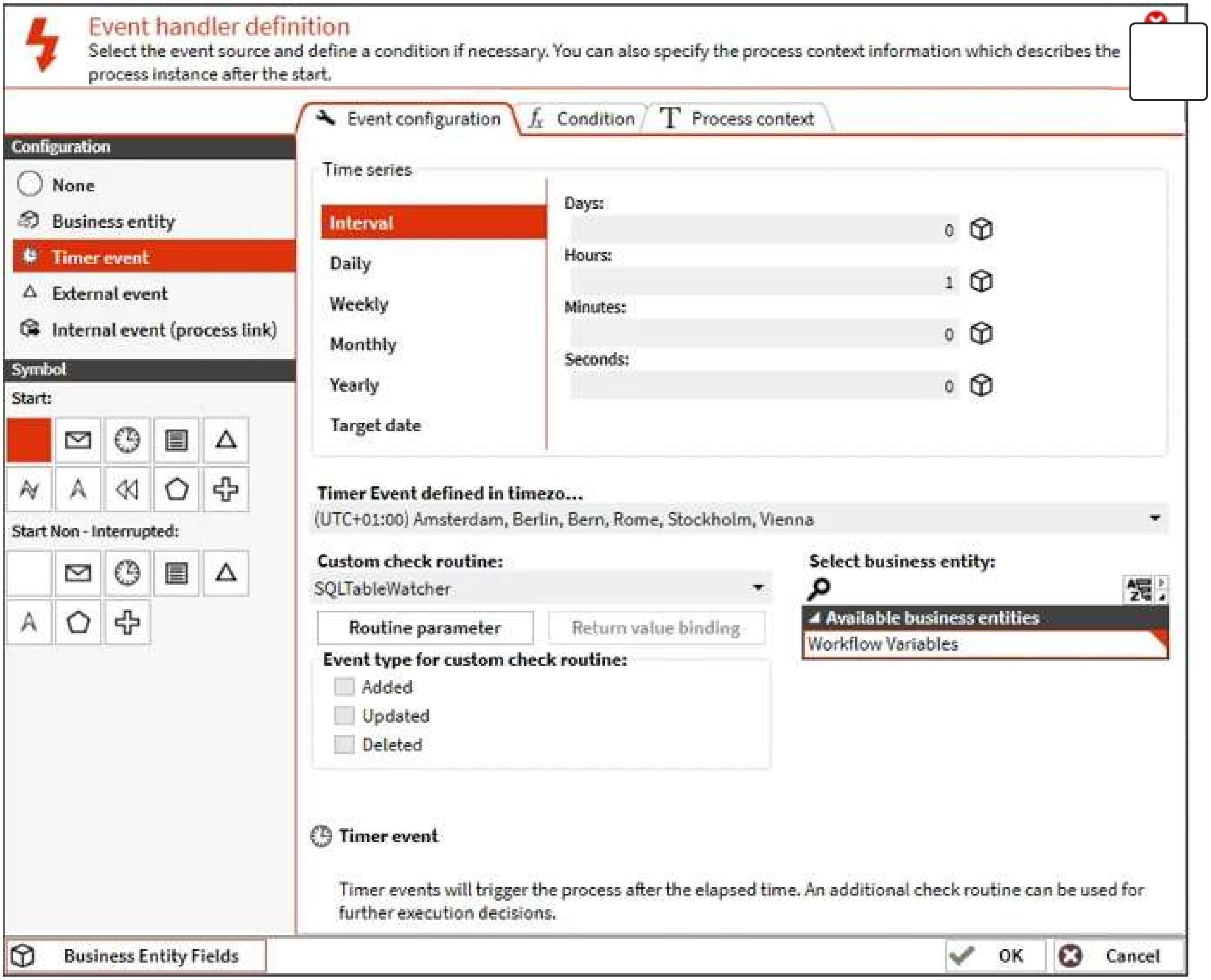
SQL Table Watcher

Operating Principle

The **SQL Table Watcher** monitors a specific database table and reads all entries of the table in an interval which can be configured over the **Timer event.** A workflow is started for every new table entry. When configuring the SQL table watcher several database information is required, which will be described in the following.

Configuration

To configure the SQL table watcher, you first need to set a **Timer event** as **Start event**. In the timer event configuration, you can now select the **SQLTableWatcher** under the **Custom Check Routine**.



Event handler definition

Select the event source and define a condition if necessary. You can also specify the process context information which describes the process instance after the start.

Configuration

- ○ None
- Business entity
- Timer event
- External event
- Internal event (process link)

Symbol

Start:

Start Non - Interrupted:

Time series

- Interval
- Daily
- Weekly
- Monthly
- Yearly
- Target date

Days: 0

Hours: 1

Minutes: 0

Seconds: 0

Timer Event defined in timezo...: (UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna

Custom check routine: SQLTableWatcher

Event type for custom check routine:

- □ Added
- □ Updated
- □ Deleted

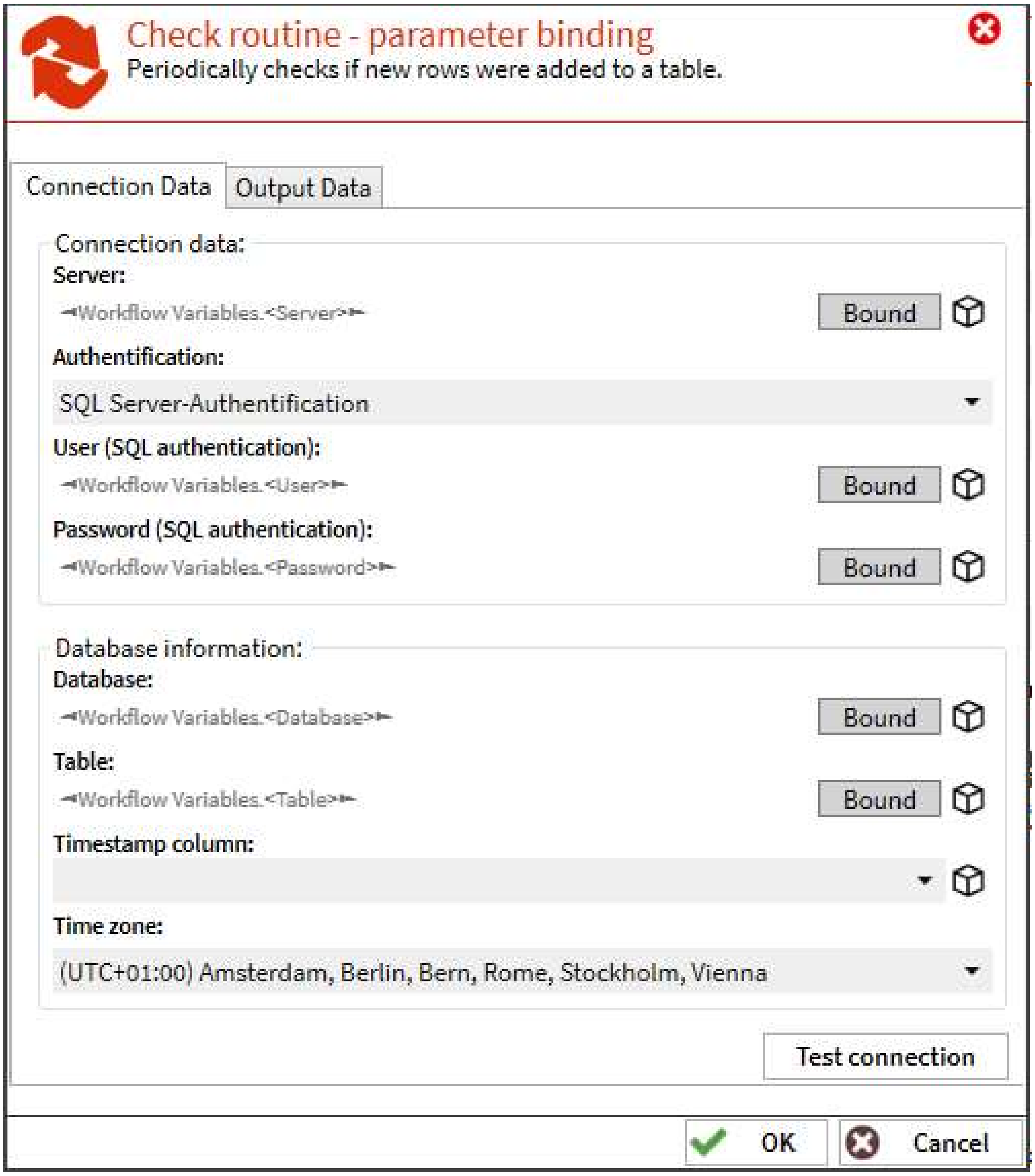
Select business entity:

- Available business entities
  - Workflow Variables

Timer event

Timer events will trigger the process after the elapsed time. An additional check routine can be used for further execution decisions.

After that, the input parameters need to be configured. To do that, you first can select a business entity and then click the button **Routine parameter** to open the dialog. If you would like to use variables for the credentials, you now have all fields of the previously selected business entity available. But the credentials can also be filled by directly writing data into the boxes. When using the **SQL-Authentication** you also have to specify the username and password which is authorized to access the database and table. When using **Windows-Authentication**, this won't be necessary. After you filled all credentials you can test the connection whether it works or not and the output data can also be configured when switching to the **Output Data** tab.



Check routine - parameter binding

Periodically checks if new rows were added to a table.

Connection Data | Output Data

Connection data:

Server: ◄Workflow Variables.<Server>► Bound

Authentification: SQL Server-Authentification

User (SQL authentication): ◄Workflow Variables.<User>► Bound

Password (SQL authentication): ◄Workflow Variables.<Password>► Bound

Database information:

Database: ◄Workflow Variables.<Database>► Bound

Table: ◄Workflow Variables.<Table>► Bound

Timestamp column:

Time zone: (UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna

The table that you see needs a column of type 'datetime2'. Set this column as 'Timestamp column' in the dialog. The table watcher checks for all entries with a newer timestamp than the last table watcher has run (interval).

Confirm the configuration with clicking **OK**. Back in the **Start event** dialog, click **OK** once more and the SQL table watcher has been configured successfully.