

Setting up DataSync Agent to share to Files

Fluorine+

The [DataSync Agent](#) can be set up to replicate table records from a Service Now instance to local file(s) on the machine where the DataSync Agent is running. This can be useful for the case where you have a separate application that is able to read files to import data.

Records can be saved in CSV, JSON, and XML formats and each record is inserted into the file (i.e. the previous version is not updated) when the Agent processes a message.

Prerequisites

- ⚠ First, you will need to set up one of the Perspectium DataSync Agents.
- ⚠ You should also stop running your DataSync Agent before making any Agent configuration changes.

Procedure

To enable file replication for the DataSync Agent, follow these steps:

1

Access your agent.xml configuration file

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

2

Edit the agent.xml file with the following configuration changes

Within the **<task>** tag, nest the following directives with your choice of how you want to save your records:

[All Records in One File](#)

[One Record per File](#)

[Records to Multiple Files](#)

All Records in One File

If you want to save all records in one file, use the following directives:

Directive	Example	Use	Required?								
<handler>	<handler>com.perspectium.replicator.file.XMLFileSubscriber</handler>	The name of the file handler class	Yes								
	<table><tr><th>File Type</th><th>Value</th></tr><tr><td>CSV</td><td>com.perspectium.replicator.file.CSVFileSubscriber</td></tr><tr><td>JSON</td><td>com.perspectium.replicator.file.JSONFileSubscriber</td></tr><tr><td>XML</td><td>com.perspectium.replicator.file.XMLFileSubscriber</td></tr></table>	File Type	Value	CSV	com.perspectium.replicator.file.CSVFileSubscriber	JSON	com.perspectium.replicator.file.JSONFileSubscriber	XML	com.perspectium.replicator.file.XMLFileSubscriber		
File Type	Value										
CSV	com.perspectium.replicator.file.CSVFileSubscriber										
JSON	com.perspectium.replicator.file.JSONFileSubscriber										
XML	com.perspectium.replicator.file.XMLFileSubscriber										
	 NOTE: Invalid JSON messages, such as contents not properly escaped, will be skipped. An error log will appear when this occurs.										
<file_name>	<file_name>records.csv</file_name>	The name of the file to which you want to save the records	Yes								

<files_directory>	For Linux: <pre><files_directory>/Downloads/subscribefiles/</files_directory></pre> For Windows: <pre><files_directory>Users\Downloads\subscribefiles\</files_directory></pre>	The directory that contains the file of the saved records NOTE: It is important to have the slash at the end of the directory path entered (/ for Linux and \ for Windows) i.e. /Downloads/subscribefiles/	Yes
<buffered_writes>	<code><buffered_writes>250</buffered_writes></code>	A number of records to buffer before writing to file (to improve performance and not write to the file upon reading each record)	No
<exclude_xml_header>	<code><exclude_xml_header/></code>	For use with the XMLFileSubscriber handler, this will only output the xml header tag (i.e. <?xml version="1.0" encoding="UTF-8"?>) once at the top of the file. That way you can treat the entire file as one XML file with multiple elements for parsing. For example, with this configuration, the file will be: <pre><?xml version="1.0" encoding="UTF-8"?> <incident></incident> <incident></incident> <cmdb_ci></cmdb_ci></pre> versus <pre><?xml version="1.0" encoding="UTF-8"?><incident></incident> <?xml version="1.0" encoding="UTF-8"?><incident></incident> <?xml version="1.0" encoding="UTF-8"?><cmdb_ci></cmdb_ci></pre>	No

One Record Per File

If you want to save one record per file, use the following directives:

Directive	Example	Use	Required?								
<handler>	<pre><handler>com.perspectium.replicator.file.XMLFileSubscriber</handler></pre> <table><tr><th>File Type</th><th>Value</th></tr><tr><td>CSV</td><td>com.perspectium.replicator.file.CSVFileSubscriber</td></tr><tr><td>JSON</td><td>com.perspectium.replicator.file.JSONFileSubscriber</td></tr><tr><td>XML</td><td>com.perspectium.replicator.file.XMLFileSubscriber</td></tr></table> NOTE: Invalid JSON messages, such as contents not properly escaped, will be skipped. An error log will appear when this occurs.	File Type	Value	CSV	com.perspectium.replicator.file.CSVFileSubscriber	JSON	com.perspectium.replicator.file.JSONFileSubscriber	XML	com.perspectium.replicator.file.XMLFileSubscriber	The name of the file handler class	Yes
File Type	Value										
CSV	com.perspectium.replicator.file.CSVFileSubscriber										
JSON	com.perspectium.replicator.file.JSONFileSubscriber										
XML	com.perspectium.replicator.file.XMLFileSubscriber										
<one_record_per_file>	<pre><one_record_per_file/></pre>	This directive will tell the agent to save each record into its own file instead of saving all records together in a single file.	Yes								
<files_directory>	For Linux : <pre><files_directory>/Downloads/subscribefiles/</files_directory></pre> enable_audit_logFor Windows : <pre><files_directory>Users\Downloads\subscribefiles\</files_directory></pre>	The directory that contains the file of the saved records NOTE: It is important to have the slash at the end of the directory path entered (/ for Linux and \ for Windows) i.e. /Downloads/subscribefiles/	Yes								
<file_prefix>	<pre><file_prefix>record</file_prefix></pre>	A prefix for the file name of each record. If this directive is not specified, "psp.replicator." will be used as the prefix.	No								
<file_suffix>	<pre><file_suffix>.xml</file_suffix></pre>	A suffix for the file name of each record. If this directive is not specified, ".xml" will be used as the suffix.	No								
<translate_newline>	<pre><translate_newline>\n</translate_newline></pre>	This directive will replace record content newline entries with a non-breaking space.	No								

Records to Multiple Files

If you want to save your records to multiple files, use the following directives:

Directive	Example	Use	Required?
-----------	---------	-----	-----------

<handler>	<handler>com.perspectium.replicator.file.XMLFileSubscriber</handler>	The name of the file handler class	Yes								
	<table><tr><th>File Type</th><th>Value</th></tr><tr><td>CSV</td><td>com.perspectium.replicator.file.CSVFileSubscriber</td></tr><tr><td>JSON</td><td>com.perspectium.replicator.file.JSONFileSubscriber</td></tr><tr><td>XML</td><td>com.perspectium.replicator.file.XMLFileSubscriber</td></tr></table> NOTE: Invalid JSON messages, such as contents not properly escaped, will be skipped. An error log will appear when this occurs.	File Type	Value	CSV	com.perspectium.replicator.file.CSVFileSubscriber	JSON	com.perspectium.replicator.file.JSONFileSubscriber	XML	com.perspectium.replicator.file.XMLFileSubscriber		
File Type	Value										
CSV	com.perspectium.replicator.file.CSVFileSubscriber										
JSON	com.perspectium.replicator.file.JSONFileSubscriber										
XML	com.perspectium.replicator.file.XMLFileSubscriber										
<buffered_writes>	<buffered_writes>250</buffered_writes>	A number of records to buffer before writing to file (to improve performance and not write to the file upon reading each record)	Yes								
		NOTE: The maximum number of records in a file will be configured in this directory.									
<files_directory>	For Linux: <files_directory>/Downloads/subscribefiles</files_directory> For Windows: <files_directory>Users\Downloads\subscribefiles</files_directory>	The directory that contains the file of the saved records	Yes								
		NOTE: It is important to have the slash at the end of the directory path entered ('/' for Linux and '\' for Windows) i.e. /Downloads/subscribefiles/									
<file_prefix>	<file_prefix>record</file_prefix> NOTE: Use the value \$table_\$d{yyyyMMdd}_\$i to set a dynamic file name where table will be the record's table, yyyyMMdd will be the time format, and i will be the file number, i.e. problem_20200530_1.json . You can modify yyyyMMdd with other time format of your choice. For example, hourly will need a yyyyMMddHH value. For other time format, see Date Format . <file_prefix>\$table_\$d{yyyyMMdd}_\$i</file_prefix>	A prefix for the file name of each record. If this directive is not specified, "psp.replicator." will be used as the prefix. NOTE: The time period will be configured in this directory.	No								
<file_suffix>	<file_suffix>.xml</file_suffix> <table><tr><th>File Type</th><th>Value</th></tr><tr><td>CSV</td><td>.csv</td></tr><tr><td>JSON</td><td>.json</td></tr><tr><td>XML</td><td>.xml</td></tr></table>	File Type	Value	CSV	.csv	JSON	.json	XML	.xml	A suffix for the file name of each record. If this directive is not specified, ".xml" will be used as the suffix.	No
File Type	Value										
CSV	.csv										
JSON	.json										
XML	.xml										
<separate_files>	<separate_files>table</separate_files>	Indicates that the files will be separated by table.	Yes								
<enable_audit_log>	<enable_audit_log/>	A self-closing directive that will generate an audit file. The audit file has information about when the records are processed, name of the file, and number of records processed. Audit files will be created in the directory as specified with the <files_directory> directive and saved in the format of: audit_<tablename>_<timestamp>.json Where <tablename> is the name of table that this audit file contains logs for (such as incident) and <timestamp> is the timestamp format as specified in <file_prefix> . If not timestamp is specified, then it uses the default format of yyyy-MM-dd_HH:mm:ss .	No								

<enable_error_log>	<enable_error_log>	Enabling this directive will create an error audit file to store all invalid JSON records. Like audit files, error files will be created in the directory as specified with the <files_directory> directive and saved in the format of: error_<tablename>_<timestamp>.json Where <tablename> is the name of table that this audit file contains logs for (such as incident) and <timestamp> is the timestamp format as specified in <file_prefix>. If not timestamp is specified, then it uses the default format of yyyy-MM-dd_HHmss. In addition, it keeps track of the number of invalid records and saves the error count in the generated audit file.	No
<translate_newline>	<translate_newline>%13</translate_newline>	This directive will replace record content newline entries with the value you input.	Varies
<file_max_size>	<file_max_size>50KB</file_max_size>  NOTE: - The size can be in KB, MB, or GB, i.e. 50KB, 250MB, 1GB. Make sure to have NO space in between the number and the unit. - The minimum value for this directive is 25KB. If you input a value less than 25KB, the value will be set to 25KB. - Conversely, the maximum value for this directive is 2GB. If you input a value greater than 2GB, the value will be set to 2GB.	Sets the maximum size for each file. Once the maximum size has been reached, a new file will be created using the current timestamp as specified in the <file_prefix> directive.  NOTE: If the next record to be saved will cause the file to be over the max size specified, then the current file is closed and a new file is created. For example, if <file_max_size> is 100KB, the current file size is 99KB and the next record is 2KB, then the current file will be closed at 99KB and a new file will be created to hold this next record.	No
<close_file_interval>	<close_file_interval>180</close_file_interval>	Sets the amount of time the Agent will wait for tmp files to be modified. When the DataSync Agent is creating the file, the file name will have a prefix "tmp" in front to indicate that the file is writing records until file_max_size or buffered_writes is reached. Once the DataSync Agent finishes creating the file due to file_max_size or buffered_writes reached, the prefix "tmp" will be removed from the file name. When the tmp file hasn't been modified for certain amount of time, the "tmp" prefix will be removed from the filename and the file will be closed.	No

3

Save your agent.xml

Save the changes you've made to your **agent.xml** and close the file.

An example **agent.xml** configuration for saving all records to a single file is shown below:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<config>
  <agent>
    <max_reads_per_connect>10</max_reads_per_connect>
    <polling_interval>20</polling_interval>
    <subscribe>
      <task>
        <task_name>file_subscribe</task_name>
        <message_connection password="password" user="user"
>https://<customer>.perspectium.net</message_connection>
        <instance_connection password="password" user="user">https://<instance>.
service-now.com</instance_connection>
        <decryption_key>The cow jumped over the moon</decryption_key>
        <handler>com.perspectium.replicator.file.XMLFileSubscriber</handler>

        <file_name>records.xml</file_name>
        <files_directory>Users/user/Downloads/<
/ files_directory>
        <exclude_xml_header/>
        <buffered_writes>250</buffered_writes>

        <enable_error_log/>
      </task>
    </subscribe>
  </agent>
</config>
```

An example **agent.xml** configuration for saving one record per file is shown below:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<config>
  <agent>
    <max_reads_per_connect>10</max_reads_per_connect>
    <polling_interval>20</polling_interval>
    <subscribe>
      <task>
        <task_name>file_subscribe</task_name>
        <message_connection password="password" user="user"
>https://<customer>.perspectium.net</message_connection>
        <instance_connection password="password" user="user">https://<instance>.
service-now.com</instance_connection>
        <decryption_key>The cow jumped over the moon</decryption_key>
        <handler>com.perspectium.replicator.file.XMLFileSubscriber</handler>

        <one_record_per_file/>
        <files_directory>/tmp/<
/files_directory>
        <file_prefix>records</file_prefix>
        <file_suffix>.xml</file_suffix>

        <enable_error_log/>
      </task>
    </subscribe>
  </agent>
</config>
```

An example **agent.xml** configuration for saving records to multiple files is shown below:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<config>
  <agent>
    <!-- the following subscribe fragment defines subscribing class -->
    <!-- and its arguments -->
    <subscribe>
      <task>
        <task_name>test_file_subscriber</task_name>
        <message_connection password="password_here" user="admin" queue="psp.in.
meshlet.example">https://<customer>.perspectium.net</message_connection>
        <instance_connection password="Adminadmin1" user="admin"
>https://<instance>.service-now.com</instance_connection>
        <decryption_key>Example_decryption_key_here</decryption_key>
        <handler>com.perspectium.replicator.file.JSONFileSubscriber<
/handler>

        <buffered_writes>10</buffered_writes>
        <files_directory>/Users/You/Downloads/Example/<
/files_directory>
        <file_prefix>$table_${yyyyMMdd}_${i}</file_prefix>
        <file_suffix>.json</file_suffix>
        <file_max_size>50KB</file_max_size>
        <translate_newline>%13</translate_newline>
        <separate_files>table</separate_files>
        <enable_audit_log/>
        <enable_error_log/>
        <close_file_interval>180</close_file_interval>
      </task>
    </subscribe>
    <max_reads_per_connect>1</max_reads_per_connect>
    <polling_interval>3</polling_interval>
  </agent>
</config>
```

Restart your DataSync Agent

After configuring your **agent.xml** file to enable file replication, [start running your DataSync Agent](#) again.

Similar topics

- [Get started with the DataSync for ServiceNow Agent](#)
- [Start/stop the DataSync for ServiceNow Agent](#)
- [Monitoring DataSync Agent logs](#)
- [Set up DataSync Agent to share to PostgreSQL](#)
- [Set up DataSync Agent to share to HP Vertica](#)

Contact Perspective Support



US: **1 888 620 8880**

UK: **44 208 068 5953**

support@perspectium.com