

# Set up temporal data

Gold

To compile timestamped "snapshots" of your ServiceNow data, you can configure your DataSync Agent with the **<temporal>** directive. **<temporal>** will allow you to indicate times when your record's data is/was **valid from** and when the data is/was **valid to**.

**NOTE:** To set up a temporal data, the table you are syncing data to must not contain any records (i.e., Temporal data cannot be captured for tables that are already being synced to a database with a DataSync Agent).



**WARNING!** If you have already configured an integration with a DataSync Agent and have been saving records in a database, but you would now like to **enable** or **disable** temporal replication, update the value within the **agent.xml**'s **<database>** directive to a new database. Otherwise records will not be processed properly.

## Prerequisites

**!** First, you will need to install a DataSync Agent and [create a ServiceNow dynamic share](#) with an **update** trigger or [create a ServiceNow bulk share](#).

## Procedure

To set up a temporal data in a local database, 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

### Add the <temporal> directive

Open your **agent.xml** file in a text editing application. To use the default configurations for syncing temporal data (columns will be named **psp\_valid\_from** and **psp\_valid\_to** and the maximum **psp\_valid\_to** value will be **9999-01-01 01:01:01**, and (in **Gold 5.0.3.3**) the archived **psp\_valid\_to** value will be **8888-01-01 01:01:01**), add the **<temporal/>** self-closing tag anywhere within the **<task>** tag. An example of this **agent.xml** is shown below. Otherwise, to create custom configurations for your temporal database, see **Step #3**.

**Example agent.xml for default temporal data configurations:**

```

<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<config>
  <agent>
    <share/>
    <subscribe>
      <task instances="4">
        <task_name>test_subscribe</task_name>
        <message_connection password="encrypted:vl0tU7lyu8N
/EFIJH85SSBtaIt7qEEfvqiqft9VZyYE=" queue="psp.out.replicator.test" use_basic_consume="true"
user="admin">amqps://test.perspectium.net</message_connection>
        <instance_connection password="encrypted:vl0tU7lyu8N/EFIJH85SSPN9aF0P5
/YViVwPEVFcGW4=" user="admin">https://mycompany.service-now.com</instance_connection>
        <handler>com.perspectium.replicator.sql.SQLSubscriber</handler>
        <decryption_key>This is my decryption key for testing</decryption_key>
        <database_type>mysql</database_type>
        <database_server>localhost</database_server>
        <database_port>3306</database_port>
        <database_user>testuser</database_user>
        <database_password>testpassword</database_password>
        <database_parms>characterEncoding=UTF-8 & useSSL=false</database_parms>
        <database_column_max_size>251</database_column_max_size>
        <database>psp_repl</database>
          <archive_datetime>2050-12-31 00:00:00</archive_datetime>
        <temporal>
          <columns>
            <column column_type="93" column_size="32" type="from" primary_key="
false">psp_from</column>
            <column column_type="93" column_size="32" type="to">psp_to</column>
          </columns>
          <max_datetime>2030-12-31 00:00:00</max_datetime>
          <timestamp_difference>5</timestamp_difference>
        </temporal>
      </task>
    </subscribe>
    <max_reads_per_connect>4000</max_reads_per_connect>
    <polling_interval>5</polling_interval>
    <skip_message_set_processing/>
  </agent>
</config>

```

### 3

## Create custom configurations (optional)

Directive	Attribute(s)	Description
<max_datetime>		<p>To customize your temporal database table's maximum date &amp; time stamp in the <b>psp_valid_to</b> column, add the directive shown below within the <b>&lt;temporal&gt;</b> tag, replacing the <b>red text</b> with any value you would like to assign for custom configuration.</p> <pre>&lt;max_datetime&gt;2030-12-31 00:00:00&lt;/max_datetime&gt;</pre> <p><b>NOTE:</b> If records already exist in your temporal database table and then the <b>&lt;maxdatetime&gt;</b> value is changed, any records that were previously timestamped in the <b>psp_valid_to</b> column will not be updated to reflect the newly entered <b>&lt;max_datetime&gt;</b> value.</p>
<archive_datetime />		<p>Available in version 5.0.3.3</p> <p>Customize your temporal database table's archive date and time stamp in the <b>psp_valid_to</b> column.</p> <pre>&lt;archive_datetime&gt;2050-12-31 00:00:00&lt;/archive_datetime&gt;</pre> <p>Replace the date and time stamp with any value you would like to assign for custom configuration.</p> <p>The date and time stamp is applied to temporal rows when the agent consumes <b>.archive</b> messages. This marks the row as "archived" and helps with querying for archived records in the database tables.</p>

<disable_archive/>		<p>Available in version 5.0.3.3</p> <p>This will disable &lt;archive_datetime/&gt; functionality and skip processing .archive messages</p>
<column>	type="from"  type="to"	You can customize the column names for the <b>psp_valid_from</b> and <b>psp_valid_to</b> columns by modifying the values within the <column type="from"> and <column type="to"> directives, respectively.
<column>	primary_key="true"  primary_key="false"	You can customize which temporal columns are primary keys by modifying the values within <column primary_key="true">. By default, both temporal columns are primary keys. However, they both cannot be non-primary keys, meaning both columns cannot have <column primary_key="false">. This is so that it can still insert temporal entries into the database and the database does not complain about duplicate entries.
<timestamp_difference>		<p>You can customize a time period between temporal entries by adding directive &lt;timestamp_difference&gt;1&lt;/timestamp_difference&gt; inside &lt;temporal&gt; and set a number. The value will be read as seconds, so if you want 1 microsecond, you would use 0.000001. The smallest it can go is nanoseconds. By default, the time period is set to 1 millisecond to keep times between entries in order.</p> <p><b>⚠ WARNING:</b> It is not recommended to input a large number in &lt;timestamp_difference&gt;. Doing so would cause the timestamped entries to not be in order as it relies on when the record is processed.</p>
<skip_duplicates/>		You can choose to skip inserting duplicate entries by adding directive <skip_duplicates/> inside <temporal>. Note that this may affect the agent's performance as it checks with the database and does a comparison every time a message comes in.
<skip_duplicates/>	check_all_fields="true"	<p>Available in version 5.0.3.1</p> <p>Check the fields values if the updated time is the same from the previous share.</p> <pre>&lt;temporal&gt;   &lt;skip_duplicates check_all_fields="true" /&gt; &lt;/temporal&gt;</pre>
<skip_duplicates/>	update_on_bulk="true"	<p>Available in version 5.0.3.1</p> <p>Use to change the temporal behavior with .bulk messages. With this attribute set to <b>true</b>, the Agent will do the following when a .bulk message is received:</p> <ol style="list-style-type: none"> <li>1) Check if the sys_updated_on value of the message is different than the latest temporal record in the table.</li> <li>2) If the value is different, insert a new temporal record.</li> <li>3) If the value is the same, update the latest temporal record.</li> </ol> <p>This is useful for when new fields are added to a table with new default values but the record itself hasn't been changed otherwise. This feature only applies to .bulk messages as .insert and .update messages will always insert new temporal records.</p> <p>The default value is <b>false</b> if this attribute is not specified in which case the Agent will have default temporal behavior and always insert new temporal records.</p> <p><b>ⓘ NOTE:</b> Executing multiple shares for the same table at the same time resulting in multiple messages for the same record may create duplicates or discrepancies.</p> <pre>&lt;temporal&gt;   &lt;skip_duplicates update_on_bulk="true" /&gt; &lt;/temporal&gt;</pre>

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

## 5

### Confirm temporal data sharing

Log into the database that your DataSync Agent is sharing data to, and confirm that **psp\_valid\_from** and **psp\_valid\_to** timestamps are being created and that the **psp\_valid\_to** timestamp is updated whenever a change is made to the same record.

**NOTE:** By default, the primary keys for the subscribed table your database will be a composite key comprised of the **sys\_id** value, **from** column value, and **to** column value.

urgency	sys_created_on	sys_id	psp_valid_from	psp_valid_to
3	2019-03-14 22:19:34	6a8a9f7fdb80330076fc7b3868961982	2019-03-14 15:48:59	2019-03-14 15:49:57
3	2019-03-14 22:19:34	6a8a9f7fdb80330076fc7b3868961982	2019-03-14 15:49:58	2019-03-14 15:56:32
1	2019-03-14 22:19:34	6a8a9f7fdb80330076fc7b3868961982	2019-03-14 15:56:33	2030-12-31 00:00:00

Value should be  
when the record

Value should be the same as  
the value in <max\_datetime>

### Similar topics

- [Change data types](#)
- [DataSync Agent field type mappings](#)
- [Monitoring DataSync Agent logs](#)
- [Skip altering of database tables](#)
- [Specify Working Directories for Agent Files](#)

### Contact Perspective Support



US: 1 888 620 8880

UK: 44 208 068 5953

[support@perspectium.com](mailto:support@perspectium.com)