

Stored Procedure Activity Overview

Last Modified on 03/28/2019 11:49 am EDT

V8.8

Overview

Use the Stored Procedure Activity to define an activity that executes an SQL stored Procedure from the Cora SeQUENCE database. The Stored Procedure Activity can accept data from the workflow as its input parameters and use that data as part of the procedure.

Use cases

- Update a large amount of data at one time to remain at the database level.
- When multiple databases are connected and you want the activity to be transactional.

Configuration

Prerequisite

- Create one or make sure that the required Stored Procedure was created in the SQL server.

Parameters

Parameter	Description	Settings
Maximum Retries	Determines the number of times the system will try to execute a stored procedure after it fails. Type: integer Values: 0 and > 0 Default: 0	"0" : No retries. The system does not try to execute the failed activity. The activity remains in "failed" status. Greater than "0" : The system tries to execute the stored procedure again according to the specified value only after all retries the activity is tagged as failed.
Retry Interval	Determines the time, in milliseconds, between each retry attempt. Type: integer Values: > 0 Default: 1	Greater than "0" : The time that the system waits before retrying to execute the failed activity.
Timeout	Determines the maximum period of time, in seconds, during which the system waits for the stored procedure to execute. Type: integer Values: > 0 Default: 1200	Null or blank (you delete the default value, and do not enter a new one): The system uses the default value.

You can set up the retry parameters in the Properties pane, under Behavior, or in the activity

configuration wizard.

Properties pane	Wizard
<p>Behavior</p> <p>Compensation <input type="text"/></p> <p>Maximum Retries 0</p> <p>Retry Interval (ms) 1</p> <p>Rollback Message <input type="checkbox"/> <input type="checkbox"/></p> <p>Timeout (seconds) 1200</p> <p>Variables <input type="text" value="LocalSync"/></p>	<p>Stored Procedure Options ?</p> <p>Maximum Retries Set the number of times the system tries to execute the stored procedure after it fails. <input type="text" value="0"/></p> <p>Retry Interval Set the time between each retry attempt. <input type="text" value="1"/> milliseconds</p> <p>Timeout Set the maximum period of time during which the system will try to execute the stored procedure before it fails. <input type="text" value="1200"/> seconds</p> <p><input type="button" value="Back"/> <input type="button" value="Finish"/> <input type="button" value="Cancel"/></p>

Procedure

1. To add the Stored Procedure activity, from the App Studio Toolbox, select **Server Side > Stored Procedure**.
2. On the Stored Procedure Activity Properties screen, enter a significant name for the activity, and then click **Next**.
3. On the Stored Procedure Definition screen, do one of the following:
 - Keep the connection to the default database.
 - Click the three dots button to add a new connection string.
4. Select the relevant stored procedure.

NOTE

The list of stored procedures displays only stored procedures that were created in the referenced database with the prefix *spWFA* and are part of the *dbo* schema.

5. Provide the stored procedure's incoming parameters using Sequence expressions.
 - Define the parameters as hard-coded values.
 - Define the parameters as the result of an advanced workflow expression.
6. Click **Next**.
7. On Stored Procedure Options screen, define the following settings:
 - Maximum Retries
 - Retry Interval
 - Timeout
8. Click **Finish**.
9. On the Properties panel, select the **ThrowOnErrorInvoking** property, to set the system to throw an exception if any of the expressions fails during execution.
10. Validate the activity to make sure that you don't have any errors.

V8.1-V8.7

Context

Use the Stored Procedure Activity to define an activity that executes a SQL Stored Procedure (i.e., pre-written SQL procedure) from the Cora SeQuence database. The Stored Procedure Activity can accept data from the workflow as its input parameters and use that data as part of the procedure.

Use Cases

- Update a large amount of data at one time to remain at the database level.
- When multiple DBs are connected and you want the activity to be transactional.

Wizard Tips

- Only stored procedures that were created in the Cora SeQuence database with the prefix *spWFA* display in the Stored Procedures List. To create stored procedures, navigate to the **Programmability > Stored Procedures** folder.
- Define the parameters you want to send from the workflow to the stored procedure.
 - Define the parameters as hard-coded values.
 - Define the parameters as the result of an advanced workflow expression.
- Make sure you send correct value types to the stored procedure.

Activity Tips

- You can monitor the stored procedure in the Cora SeQuence debugger, where you can check if the stored procedure successfully executed, if the parameters were valid, and so on.
- The Stored Procedure Activity includes the *ThrowOnErrorInvoking* property. If you select this property, the system throws an exception in any parameter expression fails during execution. The *ThrowOnErrorInvoking* property is in the Misc. section of the Properties panel.

V7.x

Context

Use the Stored Procedure Activity to define an activity that executes a SQL Stored Procedure (i.e., pre-written SQL procedure) from the Cora SeQuence database. The Stored Procedure Activity can accept data from the workflow as its input parameters and use that data as part of the procedure.

Use Cases

- Update a large amount of data at one time to remain at the database level.
- When multiple DBs are connected and you want the activity to be transactional.

Wizard Tips

- Only stored procedures that were created in the Cora SeQuence database with the prefix *spWFA* display in the Stored Procedures List. To create stored procedures, navigate to the **Programmability > Stored Procedures** folder.
- Define the parameters you want to send from the workflow to the stored procedure.
 - Define the parameters as hard-coded values.

- Define the parameters as the result of an advanced workflow expression.
- Make sure you send correct value types to the stored procedure.

Activity Tips

- You can monitor the stored procedure in the Cora SeQuence debugger, where you can check if the stored procedure successfully executed, if the parameters were valid, and so on.