IBM Cúram Social Program Management
Version 6.0.5

Working with the Cúram User Interface

Note

Before using this information and the product it supports, read the information in “Notices” on page 41

Revised: March 2014

This edition applies to IBM Cúram Social Program Management v6.0.5 and to all subsequent releases unless otherwise indicated in new editions.

Licensed Materials - Property of IBM.

© Copyright IBM Corporation 2012, 2014.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© Cúram Software Limited. 2011. All rights reserved.

Contents

© Copyright IBM Corp. 2012, 2014

Figures

© Copyright IBM Corp. 2012, 2014

Tables

© Copyright IBM Corp. 2012, 2014

Working with the Cúram User Interface

Use this information to develop user interface elements with the Cúram Client Development Environment for Java. User interface elements that can be created with the Cúram Client Development Environment for Java include shortcut panels, tabs, modal dialogs, tab navigation, and lists.

Introduction

Purpose

The purpose of this guide is as an aid to the *Cúram Web Client Reference Manual*. The guide illustrates how to use the features of the Cúram Client Development Environment for Java™ (CDEJ), as outlined in the reference manual. It will bring the reader through the creation of a simple client application and expand on this application with more complex features in subsequent chapters.

The key features of the CDEJ, i.e., the user interface elements, are highlighted on the images used throughout this guide. The text on the images is intentionally blurred so that only the key elements are in focus. Note that for cross referencing purposes, each image contains numbers that maps to specific user interface elements. For example, in Figure 1 on page 2 the number 1 maps to a tab which is later referenced in other sections as "User Interface Element 1".

Audience

This guide is intended for developers of Cúram applications, with specific focus on development of the Cúram user interface.

Prerequisites

An understanding of development using both the Cúram Client Development Environment for Java (CDEJ) and the Cúram Server Development Environment for Java (SDEJ), as covered in the *Web Client Reference Manual* and *Cúram Server Developer's Guide*, is required.

In addition, a basic understanding of Java Platform, Enterprise Edition (Java EE) development environments, Extensible Markup Language (XML), and web technologies such as Hypertext Transfer Protocol (HTTP), JavaServer Pages (JSP), Cascading Style Sheets (CSS), and JavaScript would be useful.

It is assumed that the necessary steps to install the Cúram application and the related third-party tools have been completed. For more information, see the *Cúram Installation Guide* and the *Cúram Third Party Tools Installation Guides*.

Finally, it is recommended that all developers are familiar with the *Cúram User Experience Guidelines.*

Chapters in this Guide

The following list describes the chapters within this guide:

Create a Simple Application

This chapter introduces what an application is and how to create one. At the end of this chapter you will have created a simple application with one

© Copyright IBM Corp. 2012, 2014

section and one tab, as can be seen in Figure 1.

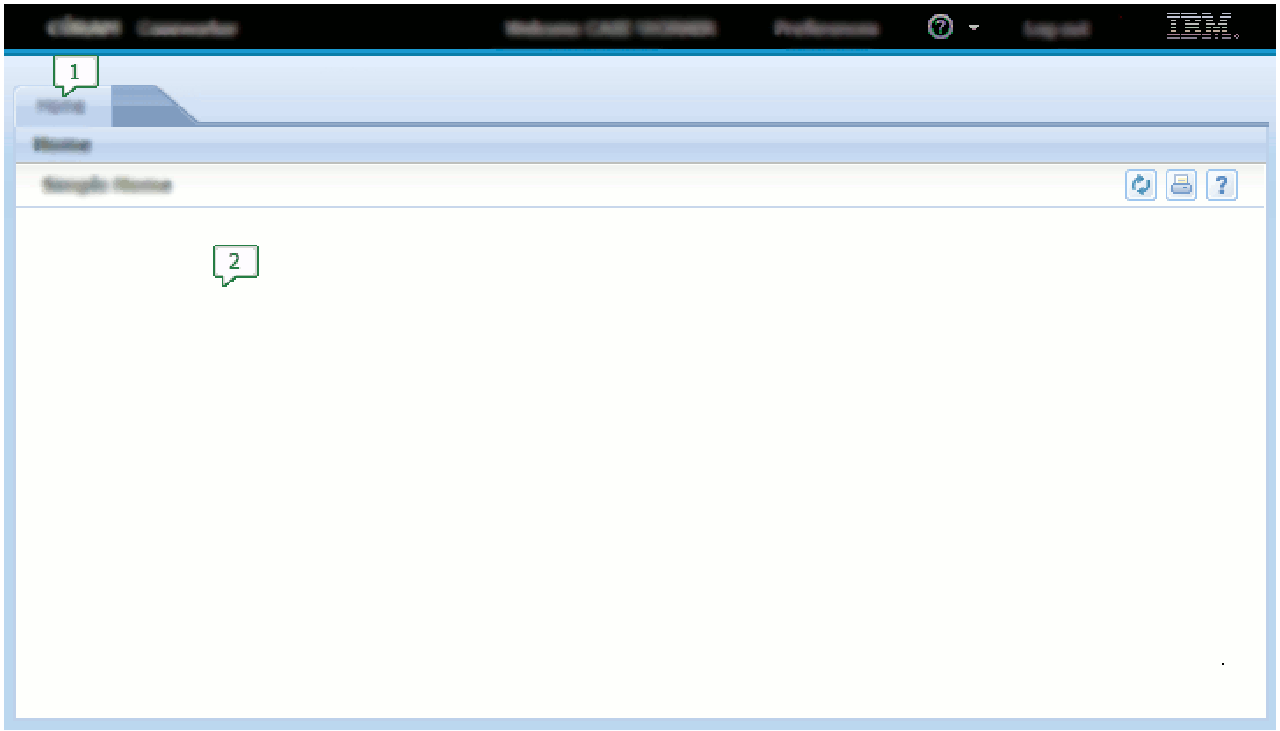

Figure 1. A Simple Application

1. Section
2. Tab

Adding a Shortcut Panel

This chapter introduces adding a shortcut panel to a section, as can be seen in Figure 2 on page 3.

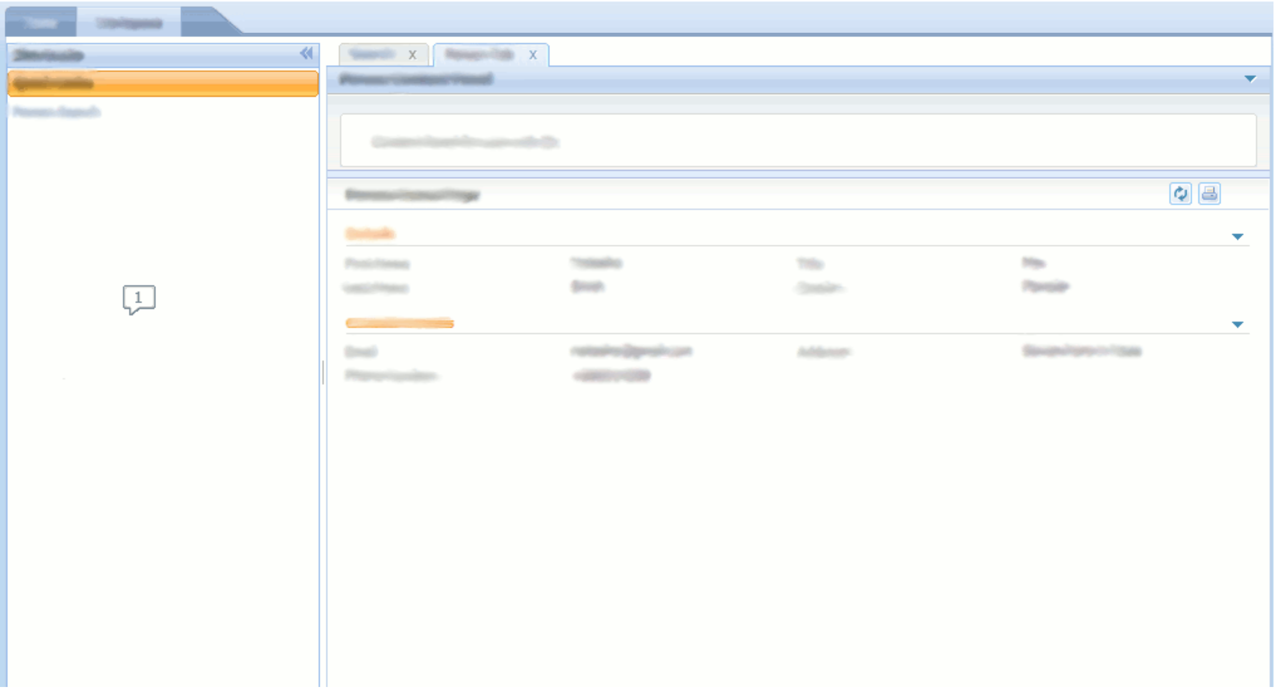

Figure 2. Shortcut Panel

1.Shortcut Panel

Adding Tab Content

This chapter introduces more complex structured tabs with the tab content of a context panel and a content area, as can be seen in Figure 3.

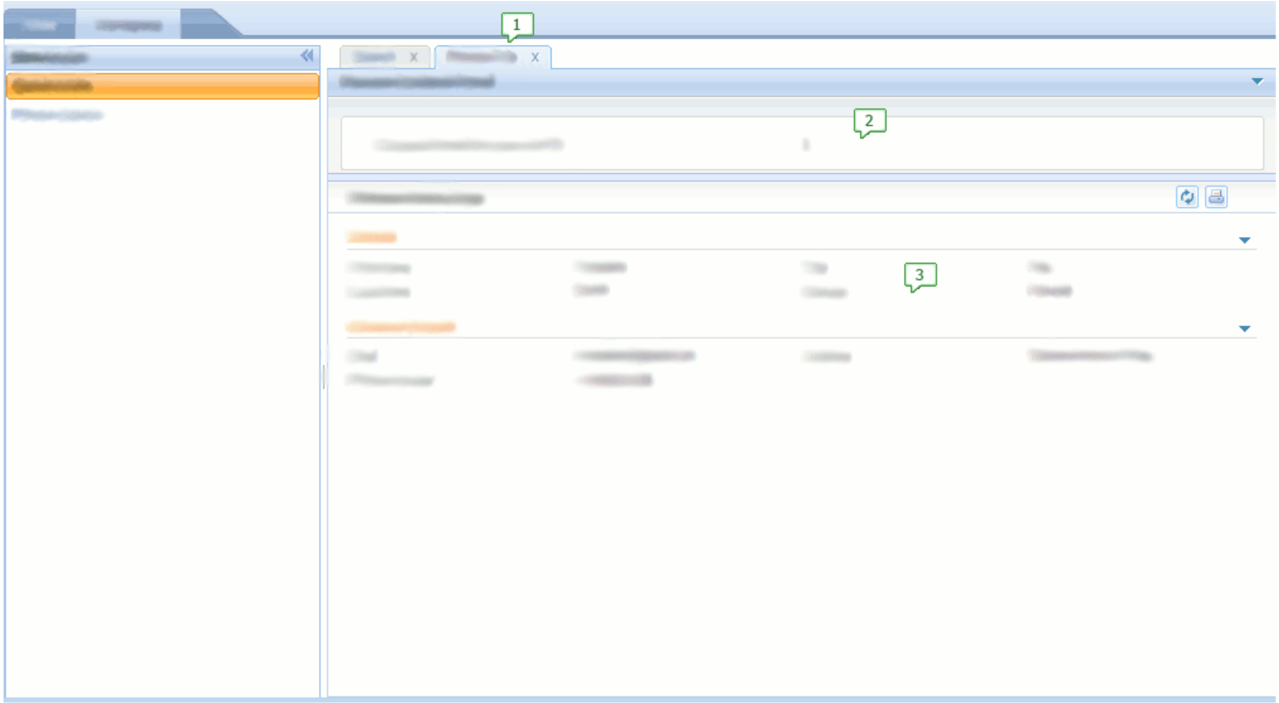

Figure 3. Tab with Context Panel

1. Tab

2. Context Panel
3. Content Area

Working with Modal Dialogs and Wizard Progress Bars

This chapter will introduce the ability to modify content in the Cúram application through modal dialogs and by incorporating wizard progress bars. An example of a modal dialog with a wizard progress bar can be seen in Figure 4.

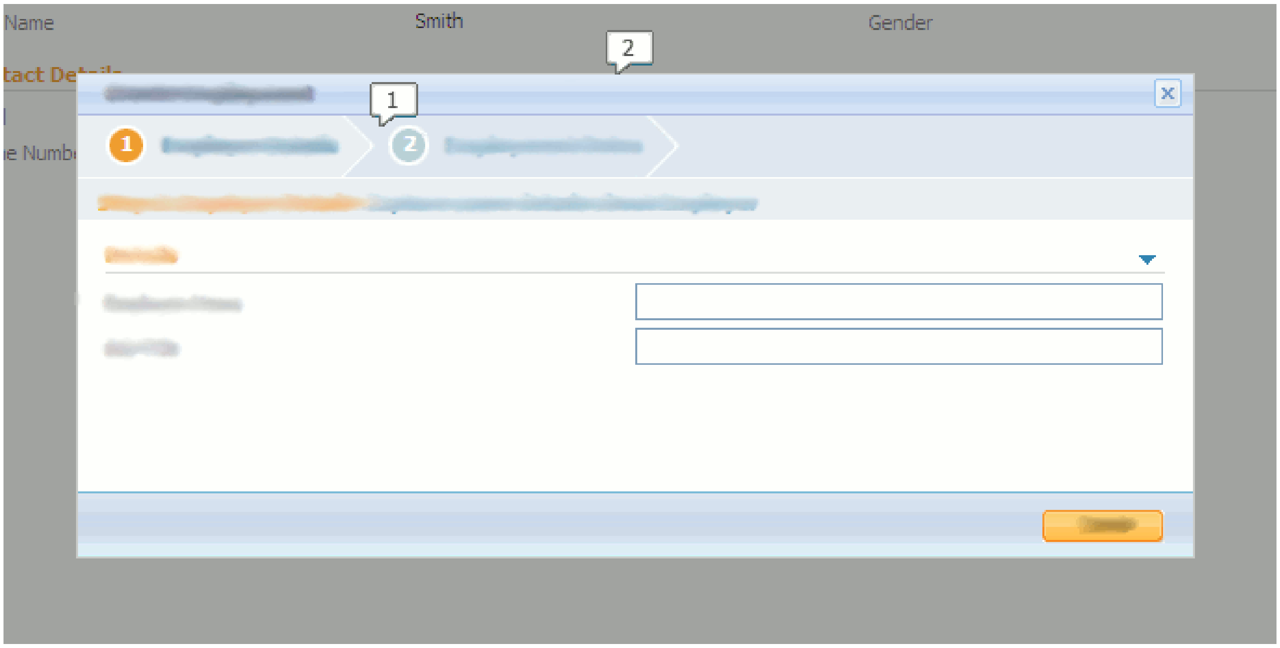

Figure 4. Modal Dialog with Wizard Progress Bar

1. Wizard Progress Bar
2. Modal Dialog

Adding Tab Navigation

This chapter will detail how to add navigation features to a tab. These features are the content area navigation bar and the page group navigation bar, as can be seen in Figure 5 on page 5

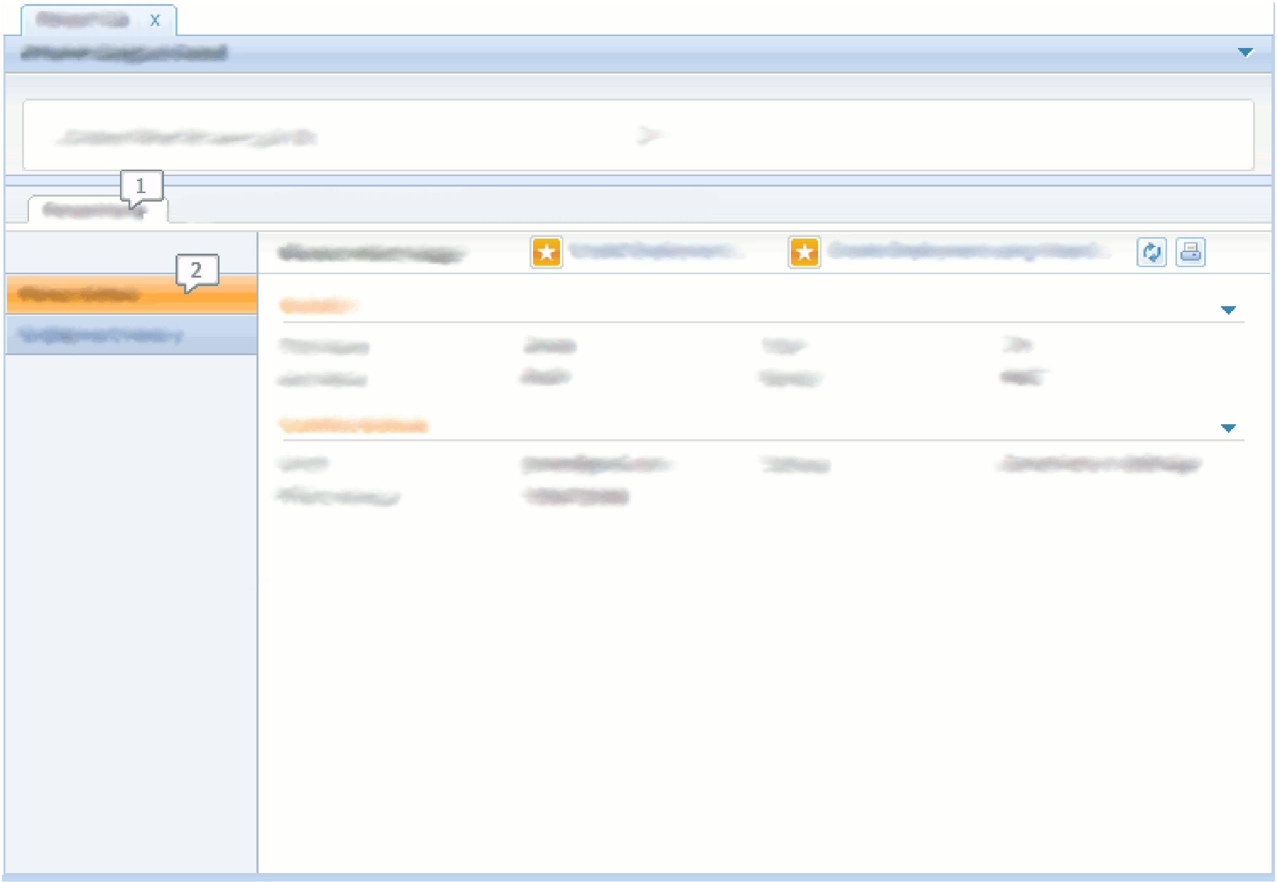

Figure 5. Navigation

1. Content Area Navigation Bar
2. Page Group Navigation Bar

Working with Lists

This chapter introduces expandable lists and list action menus. An example of an expandable list with list action menus can be seen in Figure 6 on page 6.

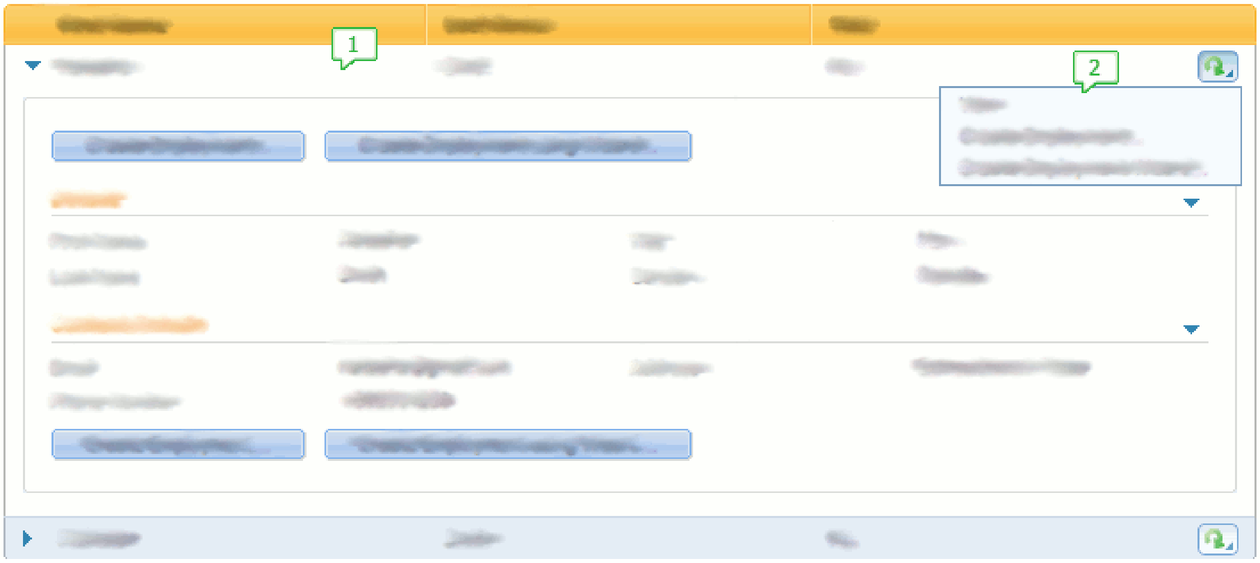

Figure 6. Expandable List with List Action Menu

1. Expandable List
2. List Actions Menu

Creating a Simple Application

Introduction

This chapter will introduce the various concepts for creating a very simple application with a single section and a single page of content. "Introduction" shows a screenshot of what this application will look like.

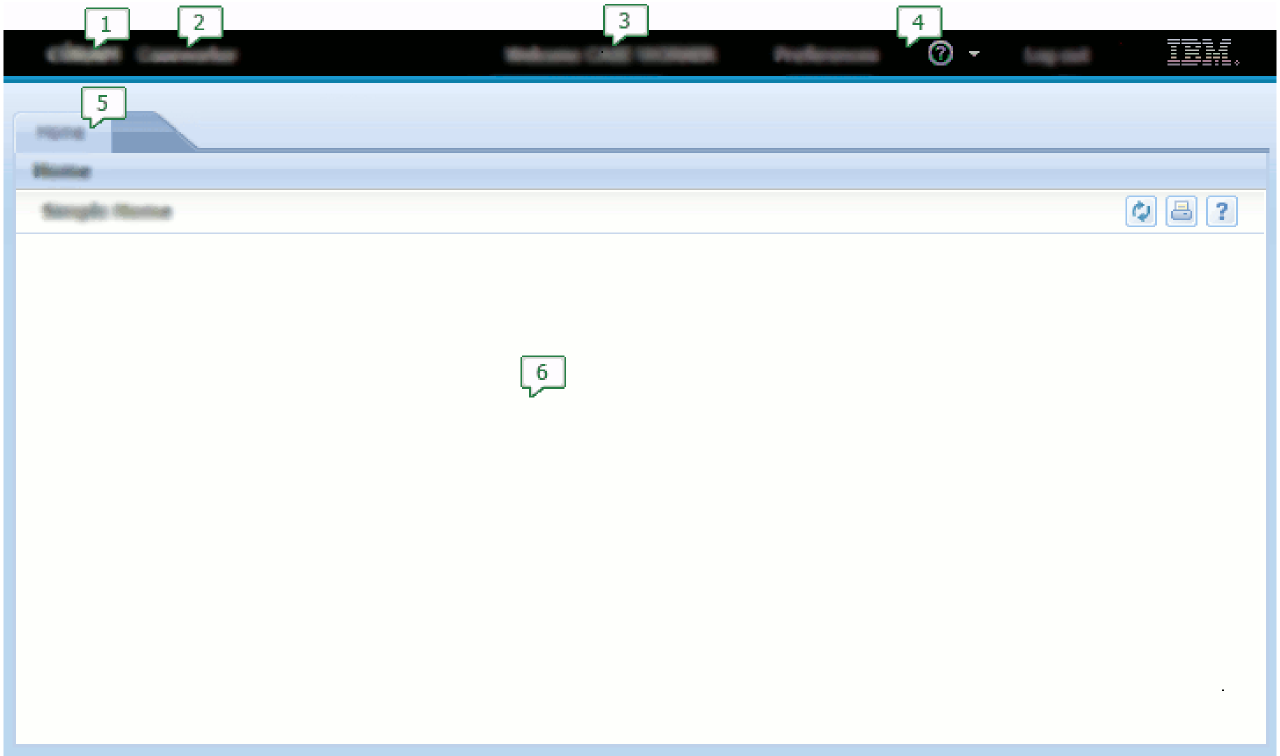

Figure 7. A Simple Application

1. Application Name
2. Application Subtitle
3. Welcome Message
4. Application Menu
5. Section
6. Tab

After installing the Cúram application and the related third party tools there are two main projects that are used for development, the `EJBServer` project and the `webclient` project. To create a simple application requires creating and modifying files in the following directories:

- `webclient\components\<component-name>\`
- `EJBServer\components\<component-name>\clientapps`
- `EJBServer\components\<component-name>\codetable`
- `EJBServer\components\<component-name>\Data_Manager`
- `EJBServer\project\config`

Where <component-name> is the name of the custom component used to store customer specific content to the Cúram application.

Define an Application

An application is a particular view of the Cúram client defined for a specific user or role, refer to "Introduction" on page 6. It is defined using an XML configuration file with the extension `.app`. The `.app` files, are located in the `EJBServer\components\<component-name>\clientapps` directory, where <component-name> is a custom component. "Define an Application" details a very simple application that contains a single section (will be covered later).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
   id="SimpleApp"
   logo="SimpleApp.logo"
   title="SimpleApp.title"
   subtitle="SimpleApp.subtitle"
   user-message="SimpleApp.UserMessage">

  <ac:application-menu>
    <ac:preferences title="preferences.title"/>
    <ac:help title="help.title"/>
    <ac:logout title="logout.title"/>
  </ac:application-menu>

  <ac:section-ref id="SimpleAppHomeSection"/>

</ac:application>
```

Figure 8. SimpleApp app

The `SimpleApp.app` should have a corresponding `SimpleApp.properties` file, which details the localizable content for the application. For example:

```
SimpleApp.logo=CDEJ/themes/v6/images/application-logo.png
SimpleApp.title=C\u00FAram
SimpleApp.subtitle=Simple Application
SimpleApp.UserMessage=Welcome, %user-full-name

preferences.title=User Preference
help.title=Help
logout.title=Logout
```

`SimpleApp.app` has configured the following elements in the application banner of the application:

- an application name (`title`), refer to User Interface Element 1 in "Introduction" on page 6;
- an application subtitle (`subtitle`), refer to User Interface Element 2 in "Introduction" on page 6;
- a welcome message (`user-message`), refer to User Interface Element 3 in "Introduction" on page 6;
- a application menu (`application-menu`), refer to User Interface Element 4 in "Introduction" on page 6.

Add a Section to the Application

An application can define between one and five sections, refer to User Interface Element 5 in "Introduction" on page 6. Each section supports displaying multiple object tabs (will be covered later). `SimpleApp.app` has referenced one section using the id `SimpleAppHomeSection`. This id refers to a section configuration file, which is an XML configuration file with the extension `.sec`. Like `SimpleApp.app`, this file will be added to the `EJBServer\components\<component-name>\clientapps` directory and the `id` attribute must match the name of the file.

"Add a Section to the Application" details an example section file, `SimpleAppHomeSection.sec`.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<sc:section
   id="SimpleAppHomeSection"
   title="Section.Home.Title"
   hide-tab-container="true">

  <sc:tab id="SimpleHome"/>

</sc:section>
```

Figure 9. SimpleAppHomeSection.sec

The `SimpleAppHomeSection.sec` file will have a corresponding `SimpleAppHomeSection.properties` file, which details the localizable content. For example:

```
Section.Home.Title=Home
```

The `title` attribute defines the name of the section tab. In addition, as there is only one tab defined for this section (SimpleHome), the `hide-tab-container` attribute has been used to hide the object tab bar.

Add a Tab to a Section

A tab typically represents a business object, e.g. a Case or a Participant, though it can also be used to represent a logical grouping of information, refer to User Interface Element `6` in "Introduction" on page 6. `SimpleAppHomeSection.sec` has referenced one tab, using the id `SimpleHome`. This id refers to a tab configuration file, which is an XML configuration file with the extension `.tab`. Like `.app` and `.sec` files, this file will be added to the `EJBServer\components\<component-name>\clientapps` directory and the `id` attribute must match the name of the file.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<tc:tab-config
    id="SimpleHome">

  <tc:context tab-name="home.tab.name"
                 tab-title="home.tab.name"/>

  <tc:navigation page-id="SimpleHome"/>

</tc:tab-config>
```

Figure 10. SimpleHome.tab

The `SimpleHome.tab` file will have a corresponding `SimpleHome.properties` file, which details the localizable content. For example:

```
home.tab.name=Home
```

The `tab-title` attribute defines what will be displayed on the tab title bar. As we have turned off the object tab bar in the `.sec` file, the `tab-name` attribute is ignored.

`SimpleHome.tab` references a single UIM page (will be covered later) using the `page-id` attribute of the `navigation` element.

Add UIM Page to a Tab

`SimpleHome.tab` has referenced the `SimpleHome` UIM page. Page content is defined using files, with the extension `.uim` and unlike the application configuration files to date, the `.uim` files are located in the `webclient\components\<component-name>`

directory.

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- This is a sample home page. -->
<PAGE PAGE_ID="SimpleHome">

  <PAGE_TITLE>
    <CONNECT><SOURCE NAME="TEXT"
                       PROPERTY="PageTitle.StaticText"/>
    </CONNECT>
  </PAGE_TITLE>

</PAGE>
```

Figure 11. SimpleHome.uim

The `SimpleHome.uim` file will have a corresponding `SimpleHome.properties` file, which details the localizable content. For example:

```
PageTitle.StaticText=Simple Home
```

The UIM page defined by `SimpleHome.uim` has no main content and only a page title (`PAGE_TITLE`). “Add UIM Page to a Tab” on page 9 shows shows what this content will look like, including a number of common actions (print, help and refresh) that exist on most UIM pages.

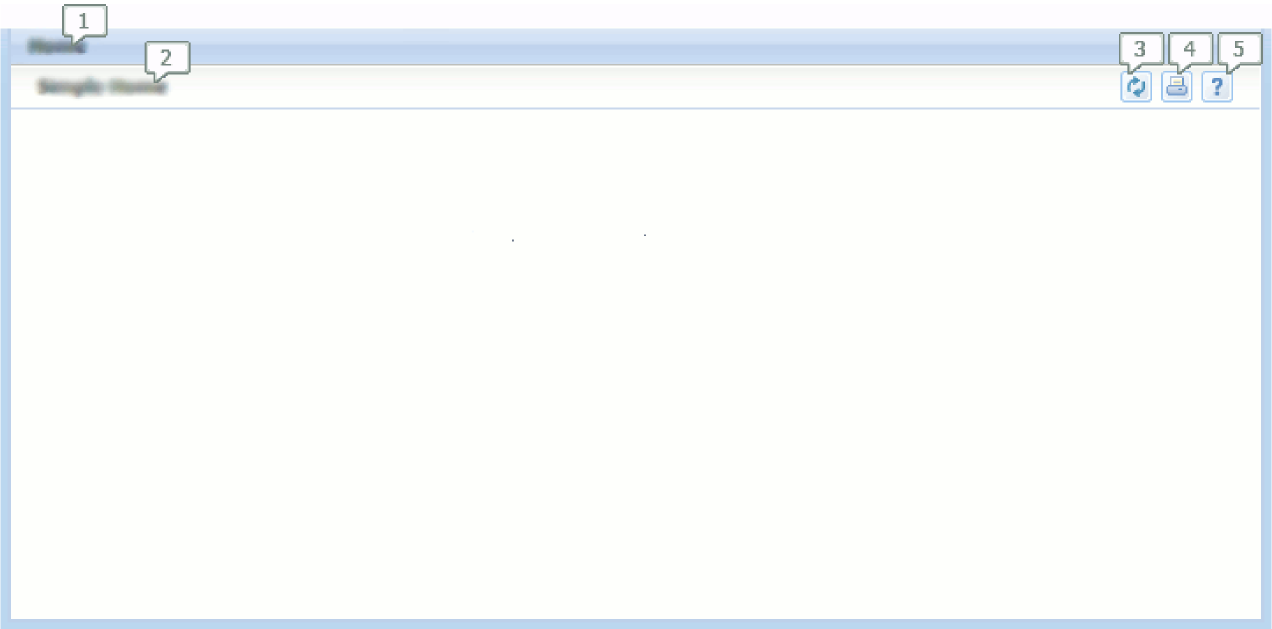

Figure 12. A Simple Tab with Basic Content

1. Tab Title
2. Page Title
3. Refresh Button
4. Print Button
5. Help Button

Associate a User with an Application

Having created the content for the simple application, a user user must be created and linked to this application. A user exists as an entry on the Users database tab. A new user can be created using a dmx file, which is added to the EJBServer\components\<component-name>\Data_Manager directory.

```
<table name="USERS">

  ...

  <row>
    <attribute name="USERNAME">
      <value>simple</value>
    </attribute>
    ...
    <attribute name="ROLENAME">
      <value>SUPERROLE</value>
    </attribute>
    <attribute name="APPLICATIONCODE">
      <value>SimpleApp</value>
    </attribute>
    ...
    <attribute name="DEFAULTLOCALE">
      <value>en</value>
    </attribute>
    <attribute name="FIRSTNAME">
      <value>Simple</value>
    </attribute>
    <attribute name="SURNAME">
      <value>User</value>
    </attribute>
  </row>

</table>
```

Figure 13. Users.dmx

The dmx file, Users.dmx, must be referenced in the datamanager_config.xml file located in the EJBServer\project\config directory. For example:

```
<entry name="components/custom/Data_Manager/USERS.dmx"
          type="dmx" base="basedir"/>
```

Once the entry is referenced from this file, it will be included on the database when the **database** target is executed.

“Associate a User with an Application” details the creation of a single user, simple, with a password of password. The field used for linking this user to a particular application is the APPLICATIONCODE field. This field references a code table entry in the APPLICATION_CODE code table. When a user logs in, the value of the APPLICATIONCODE field in the Users database table is used to determine both the application and the page to display. The latter is known as the user's home page; the first page that the user sees on log in. The value of the code matches the name of the application (.app) file to use and the description of the code value indicates the name of the UIM page to be displayed as the home page.

“Associate a User with an Application” details a CT_APPLICATION_CODE.ctx file, that should be located in the EJBServer\components\<component-name>\codetable directory.

```
<?xml version="1.0"?>
<codetables package="curam.util.testmodel.codetable">
  <codetable java_identifier="APPLICATION_CODE"
              name="APPLICATION_CODE">
    <code default="false" java_identifier="SIMPLE_HOME"
        status="ENABLED" value="SimpleApp">
      <locale language="en" sort_order="0">
        <description>SimpleHome</description>
        <annotation></annotation>
      </locale>
    </code>
  </codetable>
</codetables>
```

Figure 14. CT_APPLICATION_CODE.ctx

In this example, a code `SimpleApp` has been defined, with a description of `SimpleHome`. The code `SimpleApp`, matches the id of the `SimpleApp.app` example. The description, `SimpleHome`, maps to the `SimpleHome.uim` file.

Summary

The addition and modification of the above files requires a number of build targets to be executed. A summary of the files and which targets will process each of them is detailed below.

Table 1. Columns

File	Location	Target
SimpleApp.app and associated properties file	`EJBServer\components\<component-name>\clientapps`	inserttabconfiguration
SimpleAppHomeSection.sec and associated properties file	`EJBServer\components\<component-name>\clientapps`	inserttabconfiguration
SimpleHome.tab and associated properties file	`EJBServer\components\<component-name>\clientapps`	inserttabconfiguration
SimpleHome.uim and associated properties file	`webclient\components\<component-name>\`	client
Users.dmx	`EJBServer\components\<component-name>\Data_Manager`	database
datamanager_config.xml	`EJBServer\project\config`	database
CT_APPLICATION _CODE.ctx	`EJBServer\components\<component-name>\codetable`	server

Note: The **`inserttabconfiguration`** target is included in the **`database`** target.

Once all build targets have been completed and the server and client applications have been started, the application can be accessed using the following URL:

```
http://localhost:9080/'server_name'/AppController.do
```

To view the simple application, login as the `simple` user, with the password of `password`.

Adding a Shortcut Panel

Introduction

This chapter will introduce how to add a shortcut panel to a section.

The application in “Creating a Simple Application” on page 6 will be extended with a new section, which includes an example of a Shortcut Panel.

“Introduction” shows this new section, named as `Workspace` section, displaying the shortcut panel expanded.

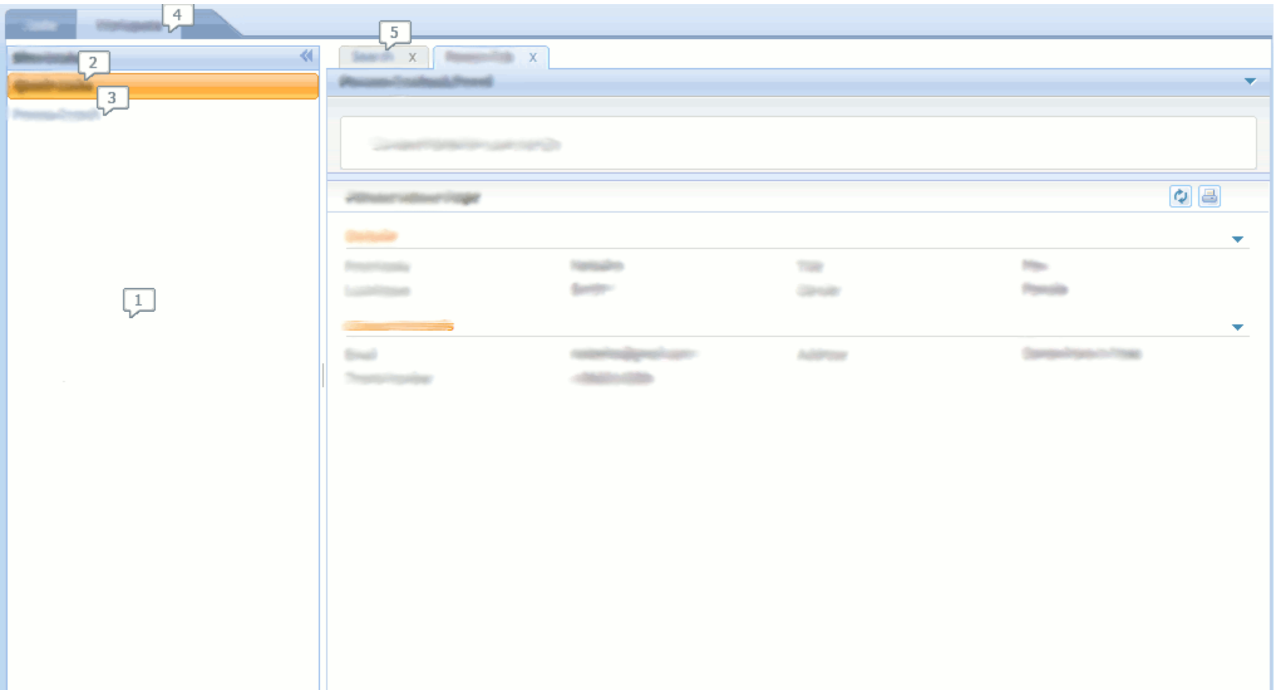

Figure 15. Shortcut Panel

1. Shortcut Panel
2. Group Node
3. Leaf Node
4. Workspace Section
5. Search Tab

Add a new Section

“Add a new Section” includes the addition of a new section, the `Workspace` Section, refer to User Interface Element 4 in “Introduction.”

```
        <ac:application
  id="SimpleApp"
  logo="SimpleApp.logo"
  title="SimpleApp.title"
  subtitle="SimpleApp.subtitle"
  user-message="SimpleApp.UserMessage">

  <ac:application-menu>
    <ac:preferences title="preferences.title"/>
    <ac:help title="help.title"/>
    <ac:logout title="logout.title"/>
  </ac:application-menu>

  <ac:section-ref id="SimpleAppHomeSection"/>
        <ac:section-ref id="SimpleAppWorkspaceSection"/>
        </ac:application>
```

Figure 16. SimpleApp app

This section is defined in the `SimpleAppWorkspaceSection.sec` file, detailed in “Add a new Section” on page 13. The difference between this section and the previously defined section is the addition of a shortcut panel, refer to User Interface Element 1 in “Introduction” on page 13. This is accomplished through the addition of the `shortcut-panel-ref` element.

`SimpleWorkspaceSection.sec` defines a structure with two tabs, the contents of which will be defined in the next chapter.

```
<sc:section
  id="SimpleAppWorkspaceSection"
  title="Section.Home.Title">

  <sc:shortcut-panel-ref id="SimpleShortcutPanel"/>

  <sc:tab id="SimpleSearch"/>
  <sc:tab id="SimplePerson"/>

</sc:section>
```

Figure 17. SimpleAppWorkspaceSection.sec

The corresponding `.properties` should contain the localizable content for the section:

```
Section.Home.Title=Workspace
```

Defining contents of a Section Shortcut Panel

The purpose behind a Section Shortcut Panel is to provide the user with quick links to open content and perform actions within the section. The panel can be collapsed and expanded as desired by the user.

The contents of a shortcut panel are configured in an XML configuration file that has an extension of `.ssp` and a corresponding properties file. “Defining contents of a Section Shortcut Panel” shows the contents of `SimpleShortcutPanel.ssp`.

```
<sc:section-shortcut-panel
  id="SimpleShortcutPanel"
  title="Panel.Title">
    <sc:nodes>
      <sc:node type="group" title="Group.Title" id="UI">
        <sc:node type="leaf" id="search" page-id="SimpleSearch"
               title="Link.Title.Search"/>
      </sc:node>
  </sc:nodes>

</sc:section-shortcut-panel>
```

Figure 18. SimpleShortcutPanel.ssp

The corresponding .properties should contain the localizable content for the shortcut panel:

```
Panel.Title=Shortcuts
Group.Title=Quick Links
Link.Title.Search=Person Search
```

The structure of the section shortcut panel consists of nodes of two different types, namely group and leaf nodes. The type is configured through the type attribute. Group nodes allow for logical grouping of leaf nodes. Each leaf node represents a link to appear on the Section Shortcut Panel.

Both group and leaf nodes have a title attribute which allows the configuration of the text to be displayed. Leaf nodes must additionally specify a page-id attribute which configures the target page of the link.

The section shortcut panel defined in “Defining contents of a Section Shortcut Panel” on page 14 is showed in the User Interface Element 1 of “Introduction” on page 13. It has a group and leaf node, where the group node contains the leaf node that holds a hyperlink, refer to User Interface Element 2 and 3 respectively in “Introduction” on page 13. Clicking this link will result in opening the search tab, which will be defined next.

Define a Search Tab

“Define a Search Tab” shows the configuration of the first of the two tabs (named as Search Tab, refer to User Interface Element 5 in “Introduction” on page 13) referenced in the new section. This tab contains a single page that allows the user to search for a person.

```
<tc:tab-config
        id="SimpleSearch">

  <tc:context tab-name="search.tab.name"
             tab-title="search.tab.title"/>
  <tc:navigation page-id="SimpleSearch"/>

</tc:tab-config>
```

Figure 19. SimpleSearch.tab

The corresponding .properties should contain the localizable content for this tab:

```
search.tab.name=Search
search.tab.title=Person Search
```

Define the Search Page

"Define the Search Page" shows a screen shot of the page to search a person (named as `Person Search`) below. The `Person Search` page has two distinct areas, a cluster that allows the user to enter search criteria and a list to display the results of a search.

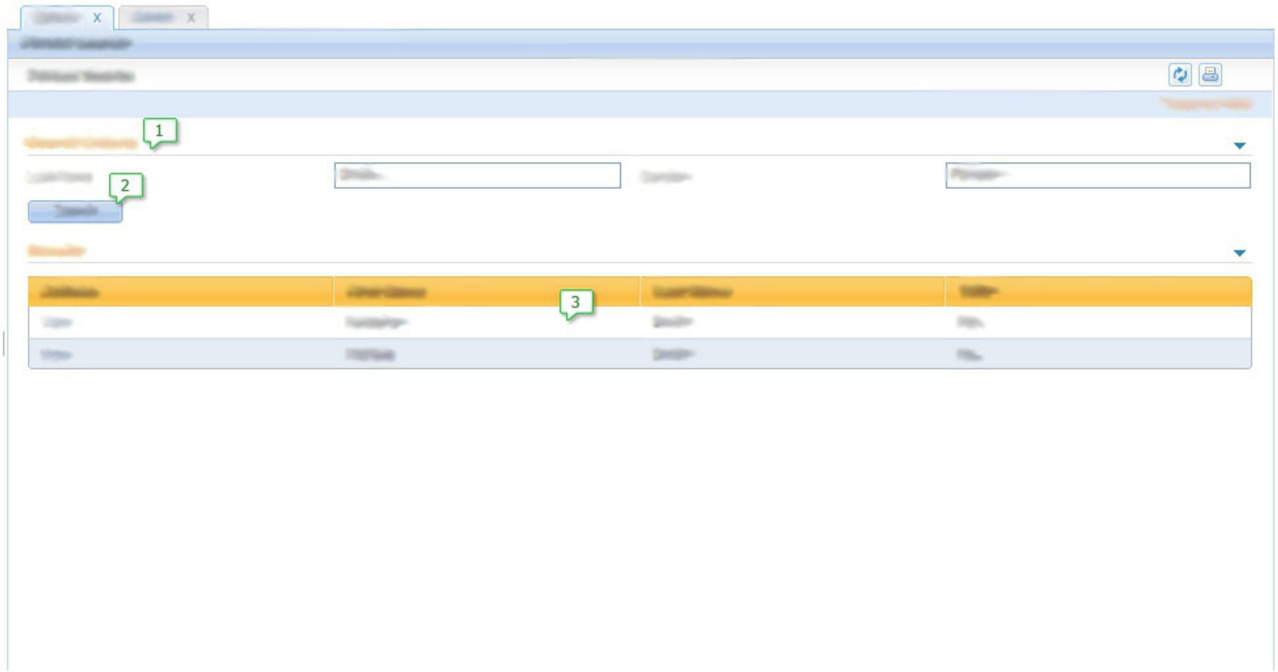

Figure 20. Person Search Page

1. Cluster
2. Action Control
3. List

The following is the UIM code for the page:

```
<PAGE PAGE_ID="SimpleSearch">

  <SERVER_INTERFACE NAME="ACTION" CLASS="PersonFacade"
    OPERATION="advancedSearch" PHASE="ACTION"/>

  <PAGE_TITLE>
    <CONNECT>
      <SOURCE NAME="TEXT"
        PROPERTY="PageTitle.StaticText"/>
    </CONNECT>
  </PAGE_TITLE>

  <CLUSTER TITLE="Cluster.Title.Search" NUM_COLS="2">
    <FIELD LABEL="Field.Label.LastName">
      <CONNECT>
        <TARGET NAME="ACTION" PROPERTY="key$dtls$lastName"/>
      </CONNECT>
    </FIELD>
    <FIELD LABEL="Field.Label.Gender">
      <CONNECT>
        <TARGET NAME="ACTION" PROPERTY="key$dtls$gender"/>
      </CONNECT>
    </FIELD>
    <ACTION_SET TOP="false">
      <ACTION_CONTROL LABEL="Control.Label.Search"
        TYPE="SUBMIT">
        <LINK PAGE_ID="THIS"/>
      </ACTION_CONTROL>
    </ACTION_SET>
  </CLUSTER>

  <LIST TITLE="List.Title.Results">
    <CONTAINER LABEL="Container.Label.Actions">
       <ACTION_CONTROL LABEL="Control.Label.View">
        <LINK PAGE_ID="SimplePerson">
          <CONNECT>
            <SOURCE NAME="ACTION" PROPERTY="personID"/>
            <TARGET NAME="PAGE" PROPERTY="personID"/>
          </CONNECT>
        </LINK>
      </ACTION_CONTROL>
    </CONTAINER>
    <FIELD LABEL="Field.Label.FirstName">
      <CONNECT>
        <SOURCE NAME="ACTION" PROPERTY="firstName"/>
      </CONNECT>
    </FIELD>
    <FIELD LABEL="Field.Label.LastName">
      <CONNECT>
        <SOURCE NAME="ACTION"
          PROPERTY="result$dtls$dtls$lastName"/>
      </CONNECT>
    </FIELD>
    <FIELD LABEL="Field.Label.Title">
      <CONNECT>
        <SOURCE NAME="ACTION" PROPERTY="title"/>
      </CONNECT>
    </FIELD>
   </LIST>
</PAGE>
```

Figure 21. SimpleSearch.uim

The following are the main elements of note on this UIM page:

- The SERVER_INTERFACE element defines which server interface method is called by the server when the form is submitted.

- The CLUSTER defines the cluster on the page that contains two fields that allow the user to enter the search criteria. These are mapped to the input parameters of the server interface method. Refer to User Interface Element 1 in “Define the Search Page” on page 16.
- An ACTION_CONTROL element defines the action control on the page that allows the search to be submitted. Refer to User Interface Element 2 in “Define the Search Page” on page 16.
- The LIST defines the list on the page that contains the results of a submitted search. For each result a row is displayed which displays the person's details, and an ACTION_CONTROL which defines a link to that person's home page. Refer to User Interface Element 3 in “Define the Search Page” on page 16. Selecting this link will open the person tab which will be defined next.

The corresponding .properties should contain the localizable content for the search page:

```
PageTitle.StaticText=Person Search

Field.Label.FirstName=First Name
Field.Label.LastName=Last Name
Field.Label.Title=Title
Field.Label.Gender=Gender
Control.Label.View=View

Container.Label.Actions=Actions

Cluster.Title.Search=Search Criteria
List.Title.Results=Results
Control.Label.Search=Search
```

Summary

The following is a summary of the files that were modified during this chapter, and a reference to the targets that process each of them.

Table 2. File Summary

File	Location	Target
SimpleApp.app and associated properties file	EJBServer\components\<component-name>\clientapps	inserttabconfiguration
SimpleWorkspaceSection.sec and associated properties file	EJBServer\components\<component-name>\clientapps	inserttabconfiguration
SimpleSearch.tab and associated properties file	EJBServer\components\<component-name>\clientapps	inserttabconfiguration
SimpleSearch.uim and associated properties file	webclient\components\<component-name>\	client

Adding Tab Content

Introduction

This chapter will detail how to add more complex structured tabs, including a context panel and a content area. The Simple Application will be extended to configure the contents of the new added section in “Add a new Section” on page 13.

"Introduction" on page 18 shows a screen shot of the new added section containing the tab (named as Person Tab) with the tab content of a context panel and a content area (named as Person Page).

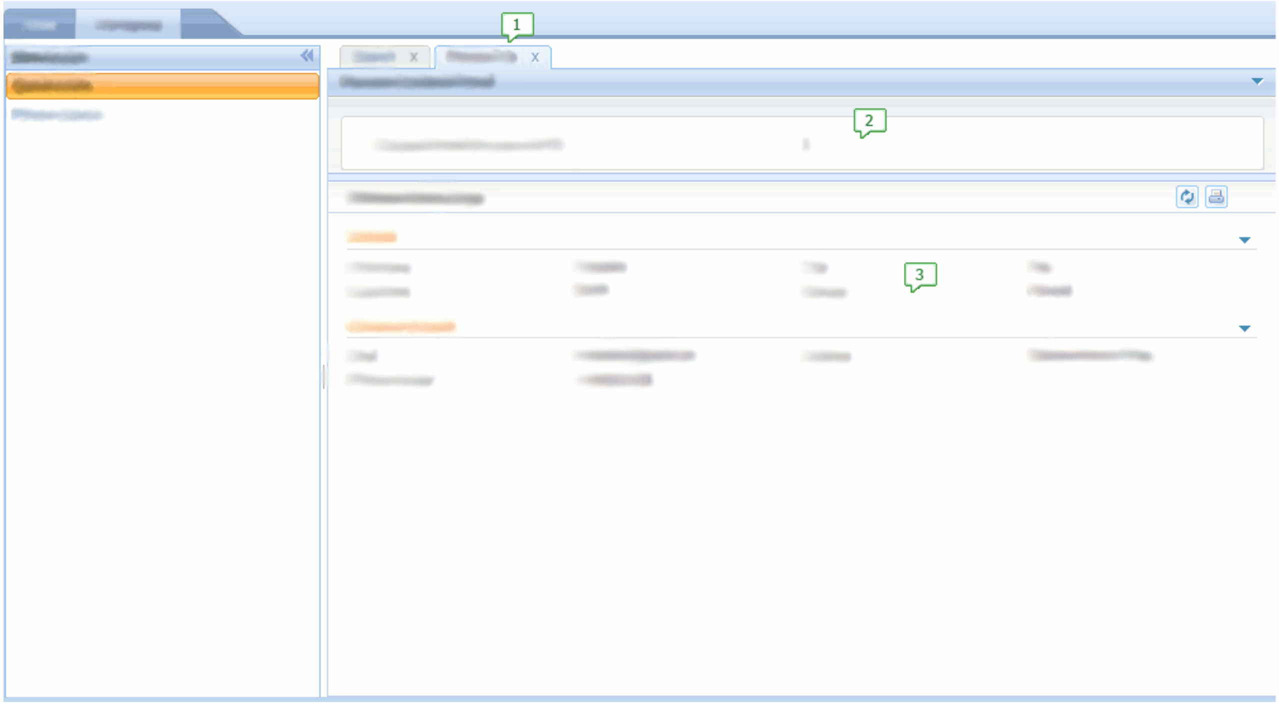

Figure 22. Workspace Section including tab with Context Panel

1. Person Tab
2. Context Panel
3. Content Area (Person Page)

Define the Person Tab

"Define the Person Tab" shows the configuration of the Person Tab referenced in the section, refer to User Interface Element 1 in "Introduction" on page 18. This Person Tab contains a single page that allows the user to display the details of a person. And the context panel is configured with the tab, refer to User Interface Element 2 in "Introduction" on page 18.

It also requires a parameter to be passed to the tab when opened, as defined by the `page-param` element.

```
<tc:tab-config>
  <tc:page-param name="personID"/>
  <tc:context page-id="SimplePersonContext"/>
  <tc:navigation page-id="SimplePerson"/>

</tc:tab-config>
```

Figure 23. SimplePerson tab

The corresponding `.properties` contains the localizable content for this Person Tab:

```
no.property.required=true
```

Define a Context Panel

When a context panel is present, it is displayed at the top of the tab's content area, and regardless of the particular page displayed below it, it is always there to provide the user with important contextual information. Refer to User Interface Element 2 in "Introduction" on page 18.

A context panel is defined using a UIM page with some limitation on the UIM allowed. The following is the UIM code for the context panel defined in the Person Tab:

```
<PAGE PAGE_ID="SimplePersonContext" TYPE="DETAILS">

  <PAGE_TITLE>
    <CONNECT>
      <SOURCE NAME="TEXT"
        PROPERTY="PageTitle.StaticText"/>
    </CONNECT>
  </PAGE_TITLE>

  <TAB_NAME>
    <CONNECT>
      <SOURCE NAME="TEXT" PROPERTY="Tab.title"/>
    </CONNECT>
  </TAB_NAME>


  <PAGE_PARAMETER NAME="personID"/>

  <CLUSTER>
    <FIELD LABEL="Field.Label.ContextPanelFor">
      <CONNECT>
        <SOURCE NAME="PAGE" PROPERTY="personID"/>
      </CONNECT>
    </FIELD>
  </CLUSTER>

</PAGE>
```

Figure 24. SimplePersonContext.uim

The TYPE attribute allows the developer to specify that a UIM page is intended as a context panel.

The TAB_NAME element defines the content that will be used as the name of the tab.

The PAGE_TITLE element is used as the tab title.

In this example, the context panel only contains one single field that outputs the unique identifier of the person. Note that the page-param matches that defined earlier in the tab configuration.

The corresponding .properties contains the localizable content for the context panel:

```
PageTitle.StaticText=Person Context Panel
Tab.title=Person Tab

Field.Label.ContextPanelFor=Context Panel for user with ID:
```

Define the Person Page

“Define the Person Page” defines the UIM for the page (named as Person Page) to be displayed in the Person Tab. Refer to User Interface Element 3 in “Introduction” on page 18.

```
<PAGE PAGE_ID="SimplePerson">
  <PAGE_TITLE>
    <CONNECT>
       <SOURCE NAME="TEXT" PROPERTY="Page.Title"/>
    </CONNECT>
    </PAGE_TITLE>

    <PAGE_PARAMETER NAME="personID"/>

    <SERVER_INTERFACE NAME="DISPLAY"
      CLASS="PersonFacade"
      OPERATION="readPerson" />

   <CONNECT>
     <SOURCE NAME="PAGE" PROPERTY="personID"/>
     <TARGET NAME="DISPLAY" PROPERTY="key$personID"/>
   </CONNECT>


   <CLUSTER TITLE="Cluster.Title.Details" NUM_COLS="2">

    <FIELD LABEL="Field.Label.FirstName">
      <CONNECT>
        <SOURCE NAME="DISPLAY" PROPERTY="firstName"/>
      </CONNECT>
    </FIELD>
    <FIELD LABEL="Field.Label.LastName">
      <CONNECT>
        <SOURCE NAME="DISPLAY" PROPERTY="lastName"/>
      </CONNECT>
    </FIELD>
    <FIELD LABEL="Field.Label.Title">
      <CONNECT>
        <SOURCE NAME="DISPLAY" PROPERTY="title"/>
      </CONNECT>
    </FIELD>
    <FIELD LABEL="Field.Label.Gender">
      <CONNECT>
        <SOURCE NAME="DISPLAY" PROPERTY="gender"/>
      </CONNECT>
    </FIELD>
  </CLUSTER>
  <CLUSTER TITLE="Cluster.Title.ContactDetails" NUM_COLS="2">
    <FIELD LABEL="Field.Label.Email">
      <CONNECT>
        <SOURCE NAME="DISPLAY" PROPERTY="email"/>
      </CONNECT>
    </FIELD>
    <FIELD LABEL="Field.Label.PhoneNumber">
      <CONNECT>
        <SOURCE NAME="DISPLAY" PROPERTY="phoneNumber"/>
      </CONNECT>
    </FIELD>
    <FIELD LABEL="Field.Label.Address">
      <CONNECT>
        <SOURCE NAME="DISPLAY" PROPERTY="address"/>
      </CONNECT>
    </FIELD>
  </CLUSTER>
</PAGE>
```

Figure 25. SimplePerson uim

This UIM is similar to what has been previously defined.

The corresponding .properties should contain the localizable content for the page:

```
Page.Title=Person Home Page

Cluster.Title.Details=Details
Cluster.Title.ContactDetails=Contact Details

Field.Value.Welcome=Field Value
Field.Label.Welcome=Field Label
Field.Label.FirstName=First Name
Field.Label.LastName=Last Name
Field.Label.Title=Title
Field.Label.Gender=Gender
Field.Label.Email=Email
Field.Label.PhoneNumber=Phone Number
Field.Label.Address=Address
```

Summary

This chapter introduced the concepts of adding tab content. The following is a summary of the files that were modified during this chapter, and a reference to the targets that will process each of them.

Table 3. File Summary

File	Location	Target
SimpleSearch.tab and associated properties file	`EJBServer\components\<component-name>\clientapps`	inserttabconfiguration
SimpleSearch.uim and associated properties file	`webclient\components\<component-name>\`	client
SimplePerson.tab and associated properties file	`EJBServer\components\<component-name>\clientapps`	inserttabconfiguration
SimpleContextPerson.uim and associated properties file	`webclient\components\<component-name>\`	client
SimplePerson.uim and associated properties file	`webclient\components\<component-name>\`	client

Working with Modal Dialogs

Introduction

This chapter will introduce the ability to modify content in the Cúram application. This includes the following features:

- Modal Dialogs
- Wizard Progress Bar

These additional features will be illustrated by extending the existing application to include the capability of adding an employment history for a person.

“Introduction” shows a screen shot of a simple modal dialog to be added.

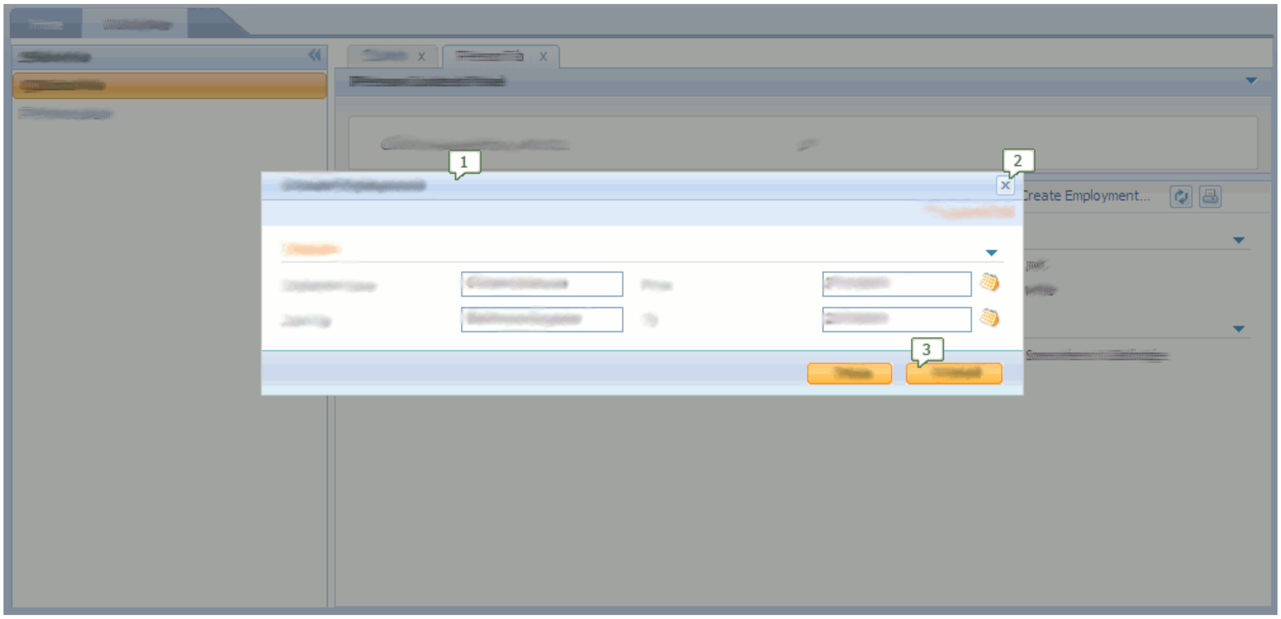

Figure 26. Modal Dialog

1. Title Bar
2. Close Button
3. Action Controls

A modal dialog is a separate window that opens over the rest of the interface. The user cannot switch back to the parent interface until the modal is closed, either by submitting it or canceling it.

Modal dialogs are recommend for editing data in the Cúram application because they make the modification of data more transactional; the user is forced to either submit changes or cancel them, and there is no ambiguity caused by the user being allowed to switch context in the middle of making changes.

Where there is too much information to for a single modal dialog, a wizard progress bar (will be covered later) can be used. This allows a sequence of modal dialogs to be used for editing.

Opening a modal dialog

Page Level Action Controls allow the user to perform certain actions on the page. For this example, the Person Page defined in "Define the Person Page" on page 21 will be extended to contain two action controls, one of which will open a basic modal dialog and another that will open a wizard progress bar.

"Opening a modal dialog" shows a screen shot of the extended page with those Page Level Action Controls.

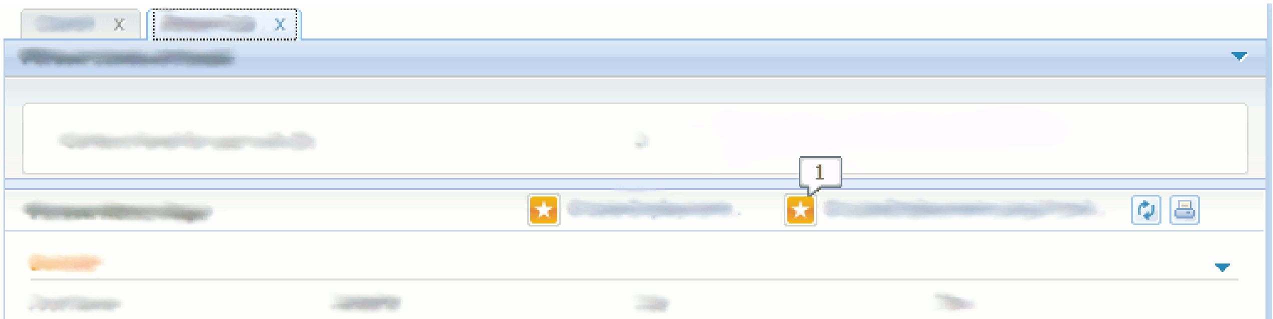

Figure 27. Extended Simple Person Page With Page Level Action Controls

1.Page Level Action Controls

"Opening a modal dialog" on page 24 shows the code of the extended `SimplePerson.uim`.

```
       <PAGE PAGE_ID="SimplePerson">

 ...
        <ACTION_SET>
    <ACTION_CONTROL LABEL="Control.Label.CreateEmployment">
      <LINK PAGE_ID="CreateEmployments" OPEN_MODAL="true">
       <CONNECT>
         <SOURCE NAME="PAGE" PROPERTY="personID"/>
         <TARGET NAME="PAGE" PROPERTY="personID"/>
       </CONNECT>
      </LINK>
   </ACTION_CONTROL>
   <ACTION_CONTROL LABEL="Control.Label.CreateEmploymentWizard">
     <LINK PAGE_ID="CreateEmploymentWizard_pageOne"
       OPEN_MODAL="true">
       <CONNECT>
         <SOURCE NAME="PAGE" PROPERTY="personID"/>
         <TARGET NAME="PAGE" PROPERTY="personID"/>
       </CONNECT>
     </LINK>
   </ACTION_CONTROL>
 </ACTION_SET>
       ...

</PAGE>
```

Figure 28. SimplePerson uim

The corresponding `.properties` is extended to include the label properties for the action controls:

```
Control.Label.CreateEmployment=Add Employment History
Control.Label.CreateEmployment=Add Employment in Wizard
```

Defining the content of the modal dialog

The content of a modal dialog is just a standard UIM page, though it is styled different when it is displayed by the browser. Specifically, in the modal dialog that we are going to define,

- the title is displayed in the title bar of the window, refer to User Interface Element 1 in "Introduction" on page 23;
- the action controls appear in a bar at the bottom of the page, refer to User Interface Element 3 in "Introduction" on page 23;

- and the close button on the title bar allows the user to close the window without submitting changes, refer to User Interface Element 2 in "Introduction" on page 23.

"Defining the content of the modal dialog" on page 25 shows the UIM code for this modal dialog.

```
<PAGE PAGE_ID="CreateEmployments" WINDOW_OPTIONS="width=250">

  <PAGE_TITLE>
    <CONNECT>
      <SOURCE NAME="TEXT" PROPERTY="Page.Title"/>
    </CONNECT>
  </PAGE_TITLE>

  <PAGE_PARAMETER NAME="personID"/>

  <CONNECT>
    <SOURCE NAME="PAGE" PROPERTY="personID"/>
    <TARGET NAME="ACTION" PROPERTY="personID"/>
  </CONNECT>

  <SERVER_INTERFACE NAME="ACTION"
    CLASS="EmploymentFacade"
    OPERATION="createEmployment"
    PHASE="ACTION"/>

  <CLUSTER TITLE="Cluster.Title.Details" NUM_COLS="2">
    <FIELD LABEL="Field.Label.EmployerName">
      <CONNECT>
        <TARGET NAME="ACTION" PROPERTY="employerName"/>
      </CONNECT>
    </FIELD>
    <FIELD LABEL="Field.Label.JobTitle">
      <CONNECT>
        <TARGET NAME="ACTION" PROPERTY="jobTitle"/>
      </CONNECT>
    </FIELD>
    <FIELD LABEL="Field.Label.FromDate">
      <CONNECT>
        <TARGET NAME="ACTION" PROPERTY="fromDate"/>
      </CONNECT>
    </FIELD>
    <FIELD LABEL="Field.Label.ToDate">
      <CONNECT>
        <TARGET NAME="ACTION" PROPERTY="toDate"/>
      </CONNECT>
    </FIELD>
  </CLUSTER>

  <ACTION_SET TOP="false">
    <ACTION_CONTROL LABEL="Control.Label.Save" TYPE="SUBMIT">
    <ACTION_CONTROL LABEL="Control.Label.Cancel" TYPE="SUBMIT"/>
  </ACTION_SET>

</PAGE>
```

Figure 29. CreateEmployments.uim

One element to note is the WINDOW_OPTIONS attribute of the PAGE element. In this example, the width has been set to 250. As no height is set, it will be automatically calculated when the dialog is displayed.

The corresponding .properties should contain the localizable content for the modal dialog:

```
Page.Title=Create Employment
Cluster.Title.Details=Details
Field.Value.Welcome=Here's the details panel for a person

Control.Label.Save=Save
Control.Label.Cancel=Cancel

Field.Label.PersonID=Person ID
Field.Label.EmployerName=Employer Name
Field.Label.JobTitle=Job Title
Field.Label.FromDate=From
Field.Label.ToDate=To
```

Using a Wizard Progress Bar

In certain scenarios, it is desirable to split modifications over a number of pages. In the Cúram application, this is achieved through the use of a Wizard Progress Bar.

"Using a Wizard Progress Bar" shows the different artifacts that the inclusion of a wizard adds to a modal dialog's display.

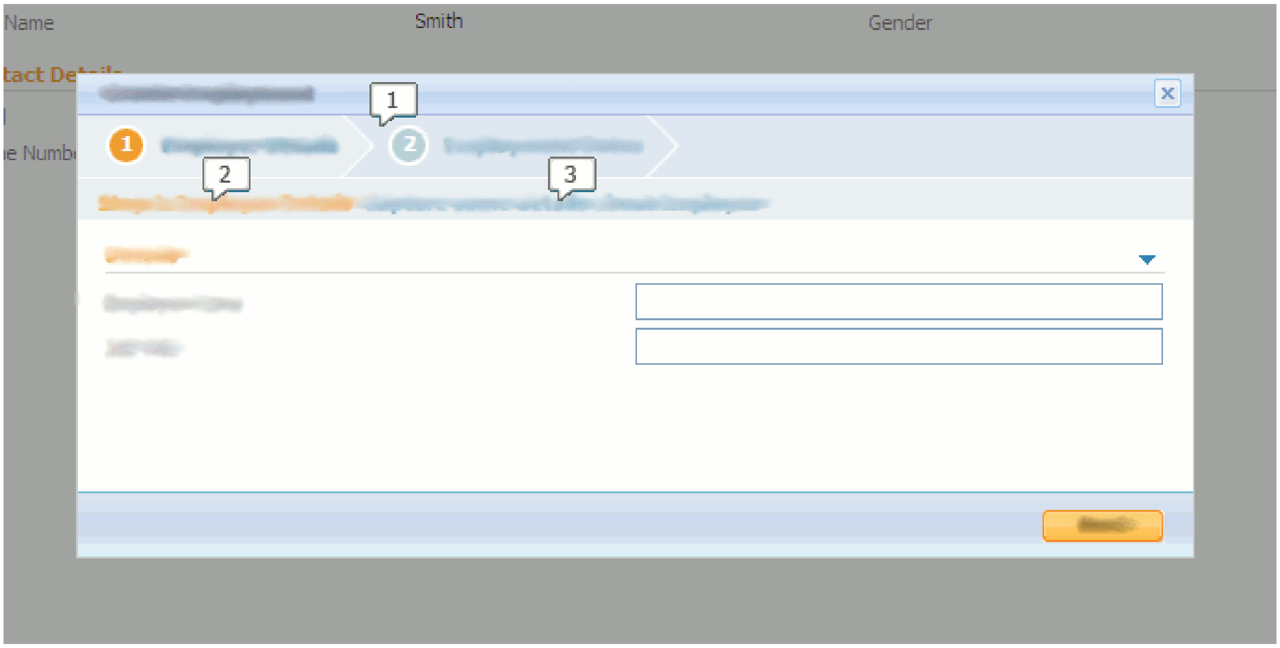

Figure 30. Create Employment Wizard Progress Bar

1. Wizard Progress Bar
2. Step Title
3. Step Description

This artifact is added to the display of a modal dialog to indicate that the particular page is part of a sequence of pages. It includes

- a wizard progress bar which shows the sequence of pages with the current page highlighted, refer to User Interface Element 1 in "Using a Wizard Progress Bar";
- a step title, refer to User Interface Element 2 in "Using a Wizard Progress Bar";

- and a step description, refer to User Interface Element 3 in “Using a Wizard Progress Bar” on page 27.

To illustrate the use of a wizard, an alternative way of adding an employment history will be added to the application. Instead of having one dialog, the information will be split over a sequence of two pages, as can be seen in “Using a Wizard Progress Bar” on page 27 and “Using a Wizard Progress Bar” on page 27

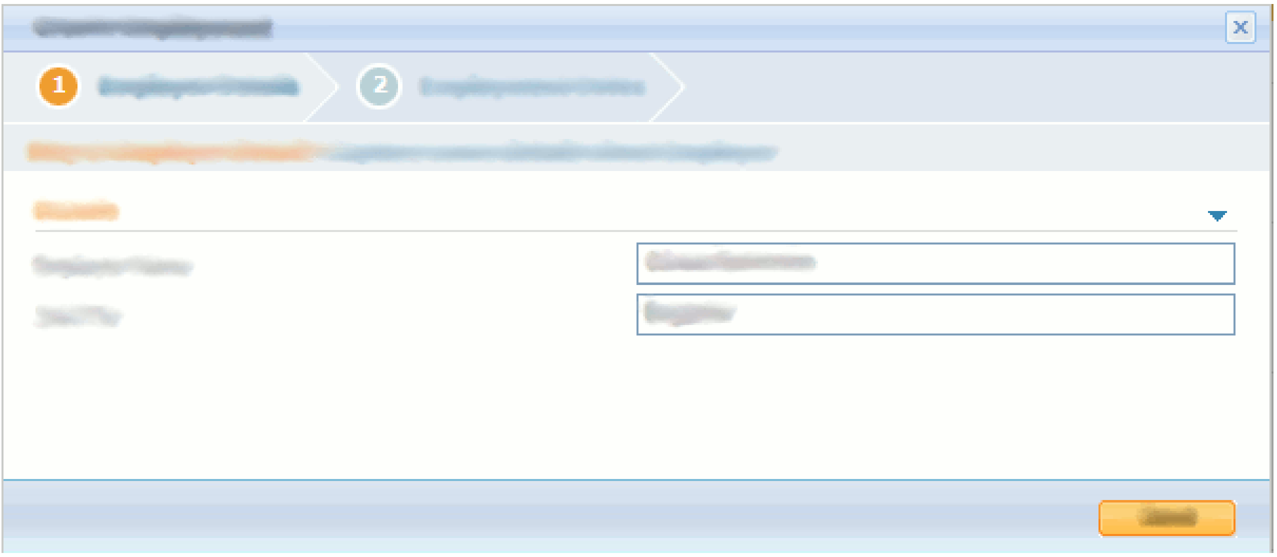

Figure 31. Create Employment Wizard - Page One

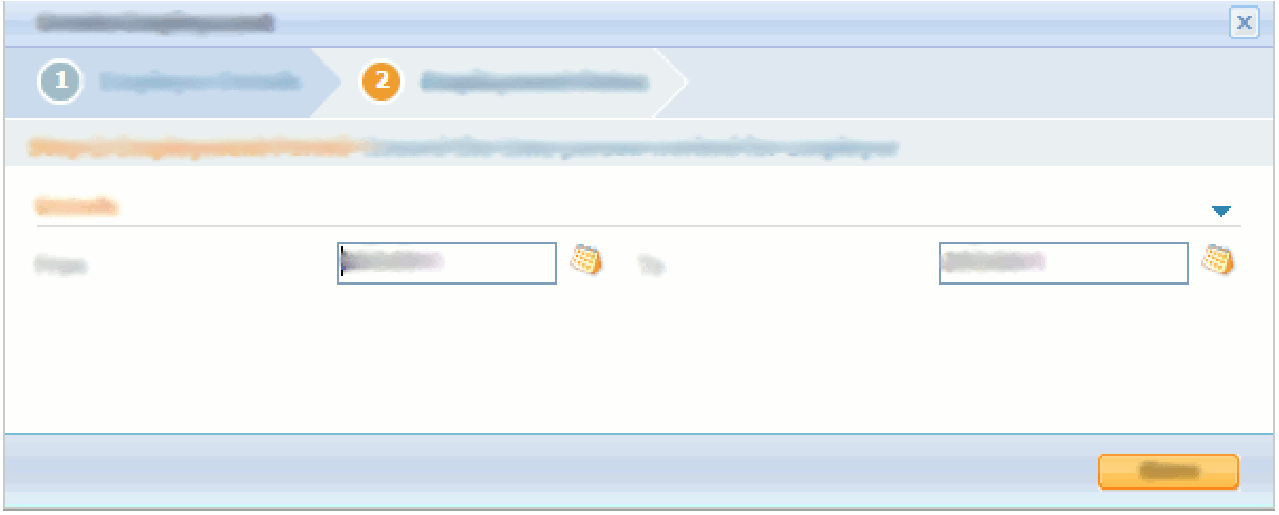

Figure 32. Create Employment Wizard - Page Two

Wizard Progress Bar Configuration File

`CreateEmploymentWizard.properties` contains the configuration of the wizard. “Wizard Progress Bar Configuration File” shows the configuration file for the wizard progress bar. The wizard has two pages and the configuration specifies the text to appear in the progress bar, the step title, and the step description for each page.

```
Number.Wizard.Pages=2

CreateEmploymentWizard_pageOne.Wizard.Item.Text=Employer Details
CreateEmploymentWizard_pageOne.Wizard.Page.Title=
  Step 1: Employer Details
CreateEmploymentWizard_pageOne.Wizard.Page.Desc=
  Capture some details about Employer
Wizard.PageID.1=CreateEmploymentWizard_pageOne

CreateEmploymentWizard_pageTwo.Wizard.Item.Text=Employment Dates
CreateEmploymentWizard_pageTwo.Wizard.Page.Title=Step 2:
  Employment Period
CreateEmploymentWizard_pageTwo.Wizard.Page.Desc=
  Record the time person worked for employer
Wizard.PageID.2=CreateEmploymentWizard_pageTwo
```

Figure 33. CreateEmploymentWizard.properties

The wizard configuration file is loaded into the data by adding the following lines to `AppResource.dmx`:

```
<row>
    <attribute name="resourceid">
      <value>1</value>
    </attribute>
    <attribute name="localeIdentifier">
      <value/>
    </attribute>
    <attribute name="name">
      <value>CreateEmploymentWizard</value>
    </attribute>
    <attribute name="contentType">
      <value>text/plain</value>
    </attribute>
    <attribute name="contentDisposition">
      <value>inline</value>
    </attribute>
    <attribute name="content">
      <value>./blob/CreateEmploymentWizard.properties</value>
    </attribute>
    <attribute name="internal">
      <value>1</value>
    </attribute>
    <attribute name="lastWritten">
      <value>2008-06-13-19.29.40</value>
    </attribute>
    <attribute name="versionNo">
      <value>1</value>
    </attribute>
    <attribute name="category">
      <value>RS_PROP</value>
    </attribute>
  </row>
```

Definition of Wizard Pages.

The sequence will contain two pages; the first requires the user to enter employer details and the second requires the user to enter dates.

```
<PAGE PAGE_ID="CreateEmploymentWizard_pageOne">

  <MENU MODE="WIZARD_PROGRESS_BAR">
      <CONNECT>
        <SOURCE NAME="CONSTANT" PROPERTY="Wizard" />
      </CONNECT>
  </MENU>

  <SERVER_INTERFACE NAME="ACTION" CLASS="EmploymentFacade"
    OPERATION="validateEmployerAndJobTitle" PHASE="ACTION"/>

  <PAGE_TITLE>
    <CONNECT>
      <SOURCE NAME="TEXT" PROPERTY="Page.Title"/>
    </CONNECT>
  </PAGE_TITLE>

  <CONNECT>
    <SOURCE NAME="PAGE" PROPERTY="personID"/>
    <TARGET NAME="ACTION" PROPERTY="personID"/>
  </CONNECT>

  <PAGE_PARAMETER NAME="personID"/>

  <CLUSTER TITLE="Cluster.Title.Details">
    <FIELD LABEL="Field.Label.EmployerName">
      <CONNECT>
        <TARGET NAME="ACTION" PROPERTY="employerName"/>
      </CONNECT>
    </FIELD>
    <FIELD LABEL="Field.Label.JobTitle">
      <CONNECT>
        <TARGET NAME="ACTION" PROPERTY="jobTitle"/>
      </CONNECT>
    </FIELD>

  </CLUSTER>

  <ACTION_SET TOP="false">
    <ACTION_CONTROL LABEL="Control.Label.Next" TYPE="SUBMIT">
      <LINK PAGE_ID="CreateEmploymentWizard_pageTwo"
        DISMISS_MODAL="false">
        <CONNECT>
           <SOURCE NAME="ACTION" PROPERTY="personID"/>
           <TARGET NAME="PAGE" PROPERTY="personID"/>
        </CONNECT>
        <CONNECT>
           <SOURCE NAME="ACTION" PROPERTY="employerName"/>
           <TARGET NAME="PAGE" PROPERTY="employerName"/>
        </CONNECT>
        <CONNECT>
          <SOURCE NAME="ACTION" PROPERTY="jobTitle"/>
          <TARGET NAME="PAGE" PROPERTY="jobTitle"/>
        </CONNECT>
      </LINK>
      </ACTION_CONTROL>
  </ACTION_SET>
</PAGE>
```

Figure 34. CreateEmploymentWizard_pageOne.uim

The wizard progress bar artifacts are added to the page through the inclusion of a MENU element with the attribute MODE="WIZARD_PROGRESS_BAR". The element references a constant called Wizard. The value of this property is defined in

Constants.properties as CreateEmploymentWizard. This is how each page is associated with the wizard progress bar configuration already loaded into the database for this example.

The corresponding .properties for the first page of the wizard includes the localizable content for that page:

```
Page.Title=Create Employment
Cluster.Title.Details=Details

Control.Label.Next=Next

Field.Label.EmployerName=Employer Name
Field.Label.JobTitle=Job Title
```

The following is the second page that displayes the wizard progress bar:

```
<PAGE PAGE_ID="CreateEmploymentWizard_pageTwo">

  <MENU MODE="WIZARD_PROGRESS_BAR">
        <CONNECT>
          <SOURCE NAME="CONSTANT" PROPERTY="Wizard" />
        </CONNECT>
  </MENU>

  <PAGE_TITLE>
    <CONNECT>
      <SOURCE NAME="TEXT" PROPERTY="Page.Title"/>
    </CONNECT>
  </PAGE_TITLE>

  <PAGE_PARAMETER NAME="personID"/>

  <CONNECT>
    <SOURCE NAME="PAGE" PROPERTY="personID"/>
    <TARGET NAME="ACTION" PROPERTY="personID"/>
  </CONNECT>

  <PAGE_PARAMETER NAME="employerName"/>

  <CONNECT>
    <SOURCE NAME="PAGE" PROPERTY="employerName"/>
    <TARGET NAME="ACTION" PROPERTY="employerName"/>
  </CONNECT>


  <PAGE_PARAMETER NAME="jobTitle"/>

  <CONNECT>
      <SOURCE NAME="PAGE" PROPERTY="jobTitle"/>
      <TARGET NAME="ACTION" PROPERTY="jobTitle"/>
  </CONNECT>

  <SERVER_INTERFACE NAME="ACTION" CLASS="EmploymentFacade"
    OPERATION="createEmployment" PHASE="ACTION"/>

  <CLUSTER TITLE="Cluster.Title.Details" NUM_COLS="2">

    <FIELD LABEL="Field.Label.FromDate">
      <CONNECT>
        <TARGET NAME="ACTION" PROPERTY="fromDate"/>
      </CONNECT>
    </FIELD>
    <FIELD LABEL="Field.Label.ToDate">
      <CONNECT>
        <TARGET NAME="ACTION" PROPERTY="toDate"/>
      </CONNECT>
    </FIELD>
  </CLUSTER>

  <ACTION_SET TOP="false">
    <ACTION_CONTROL LABEL="Control.Label.Save" TYPE="SUBMIT">
      <LINK PAGE_ID="Employments" DISMISS_MODAL="TRUE">
        <CONNECT>
    <SOURCE NAME="ACTION" PROPERTY="personID" />
    <TARGET NAME="PAGE" PROPERTY="personID"/>
        </CONNECT>
      </LINK>
    </ACTION_CONTROL>
  </ACTION_SET>

</PAGE>
```

Figure 35. CreateEmploymentWizard_pageTwo.uim

The corresponding `.properties` is includes the localizable content for the page:

```
Page.Title=Create Employment
Cluster.Title.Details=Details
Control.Label.Save=Save

Field.Label.FromDate=From
Field.Label.ToDate=To
```

Summary

This chapter introduced using modals and wizard progress bars. The following is a summary of the files that were modified during this chapter, and a reference to the targets that will process each of them.

Table 4. File Summary

File	Location	Target
SimplePerson.uim and associated properties file	`webclient\components\<component-name>\`	client
CreateEmployments.uim and associated properties file	`webclient\components\<component-name>\`	client
CreateEmployments Wizard_pageOne.uim and associated properties file	`webclient\components\<component-name>\`	client
CreateEmployments Wizard_pageTwo.uim and associated properties file	`webclient\components\<component-name>\`	client
CreateEmploymentsWizard .properties	`EJBServer\components\<component-name>\Data_Manager\scripts\blob`	client
APPRESOURCES.DMX	`EJBServer\components\<component-name>\Data_Manager\scripts`	client

Adding Tab Navigation

Introduction

This chapter will detail how to add navigation features to a tab. The Person Tab will be modified to include a navigation bar. “Introduction” shows a screenshot of the modified Person Tab with a navigation bar.

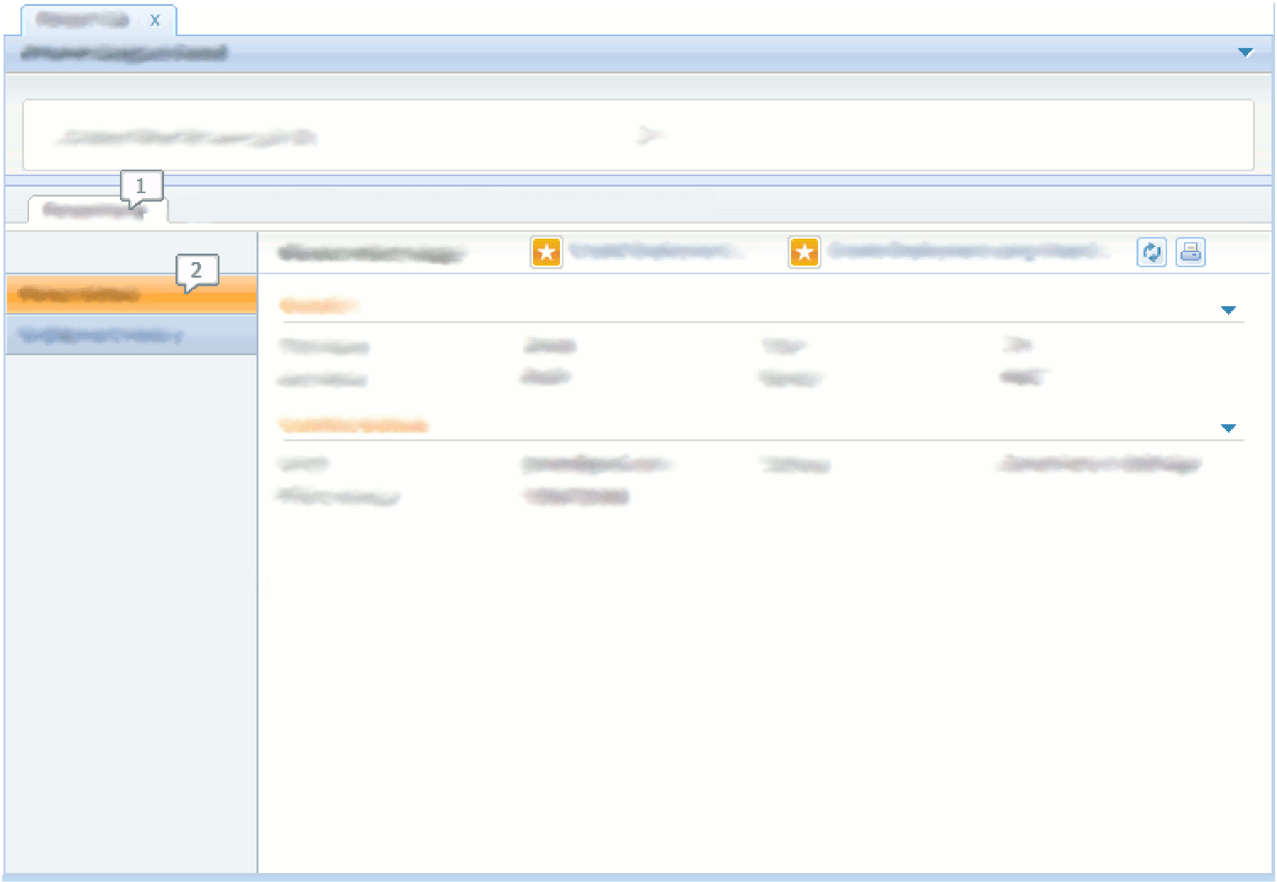

Figure 36. Tab with Navigation Bar

1. Content Area Navigation Bar
2. Page Group Navigation Bar

It contains a content area navigation bar within one navigation tab, and a page group navigation bar with two navigation pages.

Define a Navigation Bar

To configure a tab to contain a navigation bar in the content area, it is necessary to include the `id` of the navigation bar configuration in the navigation element of the tab. “Define a Navigation Bar” shows the modified version of `Person.tab`.

```
      <tc:tab-config
     id="SimplePerson">

<tc:page-param name="personID"/>

<tc:context page-id="SimplePersonContext" height="60"/>
      <tc:navigation id="SimplePersonNav"/>
      </tc:tab-config>
```

Figure 37. SimplePerson Tab

The navigation bar configuration itself is defined using an XML configuration file with the extension `.nav`. Like other tab configuration artifacts, the `.nav` files are located in the `EJBServer\components\<component-name>\clientapps` directory, where <component-name> is a custom component.

"Define a Navigation Bar" on page 34 shows the contents of `SimplePersonNav.nav`. It defines one navigation group, with two navigation pages.

```
<nc:navigation id="SimplePersonNav">
  <nc:nodes>
       <nc:navigation-group id="PersonHome" title="PersonHome"
         description="Person Details Group">
     <nc:navigation-page id="SimplePerson" page-id="SimplePerson"
       title="PersonDetails.Title"/>
     <nc:navigation-page id="Employments" page-id="Employments"
       title="EmploymentHistory.Title"/>
   </nc:navigation-group>
</nc:nodes>
</nc:navigation>
```

Figure 38. SimplePersonNav.nav

The corresponding `.properties` should contain the localizable content for the page:

```
PersonHome.Title=Person Home
EmploymentHistory.Title=Employment History
PersonDetails.Title=Person Details
```

Summary

In this chapter, a navigation bar was added to the Person Tab defined in "Define the Person Tab" on page 19. The following is a summary of the files that were modified during this chapter, and a reference to the targets will process each of them.

Table 5. File Summary

File	Location	Target
SimplePersonNav.nav and associated properties file	`EJBServer\components\<component-name>\clientapps`	inserttabconfiguration
SimplePerson.tab and associated properties file	`EJBServer\components\<component-name>\clientapps`	inserttabconfiguration

Working with Lists

Introduction

This chapter will introduce the following new features:

- Expandable Lists
- List Actions Menus

The `Person Search` page defined in "Define the Search Page" on page 16 will be extended to include examples of both an Expandable List and a List Actions Menu, as can be seen in the modified `Person Search` page of "Introduction"

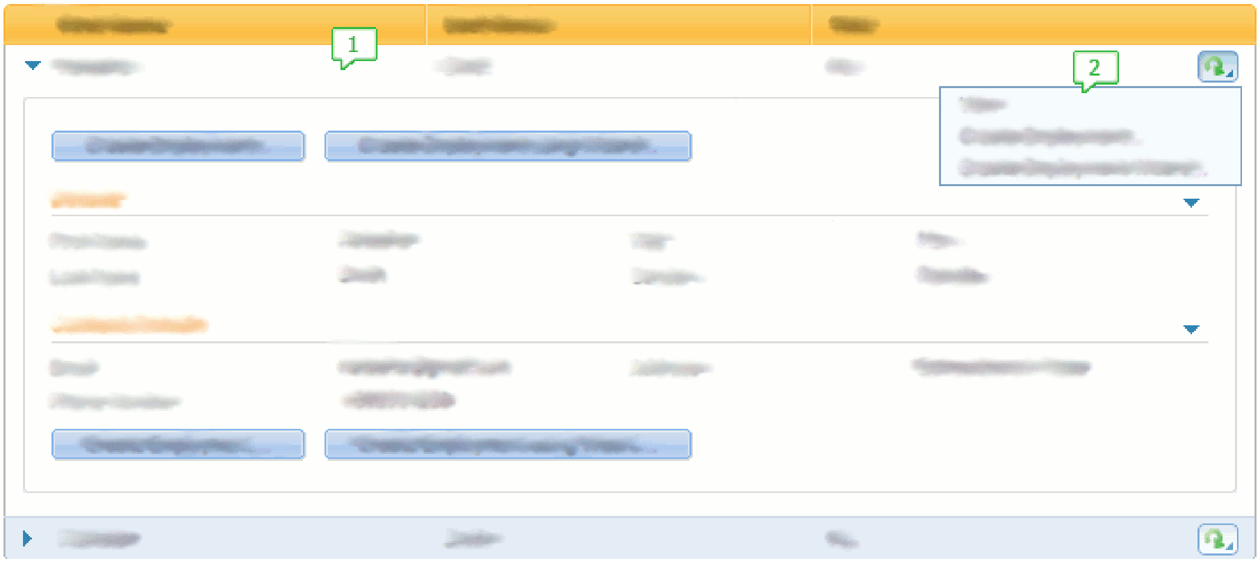

Figure 39. Expandable List With List Actions Menu

1. Expandable List
2. List Actions Menu

Define an Expandable List

The purpose of an Expandable List is to allow the user to see more information than was traditionally possible using a simple list, without navigating away from the list page itself.

An expandable list is a list where each row can be expanded by clicking on a toggle control. In the expanded state, a page relevant to the row is displayed.

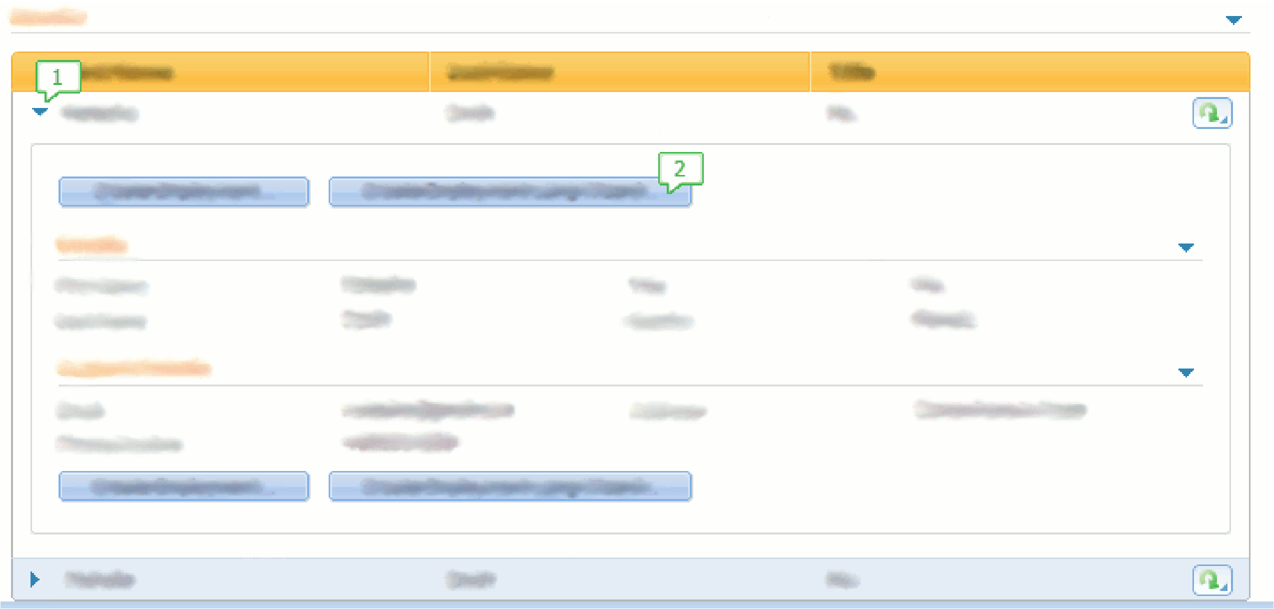

Figure 40. Person Search Page With Expandable List

1. Toggle Control

2. Page Level Action Sets

"Define an Expandable List" on page 36 shows the Person Search page which has been modified to include an expandable list. The following are some points to note:

- A Toggle Control is added to the start of each row. It allows the row to be expanded and collapsed. It is possible to expand more than one row at a time. The size of the content area adjusts automatically to accommodate this. Refer to User Interface Element 1 in "Define an Expandable List" on page 36
- Page Level Action Sets are displayed as buttons when a page is viewed inline. Refer to User Interface Element 2 in "Define an Expandable List" on page 36.

The following is the modified code for the `Person Search` page:

```
        <PAGE PAGE_ID="SimpleSearch">

  <SERVER_INTERFACE NAME="ACTION"
    CLASS="PersonFacade"
    OPERATION="advancedSearch"
    PHASE="ACTION"/>

  ...

  <LIST TITLE="List.Title.Results">
         <DETAILS_ROW>
      <INLINE_PAGE PAGE_ID="SimplePerson">
        <CONNECT>
          <SOURCE NAME="ACTION" PROPERTY="personID"/>
          <TARGET NAME="PAGE" PROPERTY="personID"/>
        </CONNECT>
      </INLINE_PAGE>
    </DETAILS_ROW>
        ...
  </LIST>
</PAGE>
```

Figure 41. SimpleSearch.uim

A new element, the `DETAILS_ROW`, has been added to the `LIST`. This defines the inline page that will be displayed when a row is expanded, including the parameters that will be passed to that page for each row.

Define a List Actions Menu

List Actions Menus allow a set of actions associated with a particular row to be encapsulated into a menu.

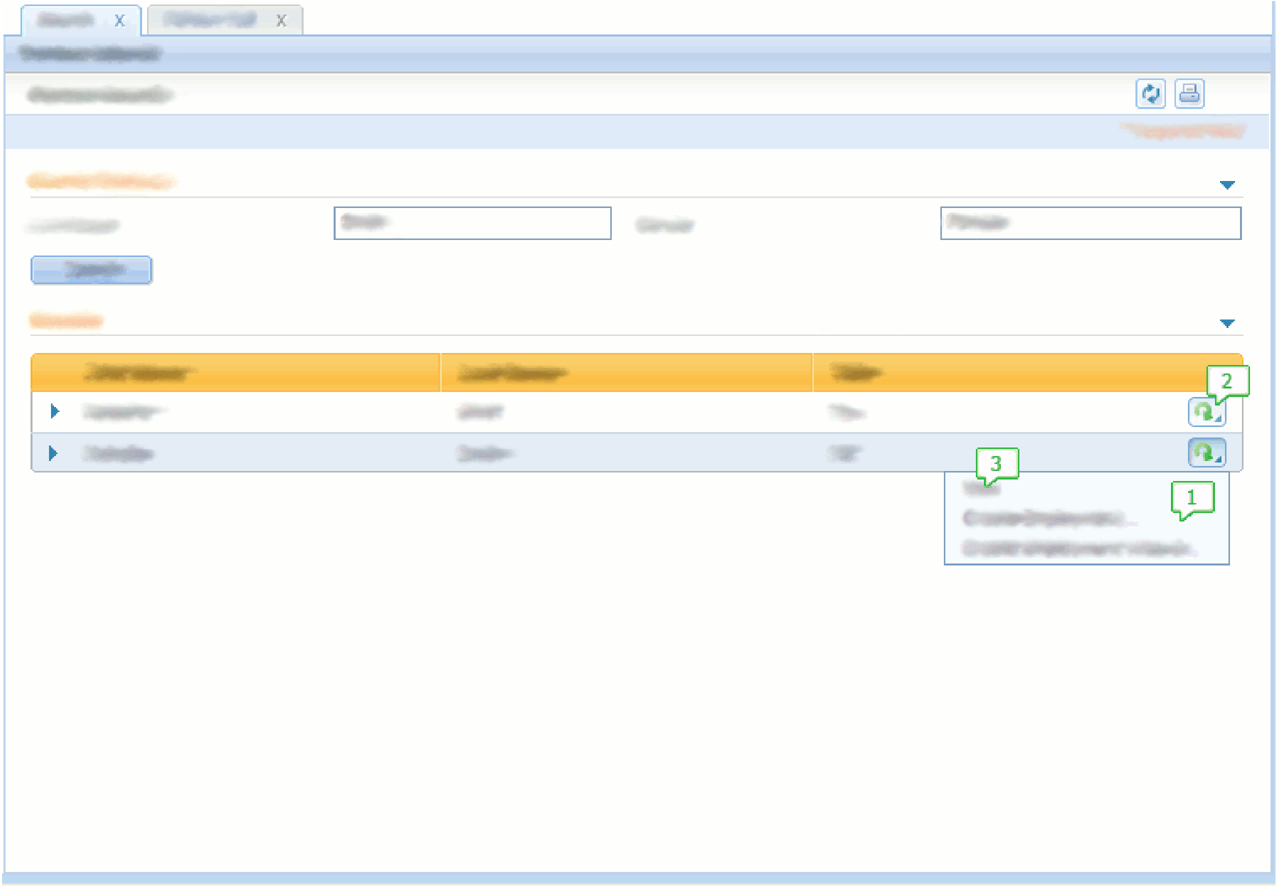

Figure 42. Person Search Page With List Actions Menu

1. Expanded List Actions Menu
2. List Actions Menu Icon
3. View Action Control

"Define a List Actions Menu" on page 37 shows the `Person Search` page which has been modified to include a list action menu. The following are some points to note:

- At the end of each row, an list actions menu icon is displayed. Clicking on this will expand the List Actions Menu.
- In this example, as a view action control was added to the List Actions Menu, the Actions column (refer to the first column in User Interface Element 4 of "Define the Search Page" on page 16) was removed.

The following is the modified code for the `Person Search` page:

```
        <PAGE PAGE_ID="SimpleSearch">

   ...

  <LIST TITLE="List.Title.Results">

     ...
         <ACTION_SET TYPE="LIST_ROW_MENU">
      <ACTION_CONTROL LABEL="Control.Label.View">
        <LINK PAGE_ID="SimplePerson">
          <CONNECT>
            <SOURCE NAME="ACTION" PROPERTY="personID"/>
            <TARGET NAME="PAGE" PROPERTY="personID"/>
         </CONNECT>
        </LINK>
      </ACTION_CONTROL>

      <ACTION_CONTROL LABEL="Control.Label.CreateEmployment">
        <LINK PAGE_ID="CreateEmployments" OPEN_MODAL="true">
          <CONNECT>
            <SOURCE NAME="ACTION" PROPERTY="personID"/>
            <TARGET NAME="PAGE" PROPERTY="personID"/>
          </CONNECT>
        </LINK>
      </ACTION_CONTROL>
      <ACTION_CONTROL
         LABEL="Control.Label.CreateEmploymentWizard">
         <LINK PAGE_ID="CreateEmploymentWizard_pageOne"
           OPEN_MODAL="true">
           <CONNECT>
             <SOURCE NAME="ACTION" PROPERTY="personID"/>
             <TARGET NAME="PAGE" PROPERTY="personID"/>
           </CONNECT>
        </LINK>
      </ACTION_CONTROL>
    </ACTION_SET>

    <!-- Removing Actions Column -->
    <!--<CONTAINER LABEL="Container.Label.Actions">
       <ACTION_CONTROL LABEL="Control.Label.View">
        <LINK PAGE_ID="SimplePerson">
          <CONNECT>
            <SOURCE NAME="ACTION" PROPERTY="personID"/>
            <TARGET NAME="PAGE" PROPERTY="personID"/>
          </CONNECT>
        </LINK>
      </ACTION_CONTROL>
    </CONTAINER>-->
        ...

   </LIST>
</PAGE>
```

Figure 43. SimpleSearch.uim

An ACTION_SET containing the three action controls has been added to the list. The attribute TYPE has been set to LIST_ROW_MENU, indicating that the action controls in this set are to be displayed on a List Actions Menu.

As the View action control has been added to the List Actions Menu, the column containing it is no longer necessary, and thus the UIM code for this has been commented out.

Localizable labels for the new action controls are placed in the corresponding `.properties` file content. For example:

```
Control.Label.CreateEmployment=Create Employment
Control.Label.CreateEmploymentWizard=Create Employment Wizard
```

Summary

In this chapter, the Person Search page has been modified to include both an Expandable List and a List Actions Set.

A summary of the files modified and the targets that will process each of them is detailed below.

Table 6. Columns

File	Location	Target
SimpleSearch.uim and associated properties file	`webclient\components\<component-name>\`	client

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law.

IBM Japan Ltd.

19-21, Nihonbashi-Hakozakicho, Chuo-ku

Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

© Copyright IBM Corp. 2012, 2014

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

Dept F6, Bldg 1

294 Route 100

Somers NY 10589-3216

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources.

IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Privacy Policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies or other similar technologies that collect each user’s name, user name, password, and/or other personally identifiable information for purposes of session management, authentication, enhanced user usability, single sign-on configuration and/or other usage tracking and/or functional purposes. These cookies or other similar technologies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and

IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/us/en/copytrade.shtml.

Java and all Java-based trademarks and logos are registered trademarks of Oracle and/or its affiliates.

Other names may be trademarks of their respective owners. Other company, product, and service names may be trademarks or service marks of others.

Printed in USA