# Web Dynpro for ABAP: Tutorial 6 – Component Usage

**SAP NetWeaver 2004s**

## Copyright

## Icons in Body Text

| Icon | Meaning |
|---|---|
| ⚠ | Caution |
| 💬 | Example |
| 💡 | Note |
| 🧭 | Recommendation |
| SYN | Syntax |

Additional icons are used in SAP Library documentation to help you identify different types of information at a glance. For more information, see *Help on Help* → *General Information Classes and Information Classes for Business Information Warehouse* on the first page of any version of *SAP Library*.

## Typographic Conventions

| Type Style | Description |
|---|---|
| *Example text* | Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options.<br>Cross-references to other documentation. |
| **`Example text`** | Emphasized words or phrases in body text, graphic titles, and table titles. |
| EXAMPLE TEXT | Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE. |
| `Example text` | Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools. |
| **`Example text`** | Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation. |
| **`<Example text>`** | Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system. |
| `EXAMPLE TEXT` | Keys on the keyboard, for example, `F2` or `ENTER`. |

# Web Dynpro for ABAP: Tutorial 6 – Component Usage

## *Preface*

Prerequistes are Tutorial 3 - Navigation and Tutorial 4 – Display a list of bookings for a selected flight.

Just to review back Tutorial 4, we created a context **Flight_List** which contains the flight information and it was advanced with sub-node *BOOKINGS* for the corresponding bookings. Automatically, the booking information was retrived in a new table (using supply function) which shows a list of bookings for a selected flight.

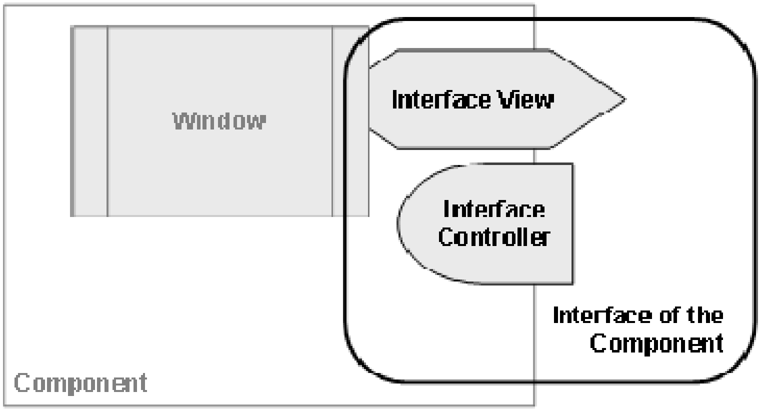
## Component Usages

Real business applications usually consist of several Web Dynpro components. Typical examples of reusable components are dedicated Web Dynpro components that are built to display a special set of data and can be used in several other components.

Every Web Dynpro component has a component interface. It contains exactly one interface controller which consitutes of context, events, methods and at least one interface view. To be able to use an external component you must first define a Component Usage.

In this tutorial, you will develop an example for such a component usage in Web Dynpro ABAP. You will further enhance tutorial 4 – Bookings in order to display customer data of a particular booking using his customer_id as a foreign key.

Note - Displaying the customer data should be implemented as a separate Web Dynpro component.



**Component Interface**

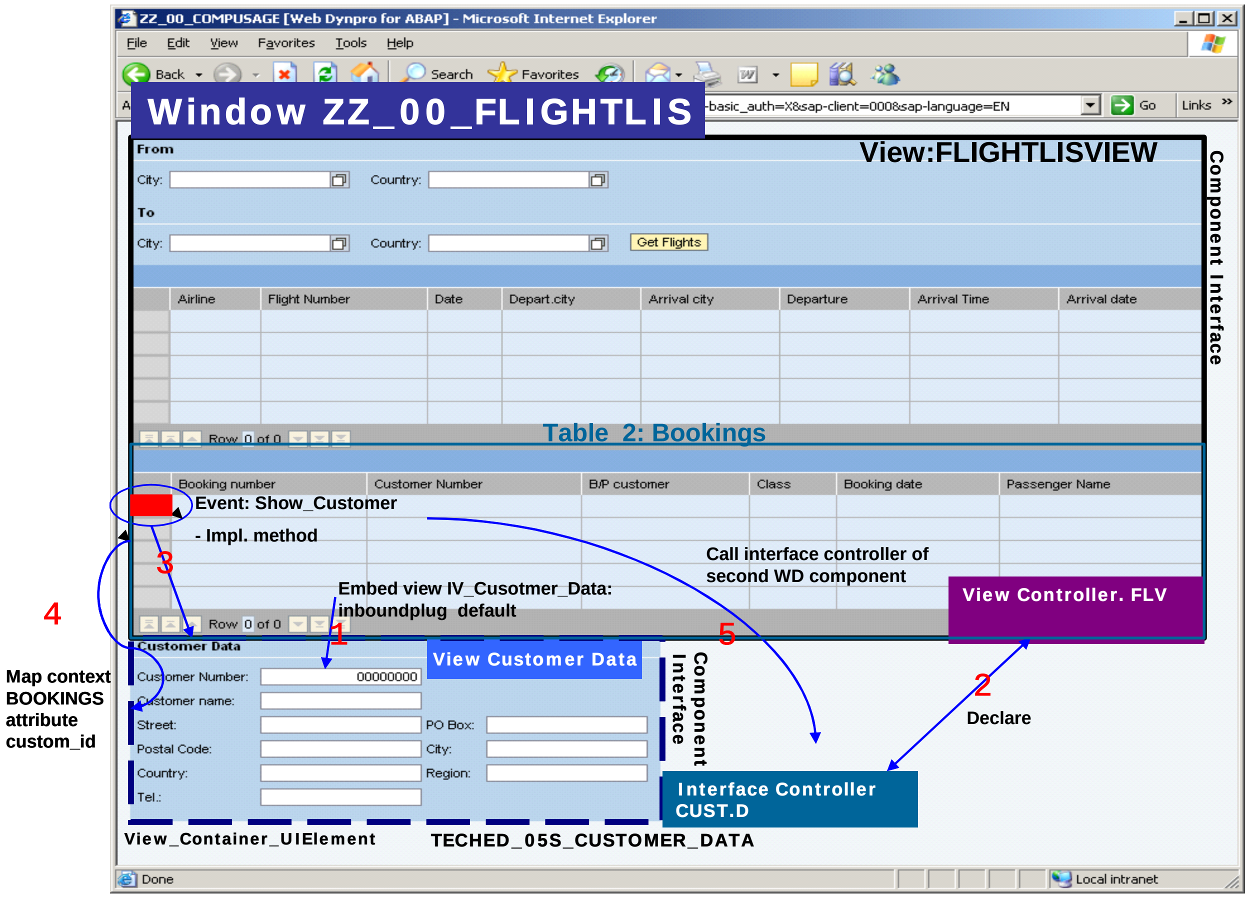
The schema below is a representation of the view composition implemennted in this tutorial.

You create a component ZZ_00_COMPUSAGE which consist of view FLIGHTLISTVIEW. Embed view customer data IV_CUSTOMER_DATA of the second Web Dynpro component TECHED_05S_CUSTOMER_DATA (used component) using a new view container UI element in the main window ZZ_00_FLIGHTLIS.

Declare the view controller of view FLIGHTLISTVIEW so that it can access the interface controller of second Web Dynpro component TECHED_05S_CUSTOMER_DATA. Use the existing UI element TABLE_2 (BOOKINGS), create an event SHOW_CUSTOMER and implement the method for it (ONACTIONSHOW_CUSTOMER).

In order to get the CUSTOMER DATA, you need to read the context attribute CUSTOM_ID from the subnode BOOKINGS and define the mapping. Later call the interface controller of second Web Dynpro Component TECHED_05S_CUSTOMER_DATA.

Selected booking information for a particular Customer ID is stored the variable *Item_Customid*.



## Component Usage

Furthermore, Web Dynpro for ABAP envinorment provides you to create generic components such as ALV, Message Viewers or Select options etc that can be used by various development teams for different or large structured projects.

# Development Objectives

At the end of the tutorial, you will be able to:

- Create a Web Dynpro component ZZ_00_COMPUSAGE with a component interface that can be used another component
- Enhance an existing component with a view container to show the interface view of another component.
- Define the data binding between the outer component and the used component

## At the end of this Tutorial you will be able to

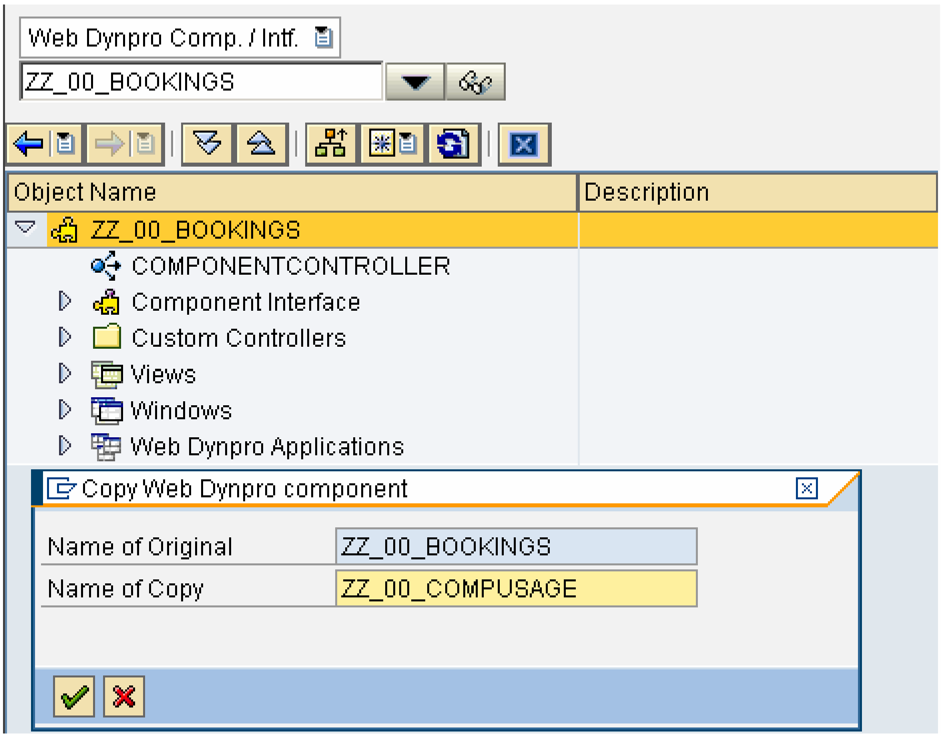
- Understand how larger Web Dynpro projects can be structured with several Web Dynpro components
- Be able to include a reusable Web Dynpro component into another Web Dynpro component
- Know how methods of an used Web Dynpro's Interface Controller is called

# Procedure

1) Copy the Web Dynpro component ZZ_00_BOOKINGS to ZZ_00_COMPUSAGE.
2) Declare a component usage of the TECHED_05S_CUSTOMER_DATA component in your ZZ_00_COMPUSAGE component.
3) Create a ViewContainerUIElement on the layout of the view FLIGHTLISTVIEW.
4) Switch to the window of the main component and embed the view IV_CUSTOMER_DATA of the used component TECHED_05S_CUSTOMER_DATA into the ViewContainerUIElement.
5) Declare that the view controller of the view FLIGHTLISTVIEW can use the interface controller of the used component TECHED_05S_CUSTOMER_DATA.
6) For the existing UI element *TABLE_2* create a new event *SHOW_CUSTOMER* onLeadSelect.
7) Implement the coding of the method *ONACTIONSHOW_CUSTOMER*.
    a. In order to display the corresponding customer data you first need to find out the customer ID of the selected booking.
    b. Call the interface controller method SHOWCUSTOMER and pass on the customer ID.
8) Create a Web Dynpro application. Run the Web Dynpro application component and select different customer records.

# Create the Web Dynpro Component ZZ_00_COMPUSAGE

1. Copy the Web Dynpro component ZZ_00_BOOKINGS to ZZ_00_COMPUSAGE



Web Dynpro Comp. / Intf.

ZZ_00_BOOKINGS

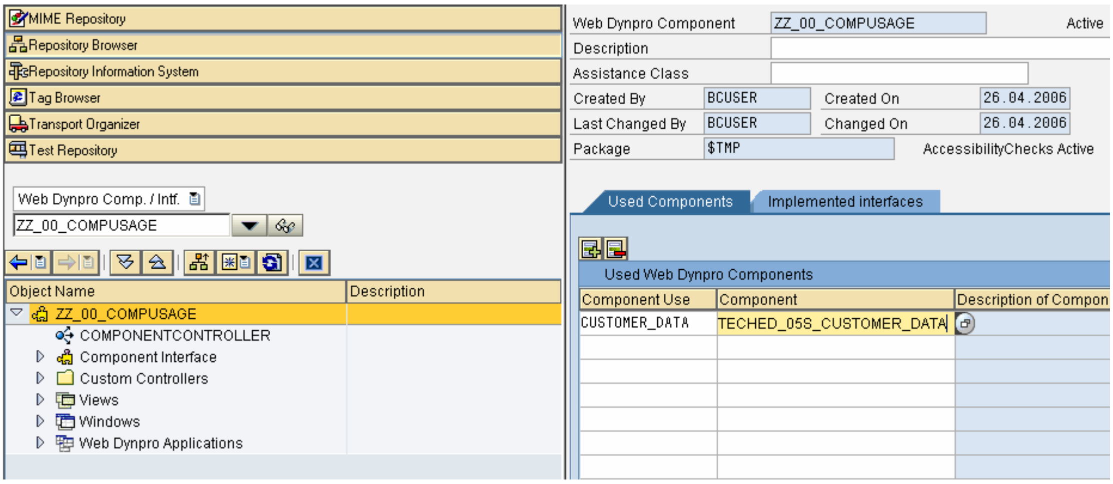
| Object Name | Description |
|---|---|
| ZZ_00_BOOKINGS | |
| COMPONENTCONTROLLER | |
| Component Interface | |
| Custom Controllers | |
| Views | |
| Windows | |
| Web Dynpro Applications | |

Copy Web Dynpro component

| | |
|---|---|
| Name of Original | ZZ_00_BOOKINGS |
| Name of Copy | ZZ_00_COMPUSAGE |

# Declaration of CUSTOMER_DATA Component within ZZ_00_COMPUSAGE for Component Usage

2. Double click the component in the object tree and choose the tab *Used Components.* Create a new Component Use CUSTOMER_DATA using the component TECHED_05S_CUSTOMER_DATA by adding the entries directly in the *Used Components List.*



Web Dynpro Comp. / Intf.

ZZ_00_COMPUSAGE

| Object Name | Description |
|---|---|
| ZZ_00_COMPUSAGE | |
| COMPONENTCONTROLLER | |
| Component Interface | |
| Custom Controllers | |
| Views | |
| Windows | |
| Web Dynpro Applications | |

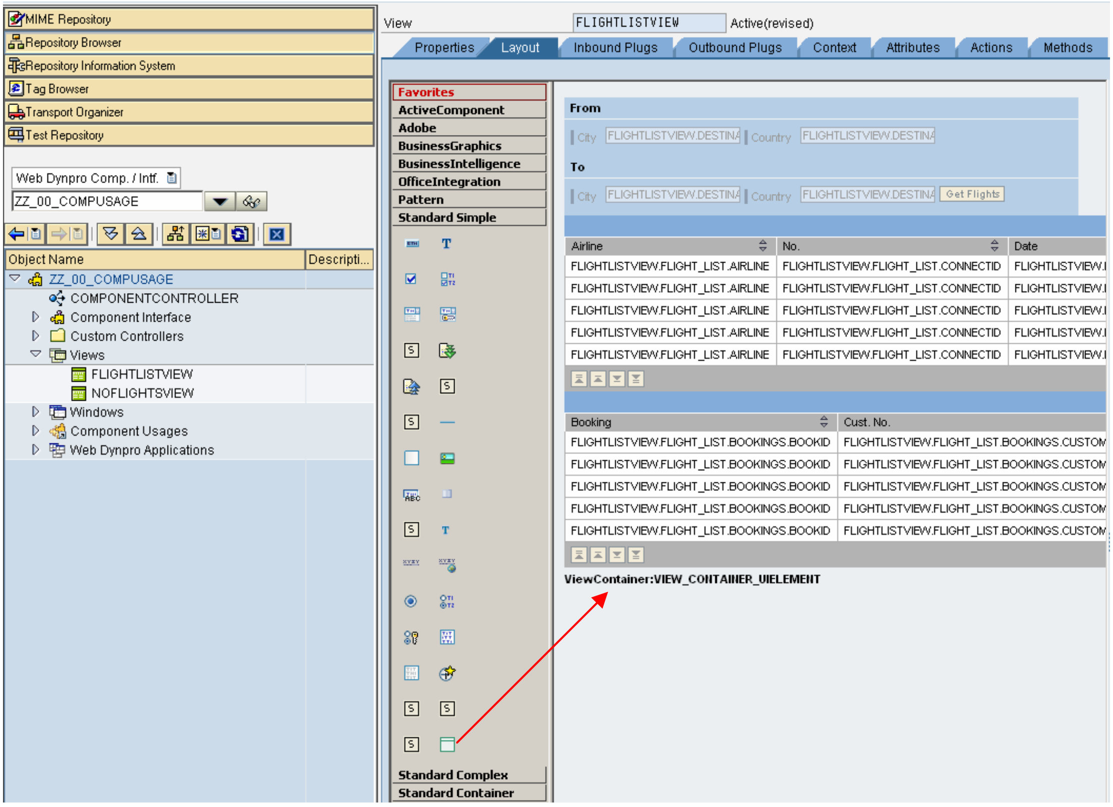
| | | | |
|---|---|---|---|
| Web Dynpro Component | ZZ_00_COMPUSAGE | Active | |
| Description | | | |
| Assistance Class | | | |
| Created By | BCUSER | Created On | 26.04.2006 |
| Last Changed By | BCUSER | Changed On | 26.04.2006 |
| Package | $TMP | AccessibilityChecks Active | |

Used Web Dynpro Components

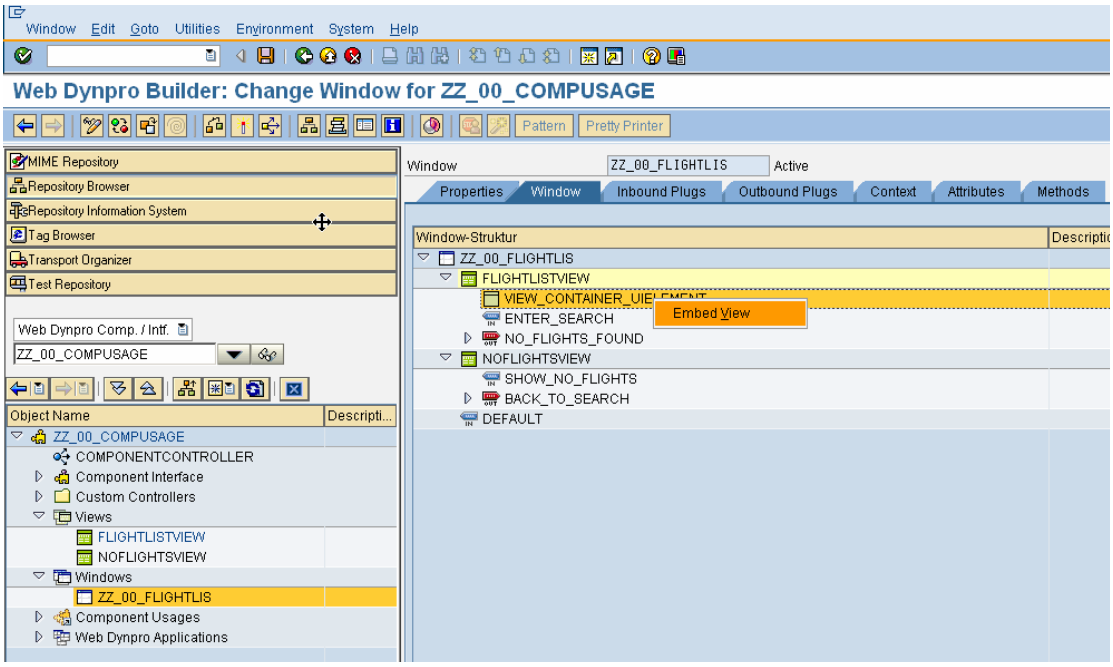
| Component Use | Component | Description of Compon |
|---|---|---|
| CUSTOMER_DATA | TECHED_05S_CUSTOMER_DATA | |

# Modify UIElement in Views and Embed the used Component

3. Switch to the view layout of FLIGHTLISTVIEW. Select from the Standard Simple UI library the *ViewContainerUIelement* which is the last element in the list and drag it to the layout of the view below the bookings table.
   A new ViewContainerUIelement named *ViewContainerUIelement_1* will be placed at the layout.



4. Switch to the Window of the component in the object list tree (the window's name may be different depending of the template used), open the node FLIGHTLISTVIEW at the right hand side and select *Embed View* by right clicking on the node *ViewContainerUIelement_1*.

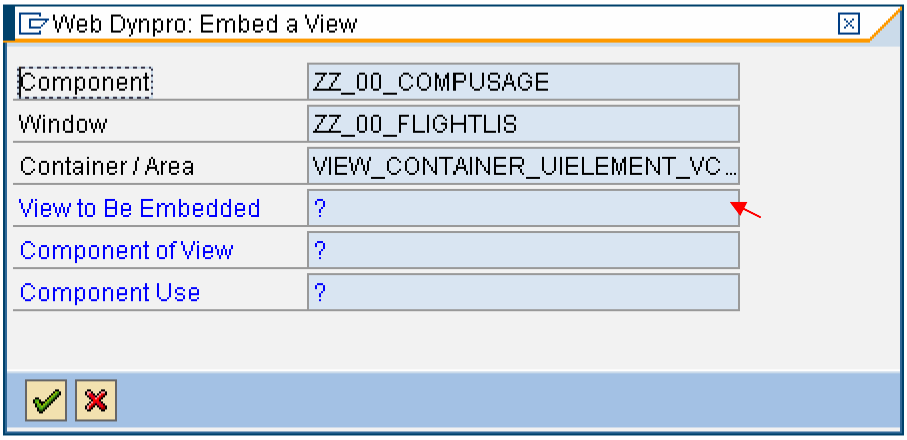


Web Dynpro Builder: Change Window for ZZ_00_COMPUSAGE

Web Dynpro Comp. / Intf.: ZZ_00_COMPUSAGE

Window: ZZ_00_FLIGHTLIS Active

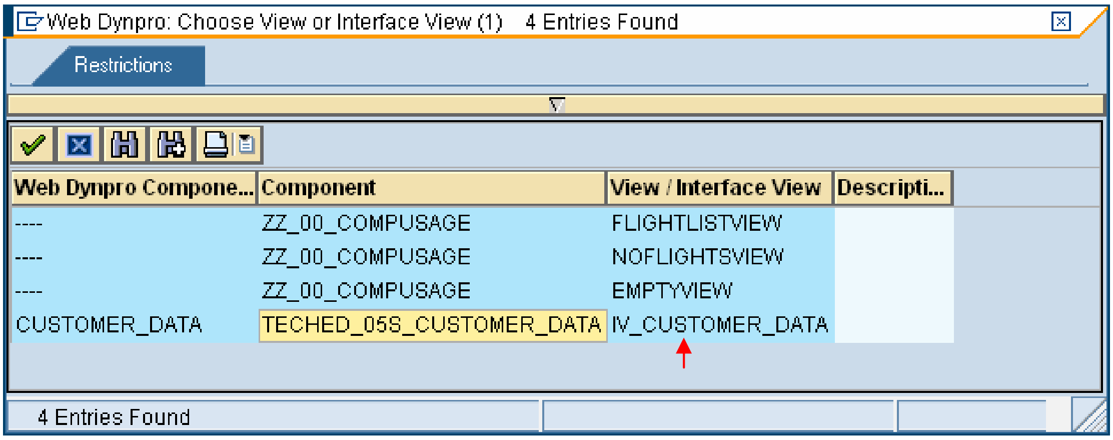
Select the Input help of the input field *View to be Embedded* of the appearing popup.



Web Dynpro: Embed a View

| | |
|---|---|
| Component | ZZ_00_COMPUSAGE |
| Window | ZZ_00_FLIGHTLIS |
| Container / Area | VIEW_CONTAINER_UIELEMENT_VC... |
| View to Be Embedded | ? |
| Component of View | ? |
| Component Use | ? |

Choose the interface view *IV_CUSTOMER_DATA* of the used component *CUSTOMER_DATA*.

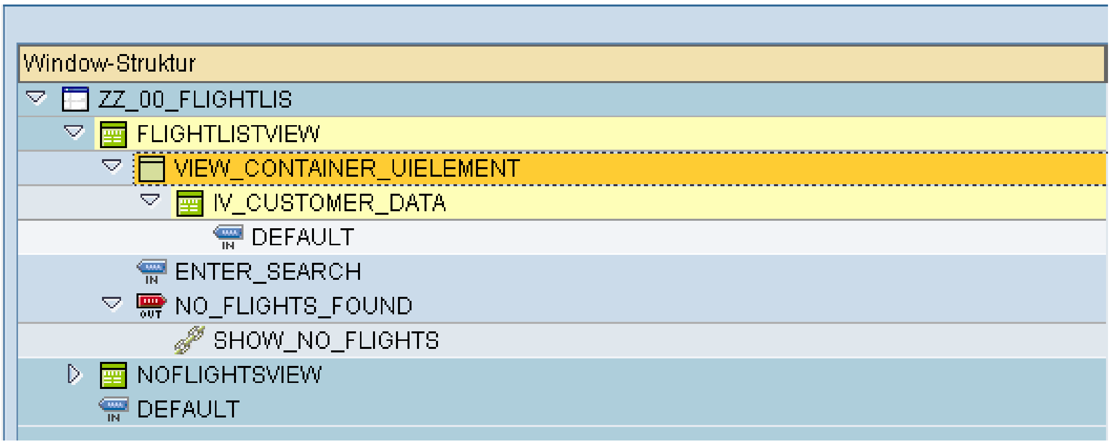


Web Dynpro: Choose View or Interface View (1)    4 Entries Found

| Web Dynpro Compone... | Component | View / Interface View | Descripti... |
|---|---|---|---|
| ---- | ZZ_00_COMPUSAGE | FLIGHTLISTVIEW | |
| ---- | ZZ_00_COMPUSAGE | NOFLIGHTSVIEW | |
| ---- | ZZ_00_COMPUSAGE | EMPTYVIEW | |
| CUSTOMER_DATA | TECHED_05S_CUSTOMER_DATA | IV_CUSTOMER_DATA | |

4 Entries Found

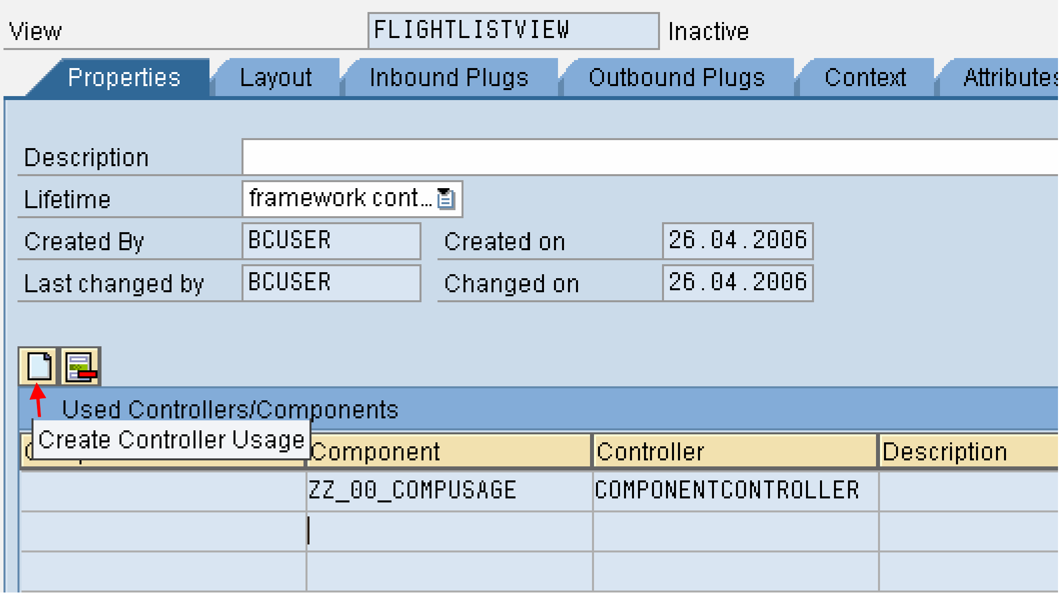
The window structure should now display the interface view *IV_CUSTOMER_DATA* within the ViewContainerUIelement in the FLIGHTLISTVIEW



```
Window-Struktur
▽ ZZ_00_FLIGHTLIS
    ▽ FLIGHTLISTVIEW
        ▽ VIEW_CONTAINER_UIELEMENT
            ▽ IV_CUSTOMER_DATA
                  DEFAULT
          ENTER_SEARCH
        ▽ NO_FLIGHTS_FOUND
              SHOW_NO_FLIGHTS
    ▷ NOFLIGHTSVIEW
      DEFAULT
```

# Declare and configure View controller and Interface controller

5. Before you can call the Interface Controller method of the used component you have to declare that the view controller can use the Interface Controller of the used component. By default only the component controller can access used components directly, view controllers have to specify this explicitly.
   Therefore go to the tab *Properties* of the FLIGHTLISTVIEW, click the button *Create Controller Use* and select the *INTERFACECONTROLLER* of the *CUSTOMER_DATA* component use.

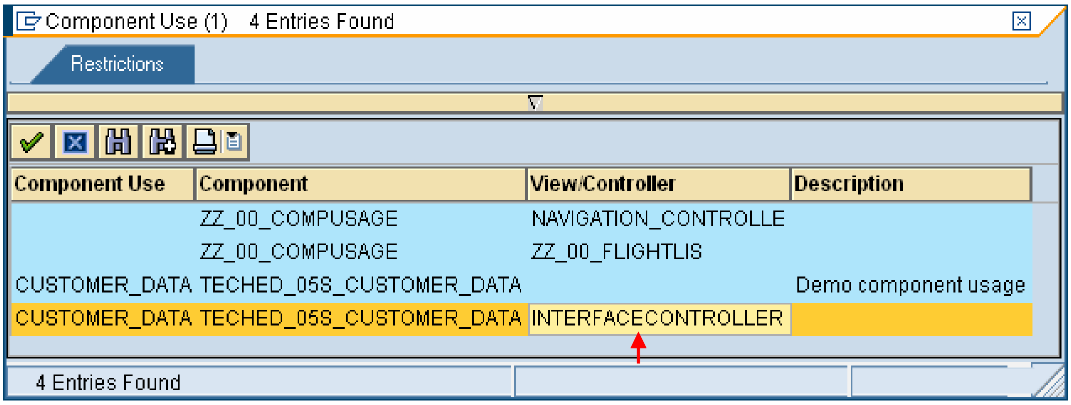


View FLIGHTLISTVIEW Inactive

| | | | |
|---|---|---|---|
| Description | | | |
| Lifetime | framework cont… | | |
| Created By | BCUSER | Created on | 26.04.2006 |
| Last changed by | BCUSER | Changed on | 26.04.2006 |

Used Controllers/Components

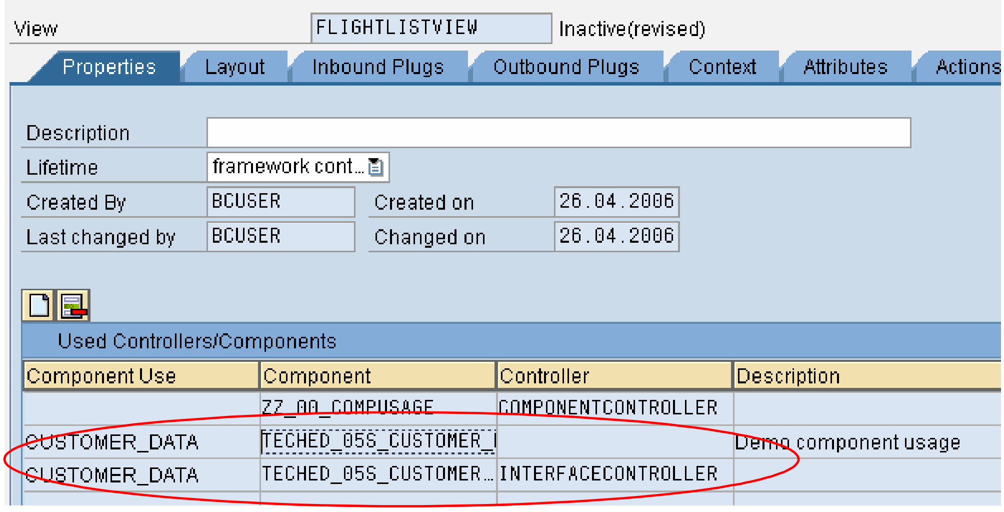
| C… | Component | Controller | Description |
|---|---|---|---|
| | ZZ_00_COMPUSAGE | COMPONENTCONTROLLER | |



Component Use (1) 4 Entries Found

| Component Use | Component | View/Controller | Description |
|---|---|---|---|
| | ZZ_00_COMPUSAGE | NAVIGATION_CONTROLLE | |
| | ZZ_00_COMPUSAGE | ZZ_00_FLIGHTLIS | |
| CUSTOMER_DATA | TECHED_05S_CUSTOMER_DATA | | Demo component usage |
| CUSTOMER_DATA | TECHED_05S_CUSTOMER_DATA | INTERFACECONTROLLER | |

4 Entries Found



View FLIGHTLISTVIEW Inactive(revised)

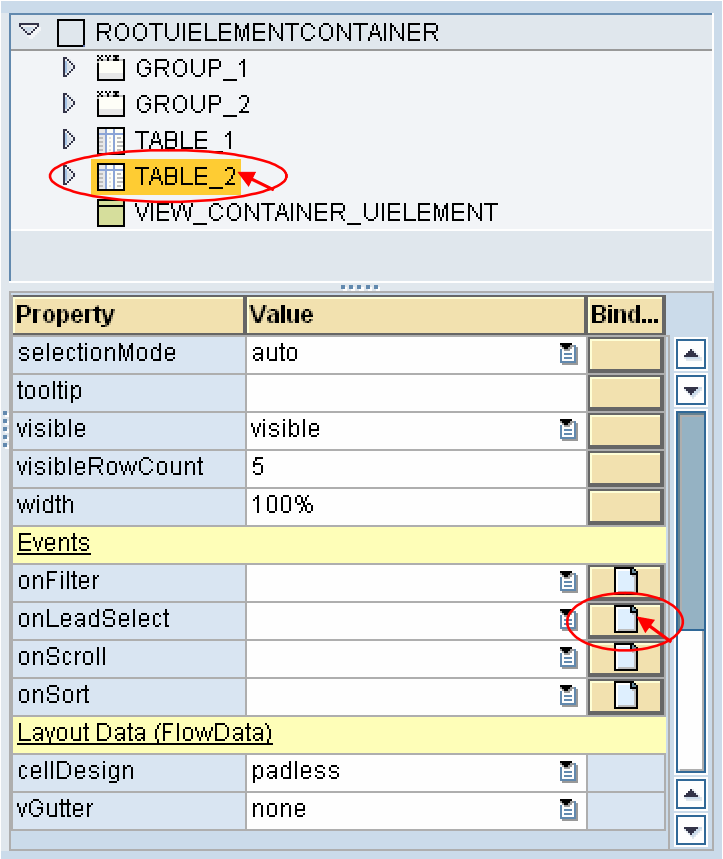
| | | | |
|---|---|---|---|
| Description | | | |
| Lifetime | framework cont… | | |
| Created By | BCUSER | Created on | 26.04.2006 |
| Last changed by | BCUSER | Changed on | 26.04.2006 |

Used Controllers/Components

| Component Use | Component | Controller | Description |
|---|---|---|---|
| | ZZ_00_COMPUSAGE | COMPONENTCONTROLLER | |
| CUSTOMER_DATA | TECHED_05S_CUSTOMER_I | | Demo component usage |
| CUSTOMER_DATA | TECHED_05S_CUSTOMER… | INTERFACECONTROLLER | |

## Create a new event for an existing UI element TABLE_2

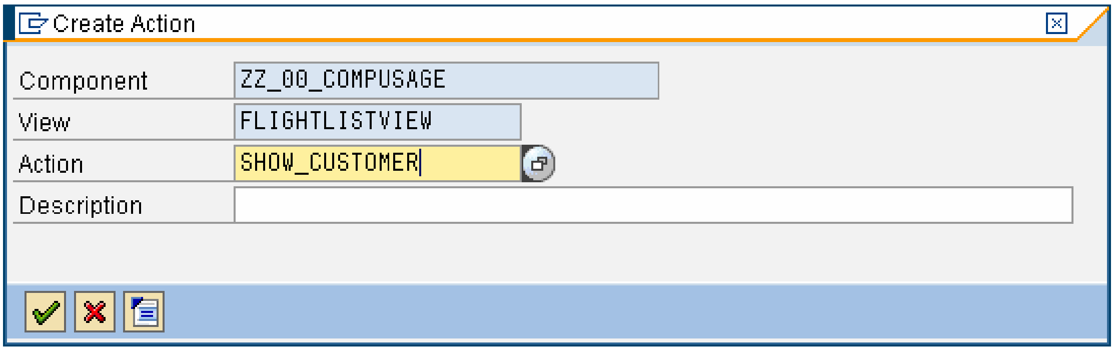
6. Switch to the Layout of the view FLIGHTLISTVIEW again, select the bookings table *TABLE_2* in the hierarchy at the top right and scroll to the Events in the properties of the TABLE_2 and create a new action for the *onLeadSelect* event by clicking the create button at the right hand side.



ROOTUIELEMENTCONTAINER
- GROUP_1
- GROUP_2
- TABLE_1
- TABLE_2
- VIEW_CONTAINER_UIELEMENT

| Property | Value | Bind... |
|---|---|---|
| selectionMode | auto | |
| tooltip | | |
| visible | visible | |
| visibleRowCount | 5 | |
| width | 100% | |
| Events | | |
| onFilter | | |
| onLeadSelect | | |
| onScroll | | |
| onSort | | |
| Layout Data (FlowData) | | |
| cellDesign | padless | |
| vGutter | none | |

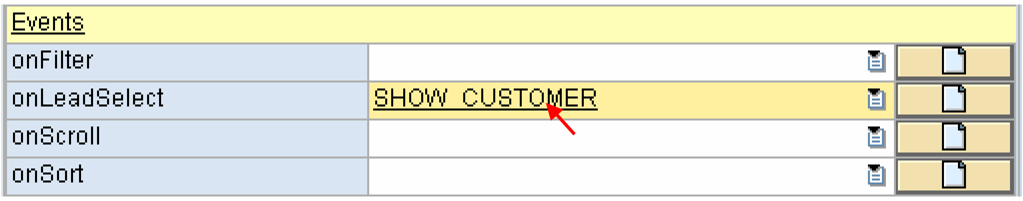
Name the new action *SHOW_CUSTOMER* and confirm.



Create Action

| | |
|---|---|
| Component | ZZ_00_COMPUSAGE |
| View | FLIGHTLISTVIEW |
| Action | SHOW_CUSTOMER |
| Description | |

Go to the events of the properties table of *TABLE_2* and double click the newly created action *SHOW_CUSTOMER* to switch to the source code of the corresponding method.



| Events | | |
|---|---|---|
| onFilter | | |
| onLeadSelect | SHOW_CUSTOMER | |
| onScroll | | |
| onSort | | |

This leads you to the source code editor of the method *ONACTIONSHOW_CUSTOMER*, which is triggered by the action *SHOW_CUSTOMER* whenever the lead selection of the bookings table is changed.

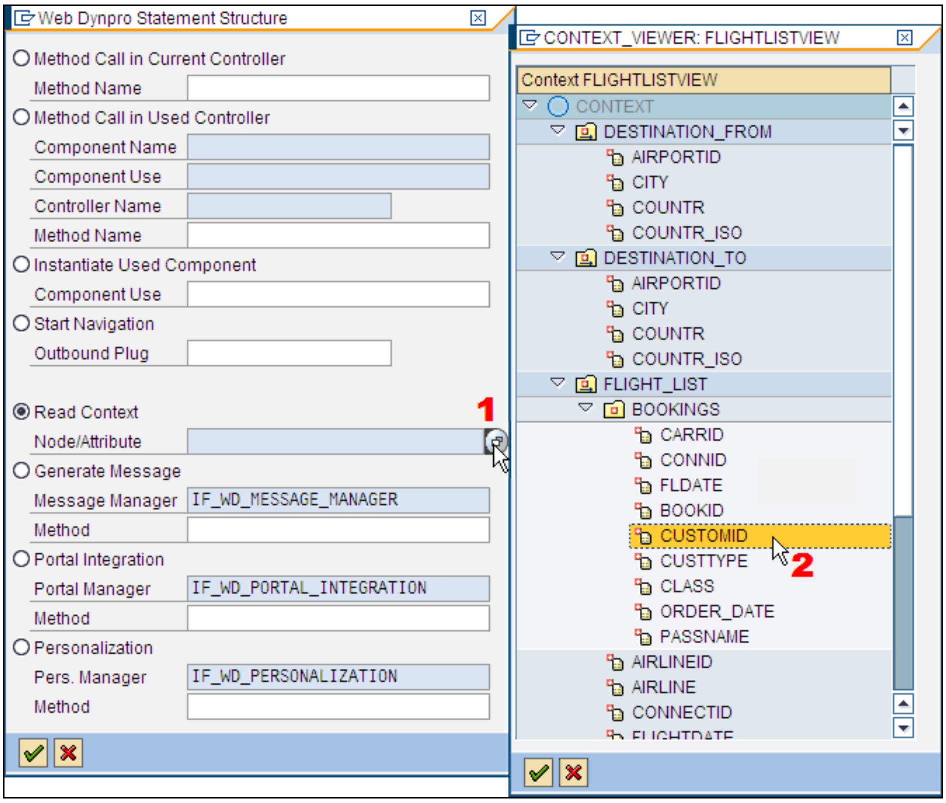
To display the corresponding customer data in the used component, the customer ID of the currently selected booking should be passed to the used component by calling its interface method *SHOWCUSTOMER*.

## Implement the method for the event

7. We implement the coding of the method *ONACTIONSHOW_CUSTOMER*

a) First you need the customer ID of the selected booking, which is stored in the corresponding context node. To get the value of the context node place the cursor in the method and use the code wizard by clicking .

Select *Read Context Node/Attribute* and click the input help (1).
Expand the BOOKINGS node in the appearing context viewer and select the attribute *CUSTOMID* (2) and confirm.



Web Dynpro Statement Structure

○ Method Call in Current Controller
Method Name
○ Method Call in Used Controller
Component Name
Component Use
Controller Name
Method Name
○ Instantiate Used Component
Component Use
○ Start Navigation
Outbound Plug
● Read Context
Node/Attribute
○ Generate Message
Message Manager: IF_WD_MESSAGE_MANAGER
Method
○ Portal Integration
Portal Manager: IF_WD_PORTAL_INTEGRATION
Method
○ Personalization
Pers. Manager: IF_WD_PERSONALIZATION
Method

CONTEXT_VIEWER: FLIGHTLISTVIEW

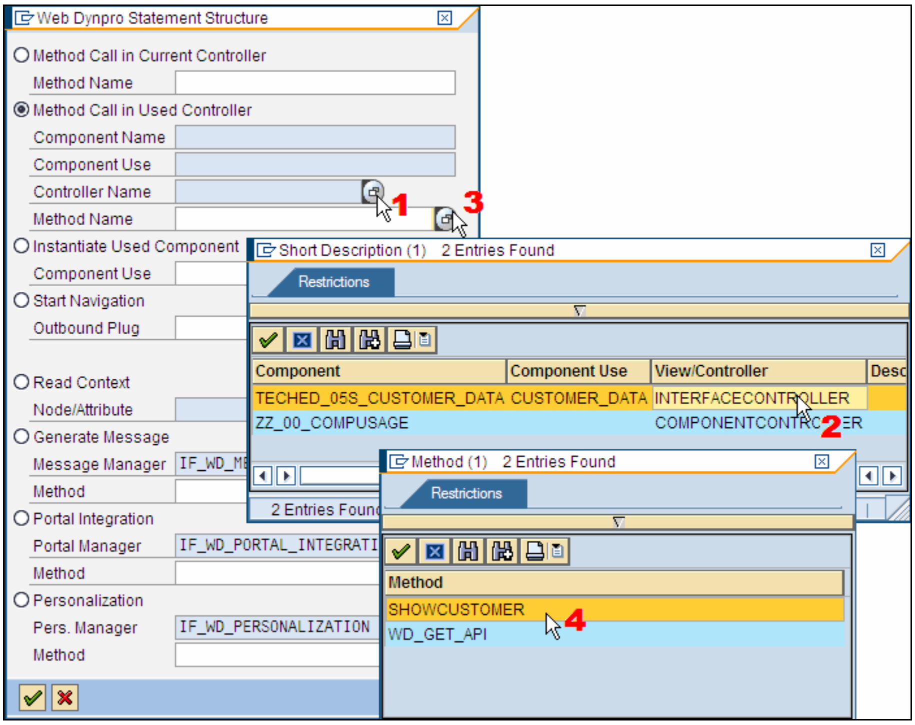
Context FLIGHTLISTVIEW
- CONTEXT
  - DESTINATION_FROM
    - AIRPORTID
    - CITY
    - COUNTR
    - COUNTR_ISO
  - DESTINATION_TO
    - AIRPORTID
    - CITY
    - COUNTR
    - COUNTR_ISO
  - FLIGHT_LIST
    - BOOKINGS
      - CARRID
      - CONNID
      - FLDATE
      - BOOKID
      - CUSTOMID
      - CUSTTYPE
      - CLASS
      - ORDER_DATE
      - PASSNAME
    - AIRLINEID
    - AIRLINE
    - CONNECTID
    - FLIGHTDATE

The wizard will generate the necessary coding to get the customer ID of the selected booking and stores it in a variable *Item_Customid* at the end of the generated code.

```
…
* get single attribute
  Elem_Bookings->get_Attribute(
    exporting
      Name =  `CUSTOMID`
    importing
      Value = Item_Customid ).
```

b) Position the cursor at the end of the generated code and call the code wizard again to generate the coding to call the Interface Controller method.

Choose Method Call in Used Controller (1)
Select the INTERFACECONTROLLER of the used component(2)
Select the input help of the Method input field (3) and
choose the *SHOWCUSTOMER* method (4)



Web Dynpro Statement Structure

○ Method Call in Current Controller
Method Name
● Method Call in Used Controller
Component Name
Component Use
Controller Name
Method Name
○ Instantiate Used Component
Component Use
○ Start Navigation
Outbound Plug
○ Read Context
Node/Attribute
○ Generate Message
Message Manager IF_WD_ME
Method
○ Portal Integration
Portal Manager IF_WD_PORTAL_INTEGRATI
Method
○ Personalization
Pers. Manager IF_WD_PERSONALIZATION
Method

Short Description (1) 2 Entries Found

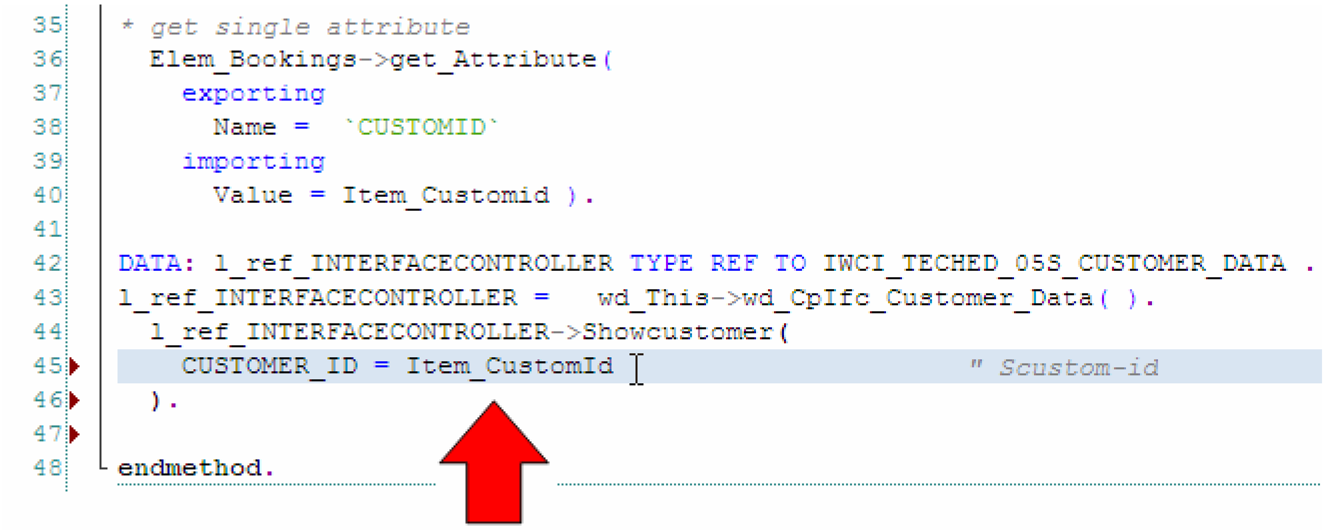
| Component | Component Use | View/Controller | Desc |
|---|---|---|---|
| TECHED_05S_CUSTOMER_DATA | CUSTOMER_DATA | INTERFACECONTROLLER | |
| ZZ_00_COMPUSAGE | | COMPONENTCONTROLLER | |

2 Entries Found

Method (1) 2 Entries Found

| Method |
|---|
| SHOWCUSTOMER |
| WD_GET_API |

The wizard will generate the coding for calling the Interface Controller's SHOWCUSTOMER method. Add the *Item_Customid* parameter to the call:



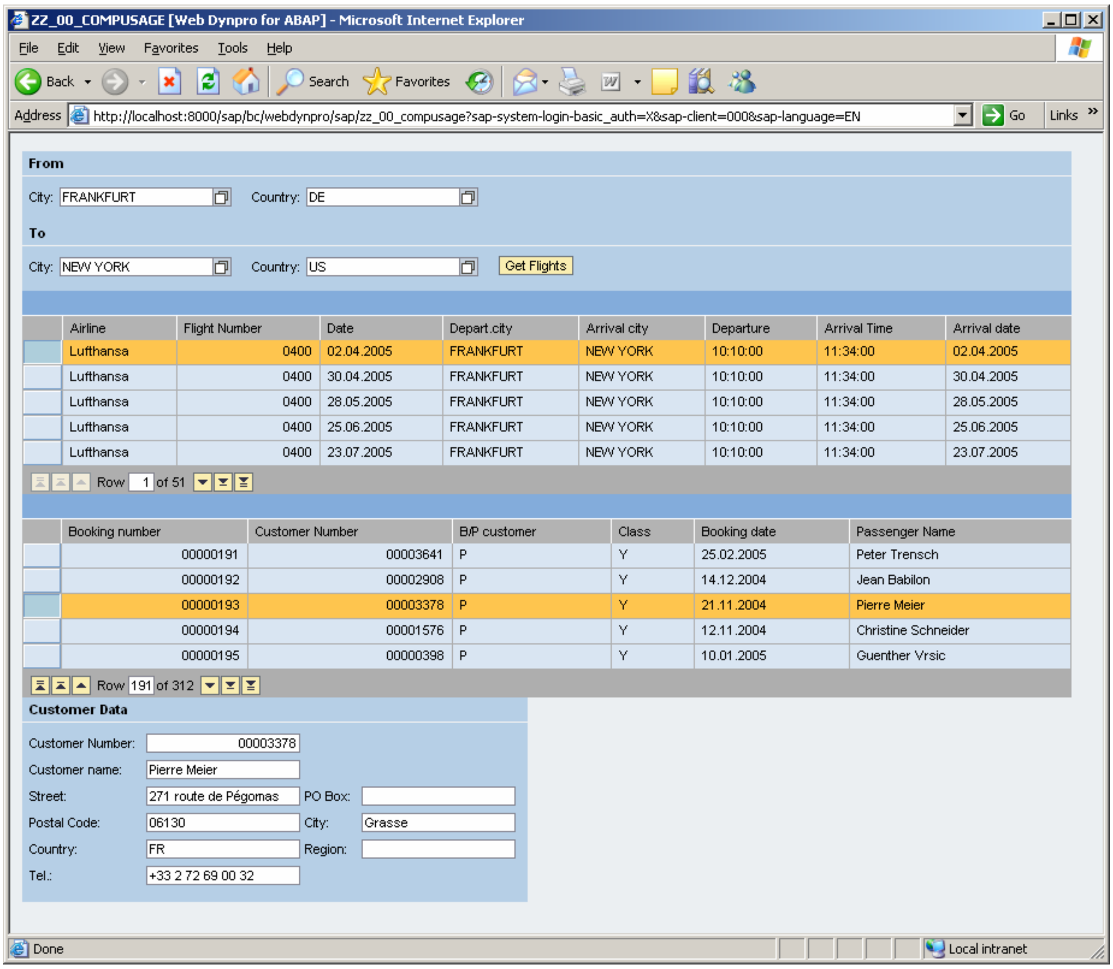
```
* get single attribute
  Elem_Bookings->get_Attribute(
    exporting
      Name =  `CUSTOMID`
    importing
      Value = Item_Customid ).

DATA: l_ref_INTERFACECONTROLLER TYPE REF TO IWCI_TECHED_05S_CUSTOMER_DATA .
l_ref_INTERFACECONTROLLER =   wd_This->wd_CpIfc_Customer_Data( ).
  l_ref_INTERFACECONTROLLER->Showcustomer(
    CUSTOMER_ID = Item_CustomId                          " Scustom-id
  ).

endmethod.
```

# Result: Test the Web Dynpro application with different customer records

8. Save, activate and generate everything and test the corresponding Web Dynpro application. Select different bookings and check that the corresponding customer records.

   You should now be able to

   - create and use a Configuration Controller,
   - steer application behaviour depending from the configuration attribute settings
   - use the Application and Component Configurators to set the values for configuration attributes


ZZ_00_COMPUSAGE [Web Dynpro for ABAP] - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Get Flights
Done
Local intranet

Address: http://localhost:8000/sap/bc/webdynpro/sap/zz_00_compusage?sap-system-login-basic_auth=X&sap-client=000&sap-language=EN

**From**

City: FRANKFURT Country: DE

**To**

City: NEW YORK Country: US

| Airline | Flight Number | Date | Depart.city | Arrival city | Departure | Arrival Time | Arrival date |
|---|---|---|---|---|---|---|---|
| Lufthansa | 0400 | 02.04.2005 | FRANKFURT | NEW YORK | 10:10:00 | 11:34:00 | 02.04.2005 |
| Lufthansa | 0400 | 30.04.2005 | FRANKFURT | NEW YORK | 10:10:00 | 11:34:00 | 30.04.2005 |
| Lufthansa | 0400 | 28.05.2005 | FRANKFURT | NEW YORK | 10:10:00 | 11:34:00 | 28.05.2005 |
| Lufthansa | 0400 | 25.06.2005 | FRANKFURT | NEW YORK | 10:10:00 | 11:34:00 | 25.06.2005 |
| Lufthansa | 0400 | 23.07.2005 | FRANKFURT | NEW YORK | 10:10:00 | 11:34:00 | 23.07.2005 |

Row 1 of 51

| Booking number | Customer Number | B/P customer | Class | Booking date | Passenger Name |
|---|---|---|---|---|---|
| 00000191 | 00003641 | P | Y | 25.02.2005 | Peter Trensch |
| 00000192 | 00002908 | P | Y | 14.12.2004 | Jean Babilon |
| 00000193 | 00003378 | P | Y | 21.11.2004 | Pierre Meier |
| 00000194 | 00001576 | P | Y | 12.11.2004 | Christine Schneider |
| 00000195 | 00000398 | P | Y | 10.01.2005 | Guenther Vrsic |

Row 191 of 312

**Customer Data**

| | | | |
|---|---|---|---|
| Customer Number: | 00003378 | | |
| Customer name: | Pierre Meier | | |
| Street: | 271 route de Pégomas | PO Box: | |
| Postal Code: | 06130 | City: | Grasse |
| Country: | FR | Region: | |
| Tel.: | +33 2 72 69 00 32 | | |

# SAP Online Help

More information on Component Usages can be found at SAP Help Portal under short link http://help.sap.com/saphelp_nw04s/helpdata/en/7c/3545415ea6f523e10000000a155106/frameset.htm or via path help.sap.com → Documentation → SAP NetWeaver → SAP NetWeaver 2004s → English → SAP NetWeaver Library → SAP NetWeaver by Key Capability → Application Platform by Key Capability → ABAP Technology → UI Technology → Web UI Technology → Web Dynpro for ABAP → Web Dynpro for ABAP: Development in Detail → Cross Component Programming → Component Usages

General information on Web Dynpro for ABAP can be found at the SAP Help Portal under the short link http://help.sap.com/saphelp_nw04s/helpdata/en/77/3545415ea6f523e10000000a155106/frameset.htm or via path help.sap.com → Documentation → SAP NetWeaver → SAP NetWeaver 2004s → English → SAP NetWeaver Library → SAP NetWeaver by Key Capability → Application Platform by Key Capability → ABAP Technology → UI Technology → Web UI Technology → Web Dynpro for ABAP.