

Session Data

Session Data is an entity used to save session context information. Each application has a single Session Data entity, located in the Repository folder. New applications are created with an empty Session Data entity which cannot be deleted. The Session Data entity enables defining variables and their types in the session context during design time. In Runtime, a Session Data object is created for each session, and it is possible to set and then use the variables' values in the current session. When using these variables in Connection Pools, whenever a connection is returned to the pool, the Session Data is reset with the default values.

Note:

When executing a path using the Execute Procedure node, by default a new session is created for the executed procedure, therefore a new Session Data object is created for the executed procedure. To use the same Session Data object as the Procedure, map the session ID to the executed procedure ID.

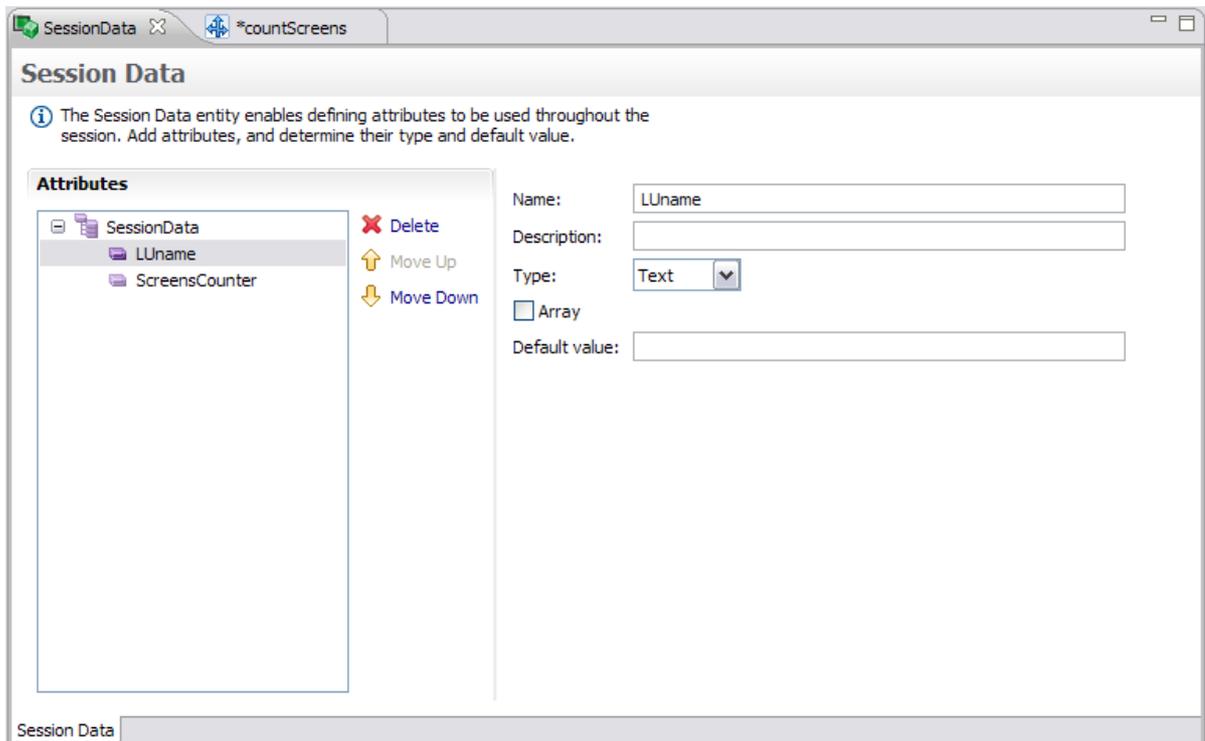
The Session Data entity can be used in the following entities:

- Path Procedure mappers
- Business Activity Mappers

For applications created in earlier ApplinX versions, when synchronizing the application database, an empty Session Data entity is created. The old database cannot contain a data structure named "SessionData" (and when such a data structure exists, it must be renamed prior to the import process).

> To define variables and types of variables:

1. Within the Repository node of the application, double-click on SessionData. The Session Data entity is displayed.



2. Click **Add Attribute**. An attribute is added to the Session Data.
3. Provide a name for the attribute and a relevant description.
4. Select the type of attribute: Text, Long, Boolean, Double, Integer, Float, Byte or Date.
5. Determine if it is an array.
6. Enter a default value.

➤ **To Set and Use the Variables:**

1. Setting and using the values is implemented by mapping to or from the session data variables (refer to the Mapper).

The following illustrates mapping from the Session Data to the Procedure's outputs.

Path Procedure

i Drag and drop nodes from the right to the procedure tree. Set the parameters of the nodes to define the procedure logic.

Show name Show details Show description

[Path] Procedure

- [Mapper_1] Map Links: 1
 - Session/SessionData/ScreensCounter -> Out/ScreensCounter

Navigation

Assignment

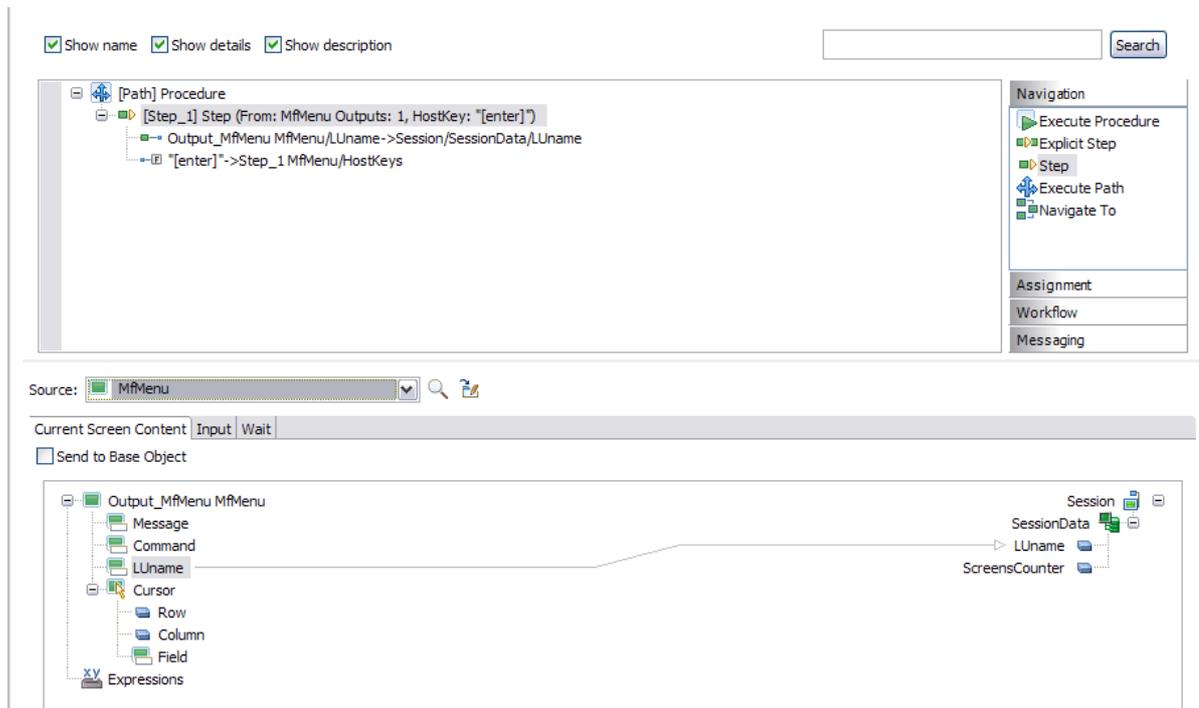
- New Object
- Mapper
- Merge Arrays
- Screen Mapper

Workflow

Messaging

```
graph LR; subgraph Left_Tree [Left Tree]; direction TB; Out1[Out] --- Session1[Session]; Session1 --- SessionId1[SessionId]; Session1 --- DeviceName1[DeviceName]; Session1 --- SessionData1[SessionData]; Session1 --- LUname1[LUname]; Session1 --- ScreensCounter1[ScreensCounter]; End1[Expressions]; end; subgraph Right_Tree [Right Tree]; direction TB; Out2[Out] --- Session2[Session]; Session2 --- SessionData2[SessionData]; Session2 --- LUname2[LUname]; Session2 --- ScreensCounter2[ScreensCounter]; end; ScreensCounter1 --- ScreensCounter2;
```

The following illustrates mapping from the screen's field content to the Session Data



The following illustrates mapping from an Expression which uses the Session Data, to the Session Data.

Path Procedure

① Drag and drop nodes from the right to the procedure tree, Set the parameters of the nodes to define the procedure logic.

Show name Show details Show description

[Path] Procedure

- [Mapper_1] Map Links: 1
 - Calculate((Session/SessionData/ScreensCounter)+1)->Session/SessionData/ScreensCounter

Navigation

Assignment

- New Object
- Mapper
- Merge Arrays
- Screen Mapper

Workflow

Messaging

Session

- SessionId
- DeviceName
- SessionData
 - LUsername
 - ScreensCounter
- Expressions
 - Calculate((Session/SessionData/ScreensCounter)+1)

Session

- SessionData
 - LUsername
 - ScreensCounter