
<b>&lt;access_key&gt;</b>	Access Key associated with your AWS account	Yes, <b>ONLY</b> when the DataSync Agent is not installed on an EC2 instance.
<b>&lt;region&gt;</b>	Region that your AWS S3 bucket resides in  <b>NOTE:</b> If a region is specified for an Agent on an EC2 instance, then the region must match the region of the EC2 instance and the region of the S3 bucket	No
<b>&lt;bucket_name&gt;</b>	Name of your AWS S3 bucket, including subdirectories if desired, to specify where the records will be uploaded e.g. bucketName/folder1/folder2.  For example, with <bucket_name>psp-bucket</bucket_name> will save records into the psp-bucket S3 bucket.	Yes
<b>&lt;secret_access_key&gt;</b>	Secret Access Key associated with your AWS account	Yes, <b>ONLY</b> when the DataSync Agent is not installed on an EC2 instance.

```

<?xml version="1.0" encoding="UTF-8"?>
<config>
  <agent>
    <max_reads_per_connect>1</max_reads_per_connect>
    <polling_interval>5</polling_interval>
    <skip_message_set_processing />
    <test_mode />
    <subscribe>
      <task>
        <task_name>oracle_subscriber_automated_test</task_name>
        <handler>com.perspectium.replicator.sql.SQLSubscriber</handler>
        <decryption_key>The cow jumped over the moon</decryption_key>
        <message_connection user="demo" password="example_pass" queue="psp.out.replicator.agent_example"
>https://example.perspectium.net</message_connection>
        <use_cache />
        <instance_connection user="admin" password="example_pass">https://example.service-now.com<
/instance_connection>
        <database_server />
        <database_user>psptest</database_user>
        <database_password />
        <database_port>1433</database_port>
        <database_type>sqlserver</database_type>
        <database>psp_repl</database>
        <skip_columns_log_interval>200</skip_columns_log_interval>
      </task>
    </subscribe>
    <share>
      <task>
        <task_name>attachment_processor</task_name>
        <handler>com.perspectium.replicator.sql.subscriber.edge.SysAttachmentHandler</handler>
        <encryption_key>The cow jumped over the moon twice in 2021</encryption_key>
        <message_connection user="demo" password="example_pass" queue="psp.out.replicator.
agent_attachments">https://example.perspectium.net</message_connection>
        <polling_interval>60</polling_interval>
        <max_writes_per_connect>1</max_writes_per_connect>
        <skip_queue />
        <skip_report />
        <topic>test</topic>
        <type>test</type>
        <key>mac</key>
        <name />
        <database_server />
        <database_user>psptest</database_user>
        <database_password />
        <database_port>1433</database_port>
        <database_type>sqlserver</database_type>
        <database>psp_repl</database>
        <attachment_storage>
          <directory>/Users/demo/Desktop/TestAttachments</directory>
          <destination>S3</destination>
          <access_key>EXAMPLEACCESSKEY</access_key>
          <secret_access_key>EXAMPLESECRETACCESSKEY</secret_access_key>
          <region>us-east-2</region>
          <bucket_name>examplebucket</bucket_name>
        </attachment_storage>
      </task>
    </share>
  </agent>
</config>

```

## Azure Blob Storage

Directive	Description	Required?
<destination>	Final location of the attachment (local, S3, GCS, ABS)	Yes
<directory>	Where the attachment will be saved locally	Yes

<b>&lt;connection_string&gt;</b>	String provided by Azure for connecting and uploading to the blob storage	Yes
<b>&lt;bucket_name&gt;</b>	Name of your Azure Blob Storage container, including subdirectories if desired, to specify where the records will be uploaded e.g. bucketName/folder1/folder2.  For example, with <bucket_name>psp-bucket</bucket_name> will save records into the psp-bucket Azure Blob Storage container.	Yes

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
  <agent>
    <max_reads_per_connect>1</max_reads_per_connect>
    <polling_interval>5</polling_interval>
    <skip_message_set_processing />
    <test_mode />
    <subscribe>
      <task>
        <task_name>oracle_subscriber_automated_test</task_name>
        <handler>com.perspectium.replicator.sql.SQLSubscriber</handler>
        <decryption_key>The cow jumped over the moon</decryption_key>
        <message_connection user="demo" password="example_pass" queue="psp.out.replicator.agent_example">https://example.perspectium.net</message_connection>
        <use_cache />
        <instance_connection user="admin" password="example_pass">https://example.service-now.com</instance_connection>
        <database_server />
        <database_user>psptest</database_user>
        <database_password />
        <database_port>1433</database_port>
        <database_type>sqlserver</database_type>
        <database>psp_repl</database>
        <skip_columns_log_interval>200</skip_columns_log_interval>
      </task>
    </subscribe>
    <share>
      <task>
        <task_name>attachment_processor</task_name>
        <handler>com.perspectium.replicator.sql.subscriber.edge.SysAttachmentHandler</handler>
        <encryption_key>The cow jumped over the moon twice in 2021</encryption_key>
        <message_connection user="demo" password="example_pass" queue="psp.out.replicator.agent_attachments">https://example.perspectium.net</message_connection>
        <polling_interval>60</polling_interval>
        <max_writes_per_connect>1</max_writes_per_connect>
        <skip_queue />
        <skip_report />
        <topic>test</topic>
        <type>test</type>
        <key>mac</key>
        <name />
        <database_server />
        <database_user>psptest</database_user>
        <database_password />
        <database_port>1433</database_port>
        <database_type>sqlserver</database_type>
        <database>psp_repl</database>
        <attachment_storage>
          <directory>/Users/demo/Desktop/TestAttachments</directory>
          <destination>ABS</destination>
          <bucket_name>examplebucket</bucket_name>
        </attachment_storage>
        <connection_string>DefaultEndpointsProtocol=https;AccountName=pspuploadtest;AccountKey=Ju6z+w5TZDQ6xGR9SlMnGfAPSKR4lC2g/YD23u+EmiTDe77qU20NOGjXnE9wTgoI095BU97QW3p3iJC5kpv1kA==;EndpointSuffix=core.windows.net</connection_string>
      </task>
    </share>
  </agent>
</config>
```

## Google Cloud Storage

Directive	Description	Required?
<destination>	Final location of the attachment (local, S3, GCS, ABS)	Yes
<directory>	Where the attachment will be saved locally	Yes
<key_path>	JSON file provided by Google for connecting and uploading to the cloud	Yes
<project_id>	Identifier of the Google project containing the bucket	Yes
<bucket_name>	<p>Name of your Google Cloud Storage bucket, including subdirectories if desired, to specify where the records will be uploaded e.g. bucketName/folder1/folder2.</p> <p>For example, with &lt;bucket_name&gt;psp-bucket&lt;/bucket_name&gt; will save records into the psp-bucket Google Cloud Storage bucket</p>	Yes

```

<?xml version="1.0" encoding="UTF-8"?>
<config>
  <agent>
    <max_reads_per_connect>1</max_reads_per_connect>
    <polling_interval>5</polling_interval>
    <skip_message_set_processing />
    <test_mode />
    <subscribe>
      <task>
        <task_name>oracle_subscriber_automated_test</task_name>
        <handler>com.perspectium.replicator.sql.SQLSubscriber</handler>
        <decryption_key>The cow jumped over the moon</decryption_key>
        <message_connection user="demo" password="example_pass" queue="psp.out.replicator.agent_example"
>https://example.perspectium.net</message_connection>
        <use_cache />
        <instance_connection user="admin" password="example_pass">https://example.service-now.com<
/instance_connection>
        <database_server />
        <database_user>psptest</database_user>
        <database_password />
        <database_port>1433</database_port>
        <database_type>sqlserver</database_type>
        <database>psp_repl</database>
        <skip_columns_log_interval>200</skip_columns_log_interval>
      </task>
    </subscribe>
    <share>
      <task>
        <task_name>attachment_processor</task_name>
        <handler>com.perspectium.replicator.sql.subscriber.edge.SysAttachmentHandler</handler>
        <encryption_key>The cow jumped over the moon twice in 2021</encryption_key>
        <message_connection user="demo" password="example_pass" queue="psp.out.replicator.
agent_attachments">https://example.perspectium.net</message_connection>
        <polling_interval>60</polling_interval>
        <max_writes_per_connect>1</max_writes_per_connect>
        <skip_queue />
        <skip_report />
        <topic>test</topic>
        <type>test</type>
        <key>mac</key>
        <name />
        <database_server />
        <database_user>psptest</database_user>
        <database_password />
        <database_port>1433</database_port>
        <database_type>sqlserver</database_type>
        <database>psp_repl</database>
        <attachment_storage>
          <directory>/Users/demo/Desktop/TestAttachments</directory>
          <destination>GCS</destination>
          <key_path>/Users/demo/Downloads/gcs-credentials.json</key_path>
          <project_id>development-174017</project_id>
          <bucket_name>examplebucket</bucket_name>
        </attachment_storage>
      </task>
    </share>
  </agent>
</config>

```