

# Set up temporal data

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.

## What's on this page?

- [Database Support](#)
- [How to set up temporal data in your agent](#)

## Database Support

The following are a list of database the temporal agent supports:

- Oracle
- MySQL
- MSSQL
- SAP Hana 2.0
- PostgreSQL

[Go to top of page](#)

## How to set up temporal data in your agent

Prerequisites:

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

To set up a temporal data in a local database, follow these steps:

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. To use the default configurations for syncing temporal data (columns will be named **psp\_valid\_from** and **psp\_valid\_to**, the maximum **psp\_valid\_to** value will be **9999-01-01 01:01:01**, and 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="1">
        <task_name>test_subscribe</task_name>
        <message_connection password="encrypted:vl0tU7lyu8N
/EFIJH85SSBtaIt7qEEfvqiqt9VZyYE=" 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>
        <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>
        <archive_datetime>2050-12-31 00:00:00</archive_datetime>
      </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>Customize your temporal database table's maximum date and time stamp in the <b>psp_valid_to</b> column.</p> <pre>&lt;max_datetime&gt;2030-12-31 00:00:00&lt;/max_datetime&gt;</pre> <p>Replace the date and time stamp with any value you would like to assign for custom configuration.</p> <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>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/>  |              | This will disable <b>&lt;archive_datetime/&gt;</b> functionality and skip processing <b>.archive</b> messages   |

|                        |                          |   |
|------------------------|--------------------------|---|
| <column>               | type="from"<br>type="to" | <p>Customize the column names for the <b>psp_valid_from</b> and <b>psp_valid_to</b> columns by modifying the values within the &lt;column type="from"&gt; and &lt;column type="to"&gt; directives, respectively.</p> <pre>&lt;column type="from"&gt;psp_valid_from&lt;/column&gt; &lt;column type="to"&gt;psp_valid_to&lt;/column&gt;</pre>   |
| <column>               | primary_key="true"       | <p>Customize which temporal columns are primary keys by modifying the values within &lt;column primary_key="true"&gt;. By default, both temporal columns are primary keys. However, they both cannot be non-primary keys, meaning both columns cannot have &lt;column primary_key="false"&gt;. This is so that it can still insert temporal entries into the database and the database does not complain about duplicate entries.</p> <pre>&lt;column primary_key="false"&gt;psp_from&lt;/column&gt;</pre>  |
| <timestamp_difference> |                          | <p>Customize a time period between temporal entries by adding &lt;timestamp_difference&gt; inside &lt;temporal&gt;.</p> <pre>&lt;temporal&gt;   &lt;timestamp_difference&gt;1&lt;/timestamp_difference&gt; &lt;/temporal&gt;</pre> <p>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/>     |                          | <p>Skip inserting duplicate entries by adding directive. &lt;skip_duplicates/&gt; inside &lt;temporal&gt;.</p> <p><b>ℹ NOTE:</b> This may affect the agent's performance as it checks with the database and does a comparison every time a message comes in.</p> <pre>&lt;temporal&gt;   &lt;skip_duplicates/&gt; &lt;/temporal&gt;</pre>   |
| <skip_duplicates/>     | check_all_fields="true"  | <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>  |
| <update_on_bulk/>      |                          | <p>Use to change the temporal behavior with <b>.bulk</b> messages. With this attribute set to <b>true</b>, the Agent will do the following when a <b>.bulk</b> 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 <b>.bulk</b> messages as <b>.insert</b> and <b>.update</b> 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>ℹ 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;update_on_bulk/&gt; &lt;/temporal&gt;</pre> |

5

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 up  
when the record is

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

[Go to top of page](#)