

# Building Web Dynpro Application accessing ABAP Functions

## Overview

This Web Dynpro application demonstrates how you connect from Web Dynpro to the remote SAP backend system using an adaptive RFC model. The user interface of this application consists of only two views combined into one viewset. In the first view, the user should be able to enter the departure and destination cities in the appropriate input fields and trigger display of the flight data from the remote SAP backend system using a "Search" button. As a result of this query, all the available flight data will be displayed as a table in the next view.

## Web Dynpro Project Name:

TutWD\_FlightList

## Requirements and Dependencies

You have access to a remote SAP backend system. To test this example application successfully, you must also make sure that this SAP system contains the function module BAPI\_FLIGHT\_GETLIST and the appropriate data from the flight data model. In order to gain access to any functionality within an SAP system, a user must provide the SAP system with some valid credentials by means of the log on process. Since you are using the Web Dynpro Adaptive RFC layer, the user id defined in the Web Dynpro Content Administrator of your J2EE engine will take care of the connection automatically. There are two configuration tasks needs to be performed in order to connect the logical systems defined in the Adaptive RFC model with a physical SAP system.

The SAP System Landscape Directory (SLD) and the SLD bridge are configured and running. In order to connect the logical systems defined in the Adaptive RFC model with a physical SAP system you need to perform two configuration tasks .

### Setting Up the SLD Connection

1. This task is done in the J2EE Engine Administrator Console while the J2EE Engine runs:
  - a. You can start the engine administrator console from the path <Drive>:\usr\sap\<System ID>\JC00\j2ee\admin\go.bat
  - b. Connect and login to J2EE Engine. You need administrator rights for the login process.
  - c. After login, select on the left side the "Cluster" view and expand the node "Server", "Services" and select the entry "SLD Data Supplier".
  - d. Select the "CIM Client Generation Settings" and insert the necessary HTTP connection parameters in order to establish a connection to the SLD.
  - e. Save the data by pressing the "Save" button and check the CIM client connectivity by pressing the "CIMClient Test" button.
  - f. If the test was successfully, you can close the J2EE Engine Administrator Console. Otherwise correct your connection parameters.

### Maintaining the JCO Destinations in the Web Dynpro Content Administrator

2. The logical system names used in the model declaration need to be associated with an actual SAP system defined in the SLD before this application can be executed. This task is done after deployment of this application (after step 3.b in the following instructions) in the JCO

Connections screen of the J2EE Engine Web Dynpro Content Administrator:

- a. Connect to the following URL `http://localhost:<j2ee-http-port>/webdynpro/welcome` on your J2EE server for opening the Content Administrator.
- b. If you use the Content Administrator for the first time you must perform the self registration.
- c. Select the "Browse" view and navigate to the application "FlightListApp" under the project "TutWD\_FlightList".
- d. Select on the right view the "JCO Connections" tab.
- e. Two JCO clients named `WD_FLIGHTLIST_MODELDATA_DEST` and `WD_FLIGHTLIST_RFC_METADATA_DEST` are shown in this view and both of them have the status red.
- f. Select `WD_FLIGHTLIST_MODELDATA_DEST` JCO and press "Create" in order to configure this client. Beside the default values enter the system and security entries in order to establish a connection. After your inputs press "Finish". You can test your configuration data by pressing "Test Connection".
- g. Do the same for `WD_FLIGHTLIST_RFC_METADATA_DEST`.
- h. If the tests were successfully, you can close the Content Administrator. Otherwise correct your connection entries.

## Instructions

To get the FlightList application running, proceed as follows:

1. Importing the project into the SAP NetWeaver Developer Studio
  - a. Unzip the complete contents of the zip file "WebDynpro\_FlightList.zip" into the workspace directory or in another folder of your choice.
  - b. Start the SAP NetWeaver Developer Studio.
  - c. Open the Web Dynpro perspective ("Window"|"Open Perspective"|"...Web Dynpro").
  - d. Choose "File"|"Import".
  - e. Select "Existing Project into Workspace" and choose "Next".
  - f. In the input field "Project Content", specify the folder containing the unzipped Web Dynpro project ("TutWD\_FlightList") and choose "Finish".The project "TutWD\_FlightList" will appear in the "Web Dynpro Explorer" view.

Note:

Depending on which Java compiler preferences are set, you may see some warnings in the "Task" view after importing the project.

If the severity level for problems of type "Unused imports" (set in Preferences – Java – Compiler) has the value "Warning", the compiler will issue a warning for unused import references.

Ignore these warnings!

2. Building the project
  - a. Open the context menu of the project node "TutWD\_FlightList" in the Web Dynpro Explorer view.
  - b. Choose "Rebuild project".
3. Deploying and running the Welcome application
  - a. Make sure that your SAP J2EE Engine is running.
  - b. Open the context menu of the project node "TutWD\_FlightList" and choose "Deploy".

- c. Associate logical system names used in the model declaration with an actual SAP system defined in the SLD before you execute this application (see requirements step 2).
  - d. Open the context menu of the project node "TutWD\_FlightList"|"Web Dynpro"|"Applications"|"FlightListApp" and choose "Run".
- 4. Testing the Flights application
  - a. Enter a valid city name for the Departure City (e.q. Frankfurt) and Arrival City (e.q. New York) and then choose the "Search" button.

**Corresponding Tutorial:**

SAP Web AS for J2EE Applications → Development Manual → Developing Web Applications → Web Dynpro → Tutorials for Web Dynpro Applications → Advanced Tutorials → Model-Tutorials → Creating a Web Dynpro Application Accessing ABAP Functions.

**Date Published:** July 7, 2004