

Define a Data Model Stored Procedure Query

Last Modified on 01/20/2021 6:38 am EST

Overview

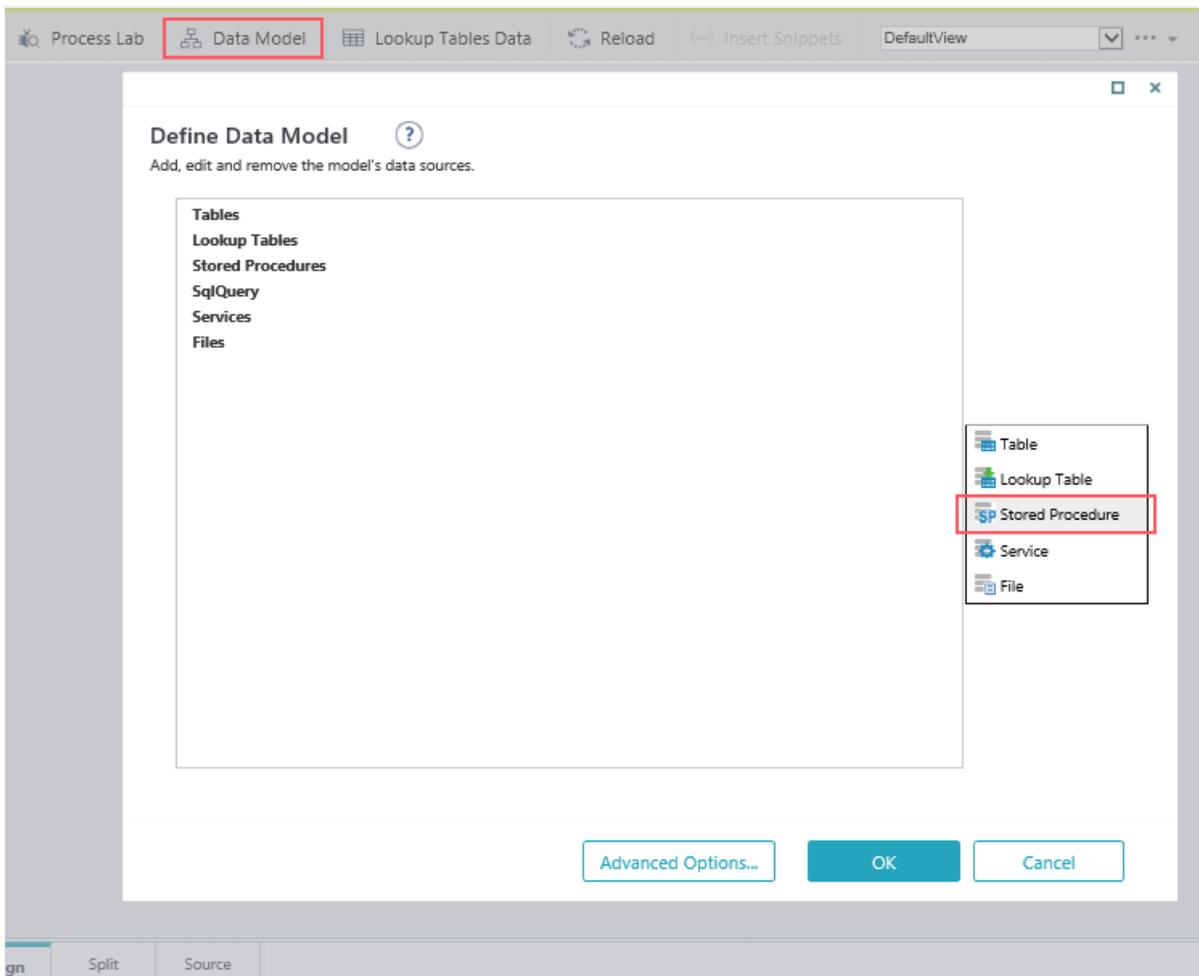
You can use a stored procedure as a query source in a Data Model. The data from the stored procedure can then be bound to form controls the same way as a regular user table, although it will be read-only.

NOTE

Stored procedure names must begin with the prefix **USP**.

Access and define the Data Model

1. In the Administration site, open a form.
2. In the UX Studio screen, click **Data Model**.
3. In the Define Data Model screen, click **Add > Stored Procedure**.



Select a stored procedure

You can select a stored procedure either from the default database or define a connection string to access another SQL Server database (available with V8.8.2 and later).

The list of stored procedures includes only stored procedures with the USP prefix.

Define input and output parameter mappings for the stored procedure from the default database

Stored Procedure ?

Default Database Connection String

Find a stored procedure...

- USP_PD_GetTechnicalInformationHistory
- USP_QuestionsWithUpdateType
- usp_ReturnSalesByNatureOfClient
- usp_SalesFunnel
- usp_SalesResults
- USP_SaveReconKeys**
- USP_Shila_OccupationAndTreatmentAreaByEmployeeId
- USP_Shila_WorkArrangement
- USP_Shila_WorkingHoursAreCorrct

Parameters of the Stored Procedure 'USP_SaveReconKeys'

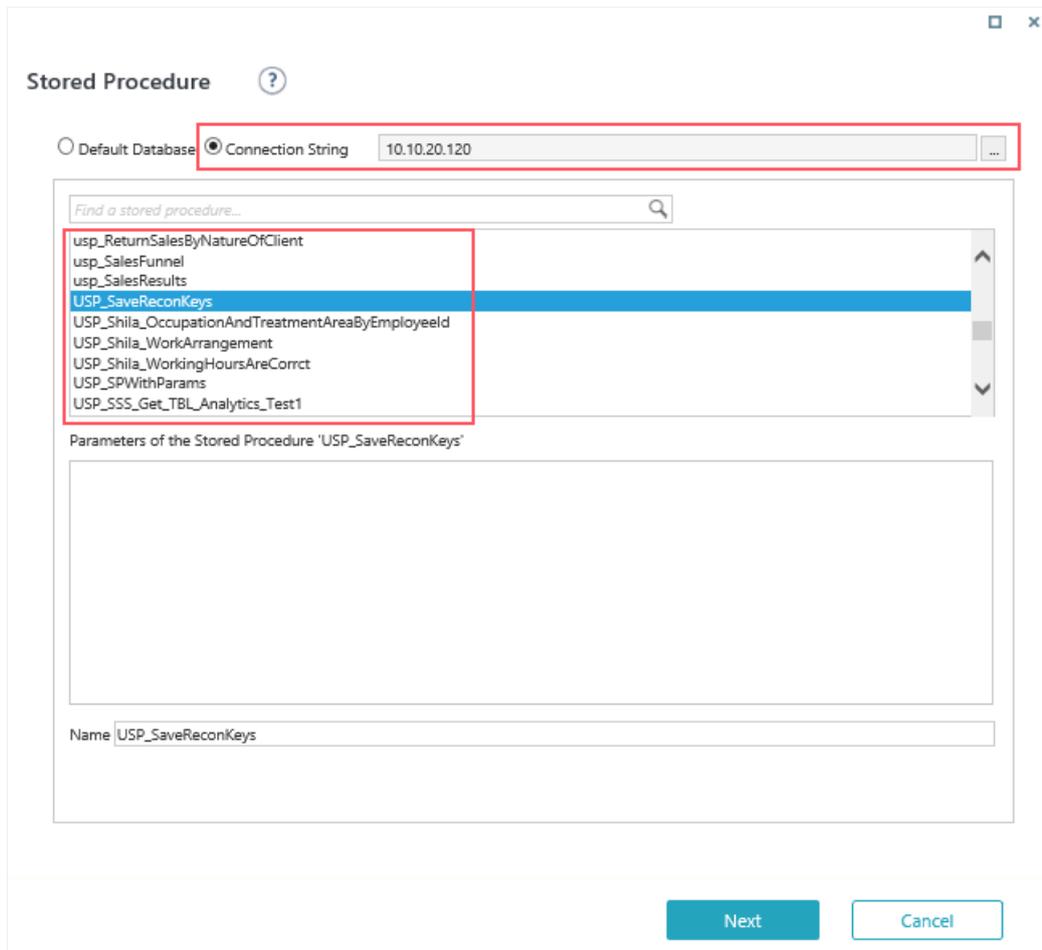
Name	Type	Size	Mask	Value
@fidiwid	Int64	0	In	
@fidiactid	Int64	0	In	
@invoiceids	AnsiString	2147483647	In	
@receiptuniquekeys	AnsiString	2147483647	In	
@reset	Boolean	0	In	

Name: USP_SaveReconKeys

Next Cancel

- Set the Mask value to map the corresponding filter or paging parameter. For more information, see [this article](#). Cora SeQuence assigns a default mask to each parameter based on the parameter name. The Mask property enables Cora SeQuence to know which data to receive from the client, and to which parameter of the stored procedure pass this data to, based on its type (input, filter, pagesize, and so on).
- In the Value field, click the ellipsis button to define the value based on an expression.

For selecting a stored procedure from another SQL Server database, you need to define a connection string and select a stored procedure from the list.



Add fields to the data source

- Select the required fields from the **Stored Procedure Fields** panel, and then click the double arrow button to add them to the **Query Fields** panel.
- **Name** is the field name in the query.
- **Projected Field** is the parameter name from the stored procedure.

Stored Procedure Data Fields
?

Stored Procedure Fields
Existing fields of your selected Stored Procedure

invoicelds
receiptUniqueKeys

Query Fields
Fields that build your query

	Name	Projected Field	
<input type="checkbox"/>	invoicelds	invoicelds	Delete
<input type="checkbox"/>	receiptUniqueKeys	receiptUniqueKeys	Delete

>>

Back
Finish
Cancel

What next?

After you finish defining the stored procedure query, you can use it as a data source for controls in your form, or in Data Insert and Assign server-side activities as the data source in an expression.

TIPS

- When adding stored procedures to a Cora SeQUENCE data model, ensure there is only one return schema for all code paths and the procedure runs as expected when all input parameters are set to default (null).
- The Cora SeQUENCE data model must be able to determine the column names and their data types returned from the stored procedure. Adding a stored procedure that returns a dynamic schema with undetermined column names and data types will fail.

In other words, if your stored procedure builds a dynamic SQL with conditional logic and executes using `EXEC` or `sp_executesql`, then Cora SeQUENCE might fail to recognize the schema, even if the stored procedure works at runtime with expected parameters. To resolve this, insert the results into a temporary table and return that table as the stored procedure result set.