

## **ApplinX Exercises**

**SOA**

Version 9.8

June 2015

This document applies to ApplinX Exercises Version 9.8.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2001-2015 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

**Document ID: APXE-SOA-98-20150624**

## Table of Contents

SOA .....	v
1 Collecting Data from Multiple Screens .....	1
2 Creating a Connection Pool .....	3
3 Creating Connection Information Sets .....	5
4 Defining a Database Connection in ApplinX Repository .....	7
5 Using a DB Connection in a Flow Procedure .....	9
6 Putting all the Pieces Together... ..	11



## Exercise Objectives

ApplinX allows you to expose procedures as either web services or procedure clients, generated to the project's code. Procedures can retrieve data from various sources such as host screens, RPG and COBOL transactions, an external Database and even an External Web Service.

At the end of the exercise you will know how to:

Use a rather simple Access (mdb) Table to get additional data about each record returned from the host screen. This exercise demonstrates how you can combine together data from the host, with data from other sources.

## Requirements

- Read the Installation and Getting Started section.
- Complete the Instant Demo Exercise. It's the starting point of this exercise.

[Collecting Data from Multiple Screens](#)

[Creating a Connection Pool](#)

[Creating Connection Information Sets](#)

[Defining a Database Connection in ApplinX Repository](#)

[Using a DB Connection in a Flow Procedure](#)

[Putting all the Pieces Together...](#)

---

# 1 Collecting Data from Multiple Screens

---

## Exercise objectives

The following exercise demonstrates how to collect data (tabular data) from several screens. This isn't the only way of doing it, but for this common scenario, of a scrollable table, it is the easiest.



### Exercise

Create a path procedure which collects all customer information as output variables and displays the output on the first customer detail web page, preferably, using tabs.



### Solution steps

1. Navigate to a screen with table formatted data (BrowseCustomers, Browseproposals).
2. Identify the screen if you haven't already done so.
3. Map all fields, single fields as well as multiple fields which represent the table's columns.
4. In the Table tab, create a new table. Delete unnecessary columns and change column captions to be more user-friendly.
5. Select Primary key columns. These would usually be columns with unique values.
6. Record a Path Procedure that scrolls through the table until it reaches an "End of Data" (or similar) message on the host screen.
7. Open the path for editing, click the step where the path is scrolled and open its output tab.
8. Select all the required fields from the table entity (hold CTRL and click for multiple selection). Then right click the selection and click "New Data Structure". A dialog box is displayed. Enter the name of the new data structure and click OK. You have just created a Structure that represents one row in the host table.

9. Add an additional attribute to the data structure in which database information will be stored.
10. Click the “Flow Procedure” node and open the output tab. Click Add structure, and name the variable. Select the newly created Data structure from the “type” Combo. Finally, check the “array” checkbox to make your output variable actually represent a table (collection of rows).
11. Add a screen mapper node after the “scroll down” step. Select the relevant screen and then drag the table node onto the newly added variable.
12. Use the “loop for each” to cycle through the output and run the previously created DB flow procedure to gather additional data about each record (figure I). Map the Procedure output to the data structure (figure II). Make sure you map it to the correct record (Figure III).
13. Finally, associate the procedure with a procedure group. Create a new procedure group in the repository then add the new procedure to the “Assigned Procedures” box by:
  - a. Clicking the “Add Procedure” button and selecting the procedure from the dialog.
  - b. Dragging the procedure from the repository on to the “Assigned Procedures” box.



Accompanying movies:

- Recording a Path Procedure
- Retrieving Data from Screens with Path Procedures
- Collecting Complex Data from Several Screens



## 2 Creating a Connection Pool

---

Creating a connection pool can be essential for performance reasons. Ideally, you'd want users to be able to connect directly to the relevant screen (where the data is) rather than have them log on, navigate to the screen and only then get the data they want.



### Exercise

Create a Connection Pool (with a fixed size of 3), where each connection logs on to the system and waits for the users on a particular screen.



Accompanying movies:

- Creating a Connection Pool



## 3 Creating Connection Information Sets

---

Now that you have a connection pool that uses a logon path to connect to the host, you need to pass the required variables to the login path that the connection pool runs.

This functionality is handled by the Connection Information Set entity. Once associated with a connection pool, you can define path variables and session parameters that will be used when a new host connection is added to the connection pool.



### Exercise

Create a Connection Information Set with sets of usernames and passwords that will be passed to the logon initialization path.



**Note:** If the host doesn't support the same user logging on more than once, you'll have to set the repeat limit of each set to 1. This way every set can only be used by one connection at a time.



### Solution steps:

Refer to the SOA demo to either the Default\_CustomerDetails\_GCT entity or to the Default\_CustomerList\_GCT entity.



Recommended reading: Designing and Developing an Application>Applinx Entities>Connection Information Sets.



## 4 Defining a Database Connection in ApplinX Repository

---



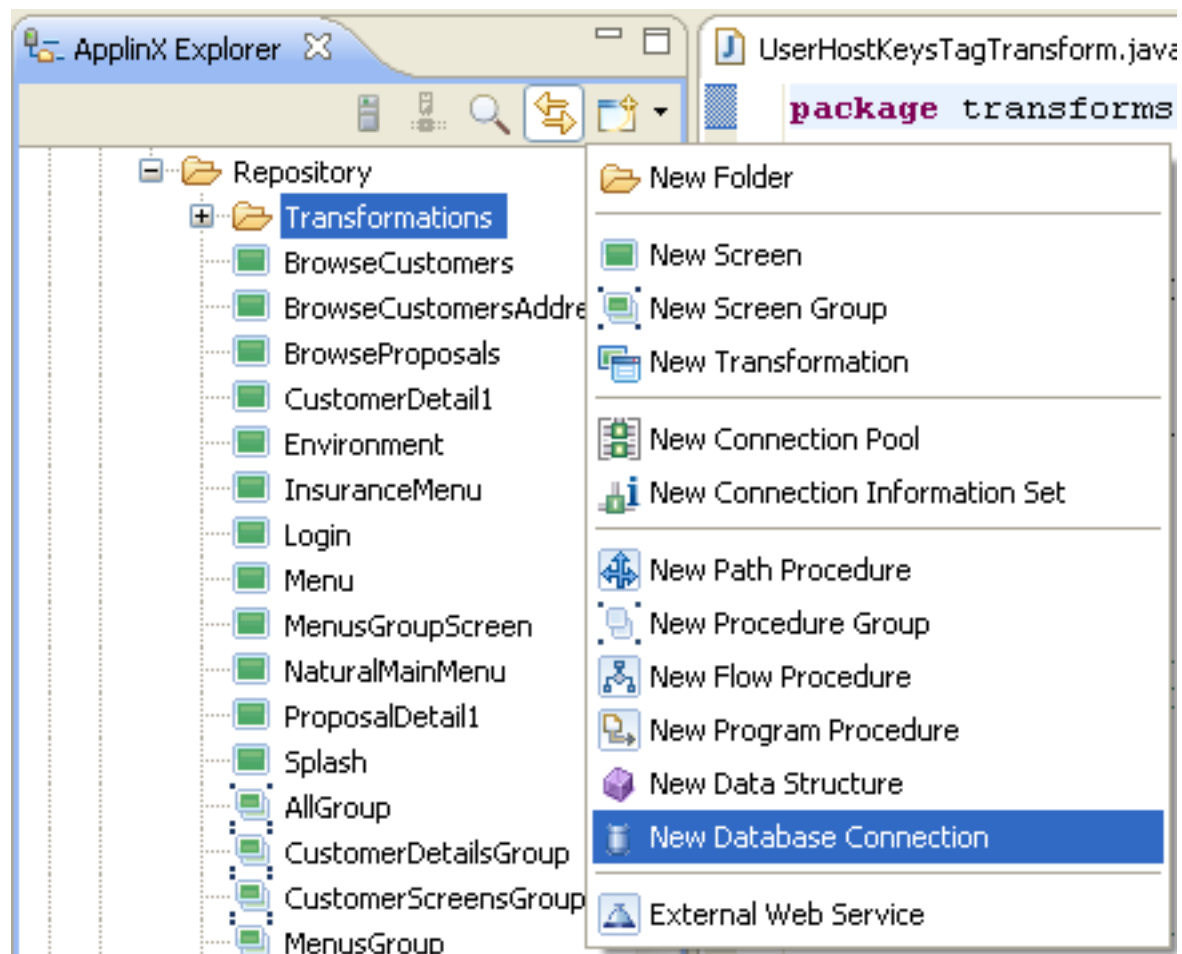
### Exercise

Define a database connection in ApplinX repository.



### Solution steps:

1. Create a new Database Connection entity in the repository.



2. In the Database Editor, enter the parameters required to establish a successful connection.
3. Test your connection by clicking on the Test Connection button (This button is not available when using an Access or derby DB).



Recommended reading in ApplinX documentation:

- Designing and Developing an Application>ApplinX Entities>Database Connection.
- Reference Guide>ApplinX Entities>Database Entity.



Accompanying movies:

- Creating a Database Connection and using it in a Flow Procedure

# 5

## Using a DB Connection in a Flow Procedure

---



### Exercise

Create a flow procedure which receives an ID string, queries it in the Database and returns an output (you can find an MDB file in the installation directory /host-Applications/SOADemo-files/policies.mdb).



### Solution steps

#### Querying a Database

1. Drag the DB Select node to the relevant place in the procedure.
2. Choose the Database Connection entity you wish to query.
3. In the SQL statement text box enter the SQL query. Use tokens, if procedure parameters need to be integrated into the statement (see figure below).
4. The Output parameters will show the query output (click the Refresh button if it doesn't).



#### Accompanying movies:

- Creating a Database Connection and using it in a Flow Procedure





## 6 Putting all the Pieces Together...

---

Now that we have a connection pool that will create connections which can take the users to the screen where the data begins, and a path procedure that collects the data, we can put everything together.



### Exercise

Create a Procedure which:

- Takes a connection from the pool.
- Runs the Path Procedure and retrieves its output.
- Returns the connection to the pool.



Accompanying movies:

- Creating a Flow Procedure and Deploying it as a Web Service
- Exposing a Procedure as a Web Service

