



Developing User Interfaces

© 1997–2004 edocs® Inc. All rights reserved.

edocs, Inc., One Apple Hill Dr., Natick, MA 01760

The information contained in this document is the confidential and proprietary information of edocs, Inc. and is subject to change without notice.

This material is protected by U.S. and international copyright laws. edocs and eaPost are registered in the U.S. Patent and Trademark Office.

No part of this publication may be reproduced or transmitted in any form or by any means without the prior written permission of edocs, Inc.

eaSuite, eaDirect, eaPay, eaCare, eaAssist, eaMarket, and eaXchange are trademarks of edocs, Inc.

All other trademark, company, and product names used herein are trademarks of their respective companies.

Printed in the USA.

Preface

In This Section

Using this Manual.....	iv
Finding the Information You Need	x
If You Need Help.....	xiv

Using this Manual

Welcome to Developing User Interfaces.

This manual covers building user interfaces of edocs Telco Service & Analytics Manager applications.

Before You Get Started

You should be familiar with the following:

- Your application architecture
- Programming Java and Java Server pages

Who Should Read this Manual

This manual is for developers and project managers who are responsible for developing the user interface.

However, there are other topics covered in this manual that may interest other members of the project development team.

- **Administrators**

You will find information about the different components that make up the user interface. You can learn the location of the different files which make up the user interface.

- **Developers**

This manual is for building user interfaces for your solution. You learn how to write JSPs that use the Presentation Manager JavaServer Page framework. You also learn how to group and program sets of JSPs. These sets, called channels, allow users to access the same solution by using different devices and protocols.

You also learn how to use the framework to create new workflows, customize menus, and manage personalization information to create interactive and customizable user interfaces.

- **Project Architect**

You can use the information in this manual to learn about channels and how they work. You can learn about the components and the flexibility of your solution when it is based on channels built on a common framework.

- **Project Manager**

You will find information about channels and the Presentation Manager JavaServer Page framework which are important when developing user interfaces. You may also be interested in reading about personalization data, menus and workflows as the their characteristics may influence how you go about developing the user interfaces of your solution.

How this Manual is Organized

This manual contains the following chapters:

- **Overview of Developing User Interfaces**

This chapter covers the basics of building User Interfaces for your solutions.

It contains information about:

- User interfaces and your solution
- Developing user interfaces

- **Overview of the JSPF**

This chapter covers the JavaServer Page Framework (JSPF).

It contains information about:

- The JSPF and its components
- Properties and configuration files
- JSPF classes
- Framework JSPs
- MyWeb Sample Application
- Sample Data

- **Working with the Presentation Logic Studio**

This chapter covers using the Presentation Logic Studio.

It contains information about:

- How the Presentation Logic Studio works
- Overview of Building Applications with the Presentation Logic Studio

- **Building Applications**

This chapter covers building applications with the PLS.

It contains information about:

- Creating the design file
- Working with application properties
- Working with JSPs

- **Declaring Actors**

This chapter covers actors and security settings.

It contains information about:

- Actors and Application security
- Applying security to application elements

- **Declaring JSPs**

This chapter covers working with JSPF-based JSPs.

It contains information about:

- Working with JSPs
- Writing a JSPF-based JSP
- Debugging JSPs

- **Building Page Flows**

This chapter covers building Page Flows.

It contains information about:

- Page flows and their elements
- Creating page flows

- **Building Menus**

This chapter covers building menus.

It contains information about:

- Building menus
- Specifying display order

- **Localizing Applications**

This chapter covers languages and localizing application strings.

It contains information about:

- Setting and using languages
- Specifying strings
- Localizing strings

What Typographical Changes and Symbols Mean

This manual uses the following conventions:

TYPEFACE	MEANING	EXAMPLE
<i>Italics</i>	Manuals, topics or other important items	Refer to <i>Developing Connectors</i> .
Small Capitals	Software and Component names	Your application uses a database called the CID.
Fixed Width	File names, commands, paths, and on screen commands	Go to <code>//home/my file</code>

Obtaining edocs Software and Documentation

You can download edocs software and documentation directly from Customer Central at <https://support.edocs.com>. After you log in, click on the Downloads button on the left. When the next page appears, you will see a table displaying all of the available downloads. To search for specific items, select the Version and/or Category and click the Search Downloads button. If you download software, an email from edocs Technical Support will automatically be sent to you (the registered owner) with your license key information.

If you received an edocs product installation CD, load it on your system and navigate from its root directory to the folder where the software installer resides for your operating system. You can run the installer from that location, or you can copy it to your file system and run it from there. The product documentation included with your CD is in the Documentation folder located in the root directory. The license key information for the products on the CD is included with the package materials shipped with the CD.

Finding the Information You Need

The product suite comes with comprehensive documentation set that covers all aspects of building solutions based on the edocs Telco Service & Analytics Manager. You should always read the release bulletin for late-breaking information.

Getting Started

If you are new to the edocs Telco Solutions, you should start by reading *Introducing Telco Service & Analytics Manager Applications*. This manual contains an overview of the various components along with the applications and their features. It introduces various concepts and components you must be familiar with before moving on to more specific documentation. Once you have finished, you can read the manual which covers different aspects of working with the application. At the beginning of each manual, you will find an introductory chapter which covers concepts and tasks.

Designing Your Solution

While reading *Introducing Telco Service & Analytics Manager Applications*, you should think about how the different components can address your solution's needs.

You can refer to *Developing Telco Service Manager (TSM)* for information about extending the object model, application security, and other design issues. The *CID Reference Guide* also gives you the information about how the information in your solution is managed and stored.

You can refer to *Developing Telco Analytics Manager (TAM)* for information about customizing the database, synchronizing data with TSM, loading data from external invoice files, and other design issues. The *CBU Reference Guide* also gives you the information about how the information in your solution is managed and stored. You should also read the section on integrating TAM with TSM in *Developing Telco Analytics Manager (TAM)*.

You can also read the introduction of *Developing Connectors* for information about integrating your solution.

Installing Telco Service & Analytics Manager Applications

You should start by reading the Release Bulletin. For detailed installation and configuring information, refer to *Installing Telco Service & Analytics Manager Applications*. This manual covers installing applications on one or more computers. It also contains the information you need to configure the different components you install.

You might also refer to *Developing Telco Service & Analytics Manager Applications* and *Developing Connectors* as these manuals contain information on customizing applications and working with other software.

If you are upgrading, be sure to read *Migrating Telco Service & Analytics Manager Applications*.

Building Your Solution

If you are designing and programming your solution, you have several different sources of information. If you are programming the user interface of the solution, you should read *Developing User Interfaces*. You also refer to the *BLM Specification* and *JSPF specification* for detailed information about programming the user interface. For configuring the various components, you refer to *Installing Telco Service & Analytics Manager Applications* and sections in other documents which deal with the component to configure.

If you are designing and programming TAM, you have several different sources of information. If you are programming the user interface of the solution, you should read *Developing Reports*. You also refer to the *QRA API Specification* and the *QRA Configuration File Reference Documentation* for detailed information about the different components you can use to build reports. For configuring the various components, you refer to *Installing Telco Service & Analytics Manager Applications* and sections in other documents which deal with the component to configure.

If you are working with the business logic of your solution, you should read *Developing Telco Service Manager (TSM)*. You can also refer to the *BLM Reference Guide* for more information about the design and structure of the BLM object model. For information about how this information is stored, you should refer to the *CID Reference Guide* along with the *CID Reference* documentation for your database. In order to develop your application, you most likely will need to install and run the Loopback Connector. This component mimics back-end applications for development purposes. For information about installing and running this component, refer to *Using the Loopback Connector*.

If you are working on the data warehouse side of TAM, you should read *Developing Telco Analytics Manager (TAM)*. For more information about the design and structure of the CBU, you should refer to the *CBU Reference Guide* along with the *CBU Reference* documentation for your database. You should also read *Developing Telco Analytics Manager (TAM)* for information about synchronizing data between the TAM and *Telco Service Manager (TSM)*. In this manual, you will also find information about loading data in both the CBU and the CID.

For more information about integrating your application, you should read *Building Connectors* to learn how Telco Service & Analytics Manager applications work with different software.

Integrating Your Solution

If you are involved in configuring your solution to work with Operation Support Software (OSS), you should read *Building Connectors*. This manual helps you understand the integration architecture and shows you how to build connectors to connect to today's market-leading OSS software. You can also read *Using the Loopback Connector* for information about a connector built for development purposes. Other manuals you can refer to for information about configuring your application include *Introducing Telco Service & Analytics Manager Applications*, *Developing Telco Analytics Manager (TAM)*, and *Developing Telco Service Manager (TSM)*.

Managing Telco Service & Analytics Manager Applications

If you are responsible for managing Telco Service & Analytics Manager applications, you should read the *Installing Telco Service & Analytics Manager Applications* for information about configuring various components and information about working with different application servers. *Administering Telco Service & Analytics Manager Applications* covers what you need to know about managing your solution at runtime. For information about OSS systems, you should read *Building Connectors*.

If You Need Help

Technical support is available to customers who have valid maintenance and support contracts with edocs. Technical support engineers can help you install, configure, and maintain your edocs application.

edocs provides global Technical Support services from the following Support Centers:

US Support Center

Natick, MA
Mon-Fri 8:30am – 8:00pm US EST
Telephone: 508-652-8400

Europe Support Center

London, United Kingdom
Mon-Fri 9:00am – 5:00 GMT
Telephone: +44 20 8956 2673

Asia Pac Rim Support Center

Melbourne, Australia
Mon-Fri 9:00am – 5:00pm AU
Telephone: +61 3 9909 7301

Customer Central

<https://support.edocs.com>

Email Support

<mailto:support@edocs.com>

When you report a problem, please be prepared to provide us the following information:

- What is your name and role in your organization?
- What is your company's name?
- What is your phone number and best times to call you?
- What is your e-mail address?
- In which edocs product did a problem occur?
- What is your Operating System version?
- What were you doing when the problem occurred?
- How did the system respond to the error?
- If the system generated a screen message, please send us that screen message.

- If the system wrote information to a log file, please send us that log file.

If the system crashed or hung, please tell us.

Contents

Preface	iii
<hr/>	
Overview of Developing User Interfaces	21
<hr/>	
About User Interfaces and the Personalization Manager	22
About Developing User Interfaces	23
Overview of the JSPF	25
<hr/>	
About the JSPF Framework	26
Features of the JSPF Framework	27
JSPF Framework Components	28
About the JSPF Properties file	29
About the JSPF Classes	31
Application Framework JSPs	32
framework_start.inc and framework_end.inc	33
framework_head.inc and framework_tail.inc	33
Event Zone	33
Form Handlers	34
Logic Handlers	35
Helpers	36
Menus	37
About the MyWeb Application	38
MyWeb Features	38
Graphical Charts	40
About the Sample Data	44
Contents of the Sample Data	44
Creating the CID Database with Sample Data	46
Working With the Presentation Logic Studio	49
<hr/>	
About the PLS	50
Before You Get Started	51
Overview of Building an Application	52
Building Applications	53
<hr/>	
About Building Applications	54
Working with Design Files	55
Working with Application Properties	57
About the Default Properties	57
Adding Channels	62
Declaring the Channel Media	62
Example of Creating Support of a New Channel	62

Declaring Actors **63**

About Actors and Security	64
Working with User Roles and Organization Types	65
Specifying Permissions	67

Declaring JSPs **69**

About Declaring JSPs	70
Working With JSPs	71
Writing a Framework JSP	74
Declaring JSPF Framework variables	75
Including the JSPF Framework JSPs	76
Managing internal form JSPF Framework objects	77
Managing internal JSPF Framework objects	78
Writing JSPs	79
Debugging a Framework JSP	80
Enabling Debug Mode	80
Displaying Information in the Browser	81
Reloading JSPF Configuration Information	82

Building Page Flows **83**

About Page Flows	84
About Page Flow Elements	87
Working With Page Flow References	88
Configuring Page Flows	90
Creating In Steps	93
To create In steps	93
Creating Out Steps	94
To create Out steps	94

Working with Functional Steps	95
Configuring Functional Steps	97
Connecting Steps	98
Building Menus	99
<hr/>	
About Menus	100
Working with Menus	101
Configuring Menus	103
Specifying Display Order	104
Localizing Applications	105
<hr/>	
About Localizing Applications	106
Working with Strings	107
Working with Languages	109
Localizing Strings	111
Index	113
<hr/>	

CHAPTER 1

Overview of Developing User Interfaces

In This Section

About User Interfaces and the Personalization Manager.....	22
About Developing User Interfaces	23

About User Interfaces and the Personalization Manager

The Personalization Manager is the component which handles the user interface of your application. This is the entry point for users of your application and is considered as the presentation layer of the application.

Because users access information with different devices, the Personalization Manager is built around channels. In general, a channel corresponds to one of the different types of devices users can use to access the application.

For each channel, the Personalization Manager has a corresponding set of JSPs. This way the Personalization Manager can tailor information and features for each type of device because they each have their own technical and practical constraints. For instance, the amount of information displayed by mobile Internet devices is different from what users see on the Web. Not only that, but the Internet is based on HTML whereas the mobile Internet devices use another language called WML. These sets of JSPs are referred to as application channels or channels for short.

No matter which channel you use for the user interface of your application, there are several basic application functions and features which are standard to all applications. The Personalization Manager comes with a JSP framework called the JSPF. This framework handles these basic functions and creates the foundation of any Personalization Manager-based Application.

You use the Presentation Logic Studio (PLS) to create the workflow of your application. You use this tool to bring together the JSP and create page flows which allow your users to carry out a specific task.

About Developing User Interfaces

Before you start developing your user interface, you need to become familiar with the JSPF and the PLS. The JSPF uses JSPs to build an application framework that provides seamless integration of the application JSPs and the CSS Engine. The JSPF provides a number of key features:

- Easy customization of workflow, language, available features, and so on.
- Session management, form-handling and exception handling
- Separation of application code and user presentation, enabling parallel outsourcing of the operational and image aspects of site design

By using the JSPF as the foundation of your applications, you can separate the presentation logic from basic application tasks. The JSPF also reduces development time because you do not have to rewrite all of these basic tasks for the Personalization Manager JSPs.

Once you write your JSPF-based JSPs, you use the PLS to create Page Flows , Menus, Configure the security and manage the localization of your application.

Developing user interfaces involves:

- Becoming familiar with the JSPF
- Creating the Demo Database
- Using the Presentation Logic Studio to:
 - Create the Application
 - Declare JSPs
 - Declare Actors
 - Build Page Flows
 - Build Menus
 - Localize Strings

CHAPTER 2

Overview of the JSPF

In This Section

About the JSPF Framework	26
Application Framework JSPs	32
About the MyWeb Application	38
About the Sample Data	44

About the JSPF Framework

One of the obvious drawbacks to developing JSPs is the mix of Java code and HTML. Not only does this represent a problem when trying to read and code JSPs, it also has an impact on performance and maintenance. Large-scale, robust applications can have a very large number of JSPs and this can lead to a lot of duplicated code. Each page has to carry out user authentication, error checking, and so on. When users connect to an Telco Service Manager (TSM), any JSP must carry out a number of tasks before it can begin working with the BLM. For example, users must be authenticated, their roles and scopes must be checked, and so on.

The Java Server Page Framework (JSPF) is a set of JSP pages and Java classes and configuration files that contain the code that is required by the sets of JSPs in the Personalization Manager. You can speed up your development of JSPs by using the framework to handle basic tasks and concentrate on coding the access to the BLM. The JSPF can be considered the foundation of your channels.

Features of the JSPF Framework

This open and customizable JSPF framework contains all the code you need to quickly and easily do the following:

FUNCTION	DESCRIPTION
Authentication	Checks at the start of the JSP if the user has been authenticated according to the JSPF configuration file.
Form handling	Calls the function that processes the form associated with the JSP.
Logic handling	Calls to the BLM for processing are isolated in special JSPs. By using these JSPs, you can easily separate the processing and the presentation of your application.
Localization	Displays a character string in a given language.
Exception handling	When an error occurs, intercepts the error and displays a message using the JSP defined in the JSPF configuration file.
Role verification	Checks that the role of the user matches the role of the JSP (ex: CUSTADMIN or SUBSCRIBER).
Menus	For a given JSP, displays and manages the menu items.
Basic workflow management	As the user navigates through pages, the JSPF keeps track of the information stored as a variable and passes information between JSPs that determine the displayed contents.

JSPF Framework Components

The JSPF framework contains a set of JSPs and other files needed to configure and use the JSPF Framework.

The JSPF Framework components include:

- JSPF Framework JSPs
 - Application Framework JSPs.
These files are the foundation of the JSPF Framework. They are located in:
`<home_dir>/channels/<channel_name>/fwk.`
 - form handler JSPs
These files contain JSP code that read the data to be displayed. They are to be included in your display JSPs and they have an `.inc` extension. They are located in:
`<home_dir>/channels/<channel_name>/form_handler.`
 - logic handler JSPs
These files contain generic methods that channel JSPs use to interact with the BLM. These pages then use display pages to present the data. They are located in:
`<home_dir>/channels/<channel_name>` and are prefixed with `logic_`
 - helper JSPs
These files contain helper methods for different framework needs. They are to be included in your JSPs and they have an `.inc` extension. They are located in:
`<home_dir>/channels/<channel_name>/helpers`
- JSPF Framework Classes
The JSPF uses these classes to read the configuration files, program the features of the framework, and manage other application tasks. They are located in the `<home_dir>/channels/WEB-INF/lib/nmyjfn.jar` archive.

The JSPF Framework JSPs are JSP 1.0. Your JSPs must be based on the Java Server Pages Specification 1.0 or higher

About the JSPF Properties file

The JSPF Framework uses the `jfnApplication.properties` configuration file to set the location of the JSPF Framework configuration file and which channel to use for the application. This file is located in `<home_dir>/channels/WEB-INF/classes/nmycfg/jfn`.

This file has the following sections:

- Character encoding settings
- Default channel settings

For each channel, this file specifies:

- The location of the Presentation Logic design file (`.plad`)
- The associated URL path.

When this path is found in the URL requested by the browser, this media is selected.

The location of the JSPF Framework configuration file has the syntax:

```
media.<channel_name>=<plad file path>
```

This setting can be one of the following:

- Absolute path by entering the full path of the Presentation Logic design file
- Classpath by using the following format:

```
media.<channel_name>=res:<classpath>
```

The associated URL path of the channel has the syntax:

```
media.<channel_name>.path1=<URL path of the application>
```

EXAMPLE OF JSPFAPPLICATION.PROPERTIES

```
# List of media installed in this application server
# default => JSP pages. It will be the default to be used when the user-agent is not recognized

media.encoding=ISO-8859-1
media.reencoding=8859_1

media.default=res:nmycfg/jfn/MyWeb.plad
media.default.path1=/MyWeb
```


About the JSPF Classes

To help you program and build applications based on the JSPF framework, you use the extensive set of Java classes. These classes help you manage any JSPF-based application.

The `com.netonomy.jfn` package manages the JSPF framework configuration file and your application.

This package is in the `<home_dir>/channels/WEB-INF/lib/nmyjfn.jar` archive.

For more information about the classes and the methods in this package, refer to the *JSPF API Reference Documentation*.

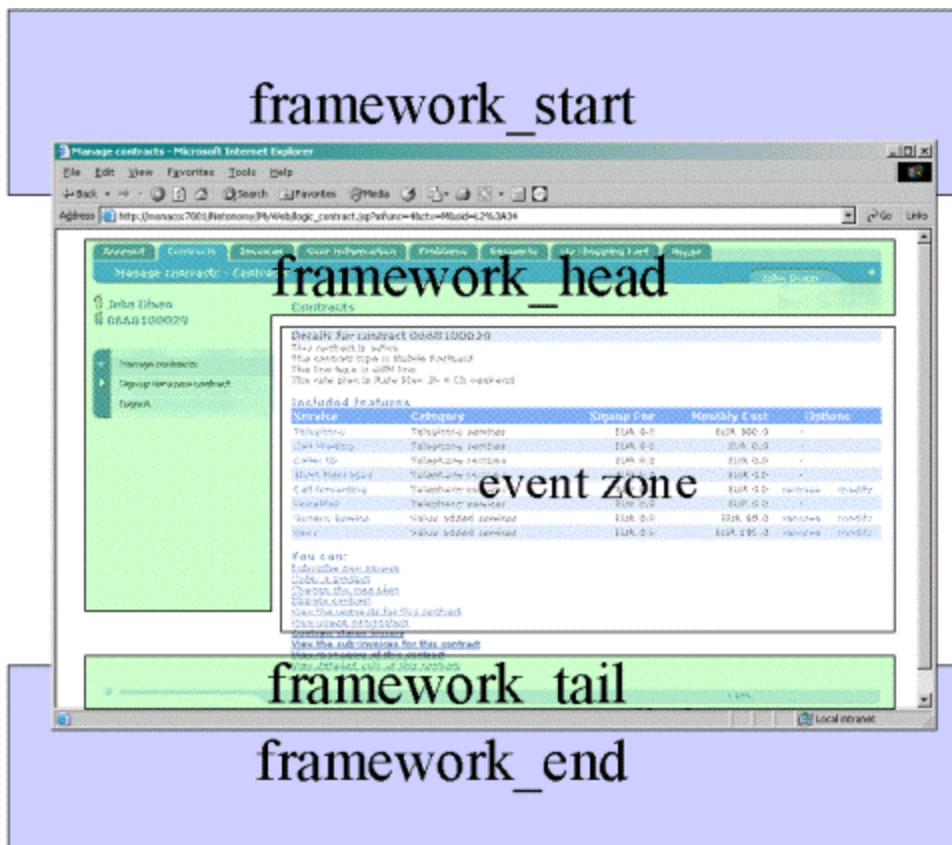
Application Framework JSPs

These JSPs contain the code to carry out most of the basic functions of the framework so you can concentrate on programming the access to information in the CID. The Personalization Manager JSPs that make up the MyWeb channel also use the framework files.

The Personalization Manager JSPs are installed in the following directories:

- `<home_dir>/channels/MyWeb` for the Web application

This is a diagram of the different **JSPF** framework JSPs and how they relate to a MyWeb page.



framework_start.inc and framework_end.inc

This JSP is the first part of the main framework code for JSPs.

If you include `framework_start.inc`, you must also include `framework_end.inc` at the end of your JSP. Refer to `login.jsp` as an example.

This JSP carries out the following functions:

- Declares the variables for the JSP
- Gets the session object
- Gets the JSP configuration
- Checks if the session is new. If yes, redirects the user to the login page if they do not come from this page
- Retrieves the BLM session and adds it to the session manager
- Checks authentication and redirects user to the login page if it fails
- Checks role and redirects to a message if this fails
- Creates a repository object for errors
- Creates a repository object for results
- Calls the form or logic handler declared in the functional step
- Finds the menu to display

framework_head.inc and framework_tail.inc

These JSPs start and end the static HTML code of the application page.

The root `framework_start` JSP contains the test of the graphical chart to use and the include statement of the corresponding `framework_start`.

These files are located in subdirectories corresponding to the graphical chart.

The `framework_head` contains simple html tags. You can add a banner ad rotator here.

The `framework_tail` contains the closing tags of the html.

Event Zone

This is your JSP that contains the display of information or the application's context. The heart of the system, you can easily write JSP to interact with the framework files and the BLM to build applications. These JSPs are referred to as display pages.

Form Handlers

A form handler is a JSP dedicated to preprocessing information to be sent to a JSP for display.

A form handler is simply a method declared in a form handler JSP. The method name begins with `formHandler_`.

The form handler JSP contains a set of preprogrammed functions you use to:

- Speed up development
This code can be used by all the JSPs that make up your application. And as it is shared code, you can easily modify the behavior of your application without having to change several JSPs.
- Isolate display logic from display processing
This code handles extracting information from the CID and preprocessing it. You then send the results to the display page.

For example, the `form_handler/getContract.inc` includes the following form handlers:

- `formHandler_bulkAddContractsToManagerReport`
- `formHandler_bulkRemoveContractsFromManagerReport`
- `formHandler_getContract`
- `formHandler_getContractAndServices`
- `formHandler_getContracts`
- `formHandler_getContractsToRemoveFromManager`
- `formHandler_getContractsToRemoveFromOrgView`
- `formHandler_getNewContract`
- `formHandler_prepareUnassignContractConfirmation`

These `formHandler` JSPs are executed by the JSPF framework just before the page starts to display.

You can create your own form handlers.

Logic Handlers

A JSPF logic handler JSP is a JSP dedicated to workflow and business rules processing. A logic handler processes and calls the next JSPF logic handler to execute or redirect to a JSPF display JSP.

When a JSPF logic handler calls a JSPF display JSP, you can also call a form handler.

JSPF logic handlers are functions grouped in a file which act as a library located in the channel directory. The logic handler JSP names begin with `logic_` to make them easy to identify and program. The same applies to the methods found inside the JSPs.

For example, in `logic_member.jsp` you will find the following methods:

- `logic_childCreateMemberForNext`
- `logic_searchMembers`

These logic handlers are executed by the JSPF framework at the beginning of the JSP. These pages do not display anything, they handle logic. Once they finish processing, they call a JSPF display JSP via a JSP server redirect or can call another process if needed.

You can create your own logic handlers.

Helpers

JSPF helper files are JSPs that are used as helpers for different components of the application.

They include:

- `logic_helper.inc`
This logic helper JSP contains methods to simplify `logic_handler`, `form_handler` and display JSP development.
- `rendering_helper.inc`
This logic helper JSP contains methods to customize the way URLs are built in the application.
- `report_helper.inc`
This logic helper JSP contains methods to build reports.

Menus

In applications, JSPF framework menus change according to the role of the user that logs into the system. For instance, residential users and business users may have different menus because their needs are different. Certain features that are required for a specific set of users might not be needed by another set of users.

USER ROLE	FRAMEWORK MENU
Individual Subscriber	I_SUBSCRIBER
Business Subscriber	B_SUBSCRIBER
Business Administrator for Levels	B_CUSTADMIN_L
Business Administrator for Members	B_CUSTADMIN_M
Dealer	DEALER
Supplier	SUPPLIER
Business Contract Administrator	B_CONTADMIN_M
Telco	DEALER
Telco Account Manager	DEALER
Telco Senior Account Manager	DEALER

Menus are selected in the `framework_start.jsp`

About the MyWeb Application

edocs Telco Service & Analytics Manager comes with the following application:

- MyWeb

This set of JSPs builds the user interface for users using the Internet.

This application uses the JSPF for common application features.

This application includes:

- `MyWeb.plad` JSPF Framework Configuration file
- JSPs
- Set of Web site graphics and files

In order for applications to run correctly, you need to have information in the CID that the application can access and manage. The CID comes with sample data that you use while working with channels. The sample data are different from the reference data. Sample data cover users, their contracts, commercial offers, rate plans and so on.

The JSPF framework JSPs have calls to deprecated JSPF APIs. As this version introduced a new way to design applications, these calls remain in the JSPs in order to insure backwards compatibility.

For more information about:

- the sample data, refer to *About the Sample Data* in this manual.
 - JSPF APIs, refer to the JSPF API HTML reference documentation.
-

MyWeb Features

This channel includes the following features:

Managing Users

- Logging in
- Logging out
- Creating logins
- Changing passwords

- Changing languages

Acquiring and Activating Customer Accounts

- Creating contracts
- Migrating contracts
- Creating new organizations
- Adding levels to organizations
- Adding billing information to organizations
- Adding members to organizations
- Creating Personalization Data

Enabling Self-service Account Management

- Modifying payment information
- Adding legal contacts
- Adding billing contacts
- Modify billing contacts
- Modify legal contacts
- Modifying contracts
- Searching for organizations
- Searching organization hierarchies
- Browsing organization hierarchies
- Changing rate plans
- Listing services
- Adding services
- Changing parameters of a services
- Replacing services
- Removing services
- Listing rate plans
- Grouping contract modifications
- Ordering with a shopping cart
- Ordering documentation
- Viewing usage information
- Listing prepaid packages
- Recharging prepaid contracts
- Approving orders

- Managing contracts

Reporting and Resolving Problems

- Creating trouble tickets
- Searching for trouble tickets
- Viewing trouble ticket details
- Modifying trouble tickets
- Declaring of loss or theft

Viewing Account Information Online

- Searching for invoices
- Viewing invoice details
- Searching payments
- Viewing payments
- Creating organization views
- Modifying organization views

System Features

- Displaying requests
- Auditing
- Generating user events
- Listing user events

Graphical Charts

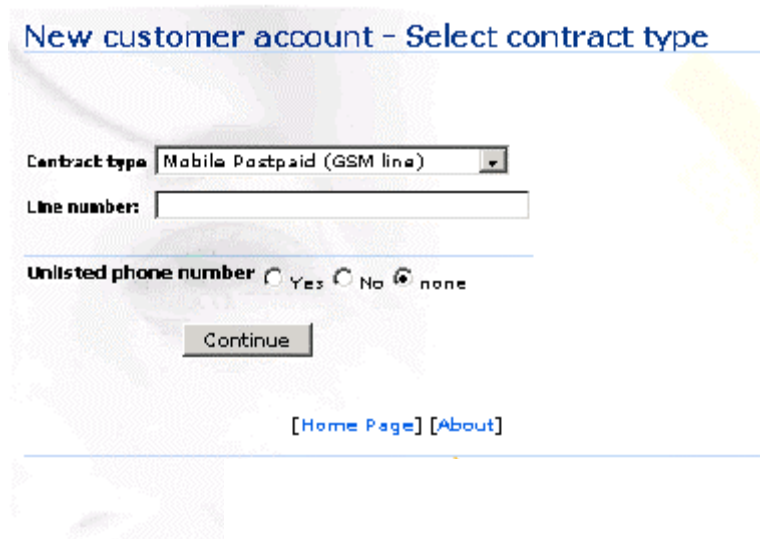
MyWeb comes with the following basic graphical charts:

- Guest
- Rich
- Simple

Guest Graphical Chart

When a user is a guest and signing up for an account, the application uses the Guest graphical chart.

An example of the graphical chart:



The screenshot shows a web form titled "New customer account - Select contract type". The form includes a dropdown menu for "Contract type" with "Mobile Postpaid (GSM line)" selected, a text input field for "Line number:", and radio buttons for "Unlisted phone number" with options "Yes", "No", and "none" (selected). A "Continue" button is located below the radio buttons. At the bottom of the form, there are links for "[Home Page]" and "[About]".

New customer account - Select contract type

Contract type: Mobile Postpaid (GSM line)

Line number:

Unlisted phone number: ☐ Yes ☐ No ☒ none

Continue

[\[Home Page\]](#) [\[About\]](#)

Rich Graphical Chart

When a user has an account and logs in, MyWeb uses the Rich graphical chart.

An example of the graphical chart:

The screenshot shows a web application interface for managing contracts. The top navigation bar includes links for Account, Contracts, Invoices, User Information, Problems, Requests, My Shopping Cart, and Home. The main header area displays 'Manage contracts - Contracts' and the user's name 'John Dixon'.

On the left sidebar, the user 'John Dixon' is identified with ID '0660100029'. Below this, there are links for 'Manage my line', 'Sign up for a new contract', and 'Logout'.

The main content area is titled 'Contracts' and shows 'Details for contract 0660100029'. The details indicate that the contract is active, it's a Mobile Postpaid contract, the line type is GSM line, and the rate plan is Rate Plan Slt + Slt (prepaid).

Below the details is a table titled 'Included features' with the following data:

Service	Category	Signup Fee	Monthly Cost	Options
Telephony	Telephony services	EUR 0.0	EUR 210.0	-
Call waiting	Telephony services	EUR 0.0	EUR 0.0	-
Caller ID	Telephony services	EUR 0.0	EUR 0.0	-
Short Messages	Telephony services	EUR 0.0	EUR 0.0	-
Call forwarding	Telephony services	EUR 0.0	EUR 0.0	-
Voice Mail	Telephony services	EUR 0.0	EUR 0.0	-
General Service	Value services	added	EUR 0.0	EUR 0.0 remove modify
Rings	Value services	added	EUR 0.0	EUR 113.0 remove modify

Below the table, a section titled 'You can:' lists several actions: 'Subscribe new service', 'Order a product', 'Change the rate plan', 'Migrate contract', 'View the requests for this contract', 'View usage information', and 'View status history'.

At the bottom right of the main content area, there is a 'Go to top' link.

Simple Graphical Chart

When specified, MyWeb can use a Simple graphical chart.

An example of the graphical chart:

[\[Account\]](#)
[\[Contracts\]](#)
[\[Invoices\]](#)
[\[User Information\]](#)
[\[Problems\]](#)
[\[Requests\]](#)
[\[My Shopping Cart\]](#)
[\[Home\]](#)

-Manage contracts-- Sign up for a new contract -

John Dixon Logout

Details for contract D660100029

This contract is active
 The contract type is Mobile Postpaid
 The line type is GSM line
 The rate plan is Rate Plan 3h + 3h weekend

Included features

Service	Category	Signup Fee	Monthly Cost	Options
Telephony	Telephony services	EUR 0.0	EUR 300.0	-
Call Waiting	Telephony services	EUR 0.0	EUR 0.0	-
Caller ID	Telephony services	EUR 0.0	EUR 0.0	-
Short Messages	Telephony services	EUR 0.0	EUR 0.0	-
Call forwarding	Telephony services	EUR 0.0	EUR 0.0	-
VoiceMail	Telephony services	EUR 0.0	EUR 0.0	-
Generic Service	Value added services	EUR 0.0	EUR 65.0	remove modify
News	Value added services	EUR 0.0	EUR 139.0	remove modify

You can:

- [Subscribe new service](#)
- [Order a product](#)
- [Change the rate plan](#)
- [Migrate contract](#)
- [View the requests for this contract](#)
- [View usage information](#)
- [View status history](#)

About the Sample Data

In order for your channels to run correctly, you need to have information in the CID that the application can access and manage. The CID comes with sample data that you use while working with channels. Sample data include reference data (countries, and so on), a product catalog, customer/dealer/customer service representatives (CSR) data and user data.

The comprehensive sample data help you build and test your application quickly and easily as you do not need to create test data.

When installing the CID for development or demonstrations, you can use the `cidAdminTool` to:

- Install the CID with system data only
When you install the CID with system data only, you can then populate it with your own sample data.
- Install the CID with system and sample data
When installing the CID with both types of data, the CID is ready to be used for development or demonstrations.

Contents of the Sample Data

Reference Data

Along with the system data, the sample data includes reference data. In the CID, the reference data includes lists of languages, countries, and other general data.

Product Catalog Data

Corresponding to a Communication Service Providers catalog, the sample data includes service data which includes services, commercial offers, rate plans and other data.

User and Associated Customer/Dealer/CSR Information

The sample data includes a set of users and user types. This set of data includes the following logins:

The password is the same as the login.

When logging in, you must enter the login and password in lowercase.

LOGIN	ACTOR	NOTES	CHANNELS
joe	Consumer administrator	Has 1 contract	WEB
tammy	Consumer administrator	Has more than 1 contract	WEB
0660100032	Consumer administrator	Prepaid subscriber	WEB
0660100034	Consumer administrator		WEB
admadm	Business administrator	Administrator of Acme corporation	WEB
adm1, adm2, michel	Business administrator	Administrators of sub-levels of Acme corporation	WEB
contractadm	Contract administrator	Manages a set of Acme corporation contracts	WEB
bigboss	Business user	User at the top level of Acme corporation Has more than 1 contract	WEB
véro, jean, paul, victor	Business user	User at a sub-level of Acme corporation	WEB
jack	Business administrator	Administrator of Jack and Co	WEB
jim	Business user	User at the top level of Jack and Co	WEB
tph	Dealer user	User at the top level of dealer organization	WEB
herve	Dealer user	User at a sub-level of dealer organization	WEB
eur	Telco user		WEB
will	Supplier user		WEB

acctmgr	Telco business account manager	Initially in charge of Acme corporation (list of managed business customers can be changed by senior_acctmgr)	WEB
acctmgr2	Telco business account manager	Initially in charge of Jack and Co (list of managed customers can be changed by senior_acctmgr)	WEB
senior_acctmgr	Telco business senior account manager	Manages all business accounts	WEB

Creating the CID Database with Sample Data

To create the CID and populate it with sample data for the Personalization Manager channels, you use the cidAdmin tool.

When working with the demo CID Database, server components can only interact with the database using the <CID_ADMIN> account.

To create the CID with system and sample data

- 1 Go to <home_dir>/bin.
- 2 Run the CID Administration tool. Use the syntax:

```
cidAdminTool create_demo_cid_test <CID> <CID_ADMIN login>  
<CID_ADMIN password>
```

where <CID>:

 - Oracle: <instance alias>
 - DB2: <database alias>
 - SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port

When finished, the CID Administration tool displays a message.

To remove the CID

- 1 Go to <home_dir>/bin.
- 2 Run the CID Administration tool. Use the syntax:

```
cidAdminTool drop_demo_cid_structure <CID> <CID_ADMIN login>  
<CID_ADMIN password>
```

where <CID>:

- Oracle: <instance alias>
- DB2: <database alias>
- SQL Server: <database host> [:<port>] If no port is specified, the tool uses the default SQL server port

When finished, the CID Administration tool displays a message.

CHAPTER 3

Working With the Presentation Logic Studio

In This Section

About the PLS	50
Before You Get Started	51
Overview of Building an Application	52

About the PLS

The Presentation Logic Studio helps you build your application by bringing together all of the components you need to create a full-featured application for your users.

Instead of manually programming the way your JSPF-based JSPs interact, you can use the PLS to program the way they interact.

You use the PLS to do the following:

- Designing Page Flows
- Configuring security
- Localizing

Before You Get Started

If you want to use the PLS with applications created with previous editions, you may need to import the JSPF configuration file.

Once you have imported your JSPF configuration file, you can begin using the PLS to build your application.

If you do not have to migrate your application, you can begin using the PLS to customize the application template or create a new application.

Overview of Building an Application

Designing an Application involves:

- 1 Creating the Application
- 2 Declaring JSPs
- 3 Declaring Actors
- 4 Building Page Flows
- 5 Building Menus
- 6 Localizing Strings
- 7 Validating the Application

CHAPTER 4

Building Applications

In This Section

About Building Applications.....	54
Working with Design Files	55
Working with Application Properties.....	57
Adding Channels.....	62

About Building Applications

Each application includes the following components:

- **Page Flows**
Page flows are sequences of events which carry out functional tasks. A page flow can be very simple and contain two or three simple steps and a linear flow. Page flows can also be complex, with many steps with the flow depending on several different factors.
- **Menus**
Menus present different options for users to select from.
- **Strings**
Strings are the text displayed in the application's user interface. Your application also uses the declared languages to localize you application whenever required.
- **Properties**
Properties are application settings which determine the behavior of your application.
- **Permissions**
By using Roles and Organization types, Permissions control the access to different elements of the application.
- **JSPs**
The JSPs contain the code which builds the user interface and connects to the BLM.

Building Presentation Logic Applications involves:

- Creating the Presentation Layer Application Design File
- Specifying the Application Properties
- Adding Channel support

Working with Design Files

The Presentation Logic Application Design (PLAD) file is your application project file. This file is referred to as the JSPF configuration file.

The default MyWeb PLAD file is located in `<home_dir>/channels/MyWeb/WEB-INF/classes/nmycfg/jfn`.

This file contains all of the settings and configuration information of your User Interface.

Working with the design file involves:

- Creating design files
- Opening design files
- Closing design files
- Saving design files
- Validating design files
- Generating documentation

To create design files

- 1 Choose *File > New*. The *Define a new project* dialog box appears.
- 2 Browse and select the parent directory of your application home directory then choose *Select*.
- 3 Enter the name of your design file then choose *Save*.

To open design files

- 1 Choose *File > Open*. The *Open Application* dialog box opens.
- 2 Select the design file to open.
- 3 Choose *Open*. The design file opens in the Explorer.

To close design files

- 1 Choose *File > Close*. The design file closes. If any changes have not been saved, you are prompted to save or discard the changes.

To save design files

- 1 Do one of the following:
 - Choose *File > Save* to save the design file.
 - Choose *File > Save As* to save the design file with a different name.

To validate design files

Choose *Application > Check Application*. The PLS validates the application.

After verification, the PLS displays the result in the message pane. If your application is invalid or if a problem occurs during verification, the PLS displays messages to help you resolve the problem.

To generate documentation

- 1 Choose *Application > Generate HTML Documentation*. The *Export Documentation* dialog box opens.
- 2 Select the location then choose *Save*. The PLS generates the contents of the application in an easy-to-read HTML file. When finished, the PLS displays a message in the message pane.

Working with Application Properties

The Presentation Logic Application properties are properties which control the behavior of your application. Properties can be global and apply to the entire application or properties which apply to specific elements which make up your application.

Each application has a standard set of properties for applications using the JSPF. You can change the values of these default properties.

You can also create and call your own properties and use them in your application. You can create application-wide properties and properties for specific elements, such as page flows and elements. You can also use these properties to override global property settings whenever required.

Working with Properties involves:

- Setting Standard Properties
- Creating Custom Properties
- Overriding Properties
- Removing Custom Properties

About the Default Properties

The default application properties include the following:

- JSPF Settings
- Search Result Settings
- Complex Parameter Settings
- Report Processing Settings

JSPF Settings

PROPERTY	DESCRIPTION
FILE_INTERNAL_ERROR	Name of the JSP to display internal BLM error message
FILE_LOGIC_ERROR	Name of the JSP to display BLM business logic messages
FILE_LOGIN	Name of the JSP to open when an authentication check fails
FILE_MESSAGE	Name of the file to display messages when: <ul style="list-style-type: none">• the role checking fails• a session expires
GETURL_CALL_ENCODE_URL	True or False Encodes the URL when the application server is in URL rewriting mode to manage sessions without cookies. Used when calling the URL using JFNjspHelper.getURL.
HTTPSESSION_DURATION	True or False Saves the contents of the shopping cart when the session times out and log an event.
JFN_DEBUG	True or False For debugging purposes only. Displays debugging information in the application server console.
JFN_DEBUG_INFO_DISPLAY	True or False For debugging purposes only. In each displayed page, displays the Functional Step and Functional steps called by the current JSP.
JSP_BIND_HTTPSESSION	True or False Saves the contents of the shopping cart when the session times out and log an event.

Search Result Settings

PROPERTY	DESCRIPTION
MAX_ELEMENTS_SEARCH_INVOICE	Maximum number of elements returned by the database when a search invoice is executed
MAX_ELEMENTS_SEARCH_PAYMENT	Maximum number of elements returned by the database when a search payment is executed
MAX_ELEMENTS_SEARCH_TICKET	Maximum number of elements returned by the database when a search ticket is executed
MAX_ELEMENTS_SEARCH_REQUEST	Maximum number of elements returned by the database when a search request is executed

Complex Parameters Layout Settings

Property	DESCRIPTION
PARAM_CHARS	Number of chars to switch between textfield and textarea
PARAM_COMPOSITE_ITEMS	Max number of items to be displayed horizontally, if superior display them vertically (for a composite item)
PARAM_ITEM	Switch between list to of multiple choices and multi-value choice
PARAM_LINES_IN_LIST	Number of items display in a multi-value list
PARAM_LINES_IN_TEXTAREA	Number of lines in a text area
PARAM_LIST_ITEMS	Max Number of items to be displayed horizontally, if superior display them vertically (for a list of items)
PARAM_SIZE_DATE	Size of the input field for Date values
PARAM_SIZE_DATETIME	Size of the input field for Datetime values
PARAM_SIZE_FLOAT	Size of the input field for Float values
PARAM_SIZE_INTEGER	Size of the input field for Integer values
PARAM_SIZE_STRING	Size of the input field for String values
PARAM_SIZE_TIME	Size of the input field for Time values

Report Processing Settings

PROPERTY	DESCRIPTION
IS_UNREADREPORTS_CHECK_ENABLED	True or False Activates report notification.
REPORT_PROCESSING_TIMEOUT	Maximum amount of time in seconds that is allowed for a report to be processed asynchronously
UNREADREPORT_CHECK_DELAY	Number of seconds between two checks
UNREADREPORT_SESSION_CHECK_DELAY	Number of seconds between two checks for a single user session. This setting allows the application to check for a done report before the default delay setting.

To set Properties

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Expand *Properties*. The properties appear.
- 4 Select the property to modify. The properties appear in the properties pane.
- 5 In the properties pane, enter the new value of the property.
- 6 Save your changes.

To create Properties

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Right click *Properties* then select *Create New*. The *Create New* dialog box appears.
- 4 Enter the code of the property then choose OK. The property appears in the explorer.
- 5 Select the new property. The properties appear in the properties pane.
- 6 Enter the following properties:
 - Value
The value of the property.
- 7 Save your changes.

To remove Properties

- 1 Open the application.

- 2 In the Explorer, expand the application to display the components.
- 3 Expand Properties. The properties appear.
- 4 Right-click the property to delete then choose *Delete*. A confirmation dialog box appears.
- 5 Do one of the following:
 - Select *Yes* to permanently remove the element
 - Select *No* to cancel and return to the Explorer

To override Properties

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Find the element where you need to override the global property.
- 4 Right click the element then select *Create New*. The *Create New* dialog box appears.
- 5 Enter the name of the property to override then choose OK. The property appears in the explorer.
- 6 Select the new property. The properties appear in the properties pane.
- 7 Enter the following properties:
 - Value
The value of the property to be used with this element instead of the global setting.
- 8 Save your changes.

Adding Channels

A channel is associated with the channel media. By default, the application is the Web channel. However, your solution may require other support for other media, such as WAP or IVR.

Adding a new media involves:

- Declaring the media in the `jfnapplication.properties` configuration file

Declaring the Channel Media

1 Go to `<home_dir>/channels/WEB-INF/classes/nmycfg/jfn.`

2 Open `jfnApplication.properties`.

3 Declare the new media. Use the syntax:

`media.<media_name>=full path of the JSPF configuration file or the res:/ resource path`

`media.<media_name>.path1= string of the media to use that should appear in the request URL. This allows the JSPF to find the right configuration file to use for this media's client.`

Example:

`media.wap.path1= /MyWap`

`http://host/MyWap/login.jsp` is a Wap site URL.

4 Save your changes.

Example of Creating Support of a New Channel

1 Go to `<home_dir>/channels/WEB-INF/classes/nmycfg/jfn.`

2 Open `jfnApplication.properties`.

3 Add the following:

`media.palm=res:nmycfg/jfn/MyPalm.plad`

`media.palm.path1=/MyPalm`

4 Copy the `/channels/MyWeb` folder and its contents and rename it `MyPalm`.

5 Edit `MyPalm.plad` and change the following:

`<property name="APP_SKIN_URL">/palm/include/</property>`

6 Add your specific workflows.

CHAPTER 5

Declaring Actors

In This Section

About Actors and Security	64
Working with User Roles and Organization Types.....	65
Specifying Permissions	67

About Actors and Security

The concept of Actors is to determine the different users of your application. The security of your application is based on these actors.

Application security is based on the concept of granting access to different parts of the application using the different actors. Actors are divided into the following types:

- User Roles
- Organization Types

For different elements of an application, you specify which User Role, Organization Type or both a user must have in order to access the element. Together, these are referred to as Permissions.

By default, all User Roles and Organization Types have access to your application and elements. In order to restrict access, you specify the User Roles and Organization Types you want to access the element.

In the PLS, you create User Role and Organization Type references which correspond to the User Roles and Organization Types declared in the CID. Once you create these permissions, you then specify the permissions for application elements.

The PLS does not verify the existence of the specified User Roles or Organization Types. Make sure they correspond to the existing User Roles and Organization Types in the CID. It also does not verify the spelling of the User Roles and Organization Types.

For detailed information about creating and modifying User Roles refer to *Managing Access to BLM Objects* in the *BLM Reference Guide*. For information on how roles User Roles and Organization Types are managed in the CID, refer to the *CID Reference Guide*.

Working with the Security Permissions involves:

- Creating User Roles and Organization Types
- Specifying the Permissions for:
 - Page Flows
 - Functional Steps
 - Menus
 - JSPs

Working with User Roles and Organization Types

Working with User Roles and Organization Types involves:

- Creating User Roles
- Creating Organization Types
- Deleting User Roles
- Deleting Organization Types

To create User Roles

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Right click User Roles then select *Create New*. The *Create New* dialog box appears.
- 4 Enter the code of the User Role then choose OK. The User Role appears in the explorer.

User Role names may not contain any spaces or special characters.

To create Organization Types

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Right click OrgType then select *Create New*. The *Create New* dialog box appears.
- 4 Enter the code of the Organization Type then choose OK. The Organization Type appears in the explorer.

Organization Type names may not contain any spaces or special characters.

To remove User Roles

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Expand User Roles. The User Roles appear.
- 4 Right-click the User Role to delete then choose *Delete*. A confirmation dialog box appears.
- 5 Do one of the following:

- Select *Yes* to permanently remove the element
- Select *No* to cancel and return to the Explorer

To remove Organization Types

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Expand OrgType. The Organization Types appear.
- 4 Right-click the Organization Type to delete then choose *Delete*. A confirmation dialog box appears.
- 5 Do one of the following:
 - Select *Yes* to permanently remove the element
 - Select *No* to cancel and return to the Explorer

Specifying Permissions

To specify Role Codes

- 1 In the Explorer, find the element you want to assign access to by User Roles.
- 2 Expand the *Permissions* element. The *Role Codes* and *Organization Types* elements appear.
- 3 Click *Role Codes*. The properties pane displays the roles having permission to access the element.
- 4 Click *Add*. A new value appears in the properties pane.
- 5 Select an available Role Code.
- 6 Repeat as required.
- 7 Save your changes.

To specify Organization Types

- 1 In the Explorer, find the element you want to assign access to by Organization Types.
- 2 Expand the *Permissions* element. The *Role Codes* and *Organization Types* elements appear.
- 3 Click *Organization Type*. The properties pane displays the Organization Types having permission to access the element.
- 4 Click *Add*. A new value appears in the properties pane.
- 5 Select an available Organization Type.
- 6 Repeat as required.
- 7 Save your changes.

CHAPTER 6

Declaring JSPs

In This Section

About Declaring JSPs	70
Working With JSPs	71
Writing a Framework JSP	74
Debugging a Framework JSP	80

About Declaring JSPs

The Presentation Layer Application is built on JSPF-based JSPs which contain the user interface code and accesses the BLM.

In the PLS, you create a JSP reference. This reference has an associated JSP file. You use this reference in your page flows and your application. If you need to change your JSP, you do not need to change the reference in your page flows, you just change the JSP associated with the JSP reference.

The PLS also comes with a basic text editor you can use to quickly open and edit the JSP.

Working with JSPs involves:

- Specifying the location of the JSPs
- Creating JSP References
- Specifying the JSP File
- Editing the JSP
- Configuring the JSP References
- Removing JSP References

Working With JSPs

To specify the home directory of JSPs

- 1 Open the design file.
- 2 In the explorer, choose the Application. The properties appear in the properties pane.
- 3 In the properties pane, enter the following:
 - JSP Directory
The location of application JSPs.
This is for design time only.
- 4 Save your changes.

To create JSP References

- 1 Open the Application.
- 2 In the Explorer, select *JSPs*.
- 3 Right click *JSPs* then choose *Create New*. The *Choose File* dialog box appears.
- 4 Do one of the following:
 - Choose the JSP file to the JSP Reference
 - Enter the name of the file to create
- 5 Choose *OK*. The *Create New* dialog box appears.
- 6 Enter the name of the new reference.
- 7 Choose *OK*. The JSP Reference appears in the properties pane.
- 8 Save your changes.

To specify JSP pages

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Expand *JSPs*. The JSP References appear.
- 4 Select the reference to modify. The properties appear in the properties pane.
- 5 In the properties pane, enter the following:
 - URL
The JSP file name
- 6 Save your changes.

To edit JSPs

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Expand *JSPs*. The JSP References appear.
- 4 Double-click the JSP under the JSP Reference. The JSP file opens in the editor.
- 5 Edit the JSP.
- 6 Save your changes.

To configure JSP References

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Expand *JSPs*. The JSP References appear.
- 4 Expand the reference to modify. The Strings and Properties subelements appear.
- 5 Declare the Strings:
 - Create the strings which are specific to this JSP.
 - Create the strings which override the global strings in this JSP. You override a global string by creating a JSP string with the same name and different value.
- 6 Declare the Properties:
 - Create the Properties which are specific to this JSP.
 - Create the Properties which override the application Properties in this JSP. You override an application Property by creating a Property with the same name and different value.
- 7 Save your changes.

To remove JSP References

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Expand JSPs. The JSP References appear.
- 4 Right-click the JSP Reference to delete then choose *Delete*. A confirmation dialog box appears.
- 5 Do one of the following:
 - Select *Yes* to remove the reference. This deletes only the reference and not the JSP file.
 - Select *No* to cancel and return to the Explorer

Writing a Framework JSP

Every JSP file you want to use in the Personalization Manager channels must include certain elements and follow certain rules.

The rules of using the JSPF in your JSPs:

- 1 Declare JSPF Framework variables
- 2 Include the JSPF framework files
- 3 Include the form handlers that manage the processing of form data
- 4 Use JSPF implicit objects
- 5 Use the JSPF framework function to declare strings and URLs

Declaring JSPF Framework variables

When you code a JSP to use the JSPF Framework, you must declare the following variables:

- `String strJSP = "<pagecode>"`
where
`<pagecode>` corresponds to the JSP code in the PLS.

Including the JSPF Framework JSPs

Include the following framework files:

- In the JSP Header:

```
<%@ include file="fwk/framework_start.inc" %>  
<%@ include file="fwk/framework_head.inc" %>  
<%@ include file="helpers/rendering_helper.inc" %>
```

- In the JSP Footer:

```
<%@ include file="fwk/framework_tail.inc" %>  
<%@ include file="fwk/framework_end.inc" %>
```

Managing internal form JSPF Framework objects

Every form managed by a JSP can provide a form handler that is automatically called by the framework engine.

In the JSP Header, include the form handlers:

- `<%@ include file="form_handler/setContact.inc" %>`

Managing internal JSPF Framework objects

Include the form handlers you want to use to manage the data and context of the page along with the following objects:

- `session` - the HTTP session
- `blmSession` - the BLM session
- `currentJsp` - the current JSP
- `jspHelper` - class that contains JSPF methods
- `errors` - the errors repository
- `results` - the results repository

Writing JSPs

When you code the display of your pages, do the following:

- For strings, use the `localize` function. This way, you can use the PLS to handle languages and localization of your application strings.
- For links or forms, use the `encodeURLfunc` or `getURL` functions in the `rendering_helper.inc` and use the PLS to create your page flows.

Debugging a Framework JSP

When working with JSPF JSPs, you can use the following to help you debug your JSPF JSPs:

- `JFN_DEBUG` application property to enable the debug mode. This mode displays information in the application server console.
- `JFN_DEBUG_INFO_DISPLAY` application property to display pageflow properties in the browser.
- `framework_appreset.jsp` to reload the plad design file without having to restart the application server
- `framework_reloadqra.jsp` to reload the report design file without having to restart the application server

Debugging Framework JSPs involves:

- Enabling the Debug Mode
- Displaying Information in the Browser
- Reloading Design Files

These debug utilities are for development purposes only and should never be used in production environments.

Enabling Debug Mode

When debugging an application built using the JSPF and the PLS, you need to have as much information as possible about the current JSP.

Enabling the Debug Mode involves:

- Setting the `JFN_DEBUG` application property

To enable the Debug Mode

- 1 Open the application in the PLS.
- 2 In the Explorer, expand the application to display the components.
- 3 Expand *Properties*. The properties appear.
- 4 Set the `JFN_DEBUG` property to `true`.

This property is case sensitive. Enter the value in lower case letters only.

- 5 Save your changes.

The application server console displays information about the functional steps and JSPF display JSPs

Displaying Information in the Browser

The `JFN_DEBUG_INFO_DISPLAY` application property to display pageflow properties in the browser. The information includes:

- The current functional step
- The main functional step
- The current display jsp page and form handler if it is the case
- The current Page Flow
- The redirect of the current functional step

Enabling the display of information in the browser involves:

- Setting the `JFN_DEBUG_INFO_DISPLAY` application property

To display JSPF information in the Browser

- 1 Open the application in the PLS.
- 2 In the Explorer, expand the application to display the components.
- 3 Expand *Properties*. The properties appear.
- 4 Make sure the `JFN_DEBUG` application property is set to `true`.
- 5 Set the `JFN_DEBUG_INFO_DISPLAY` property to `true`.

This property is case sensitive. Enter the value in lower case letters only.

- 6 Save your changes.

Reloading JSPF Configuration Information

When the application server starts, the design files are loaded into memory. Once loaded, changes made to this file are not taken into account. When developing and debugging applications, you often have to modify the contents of this file.

In order to keep you from having to stop and restart your application server every time you modify this file, you can use special JSPF JSPs:

- `framework_appreset.jsp`
- `framework_reloadqra.jsp`

These files are located in `<home_dir>/channels/common/fwk`.

These JSPs work only when the `JFN_DEBUG` property is set to `true`.

To reload configuration files

- 1 Make sure the `JFN_DEBUG` application property is set to `true`.
- 2 Run one of the following:
 - `framework_appreset.jsp` to reload the JSPF Framework configuration file
 - `framework_reloadqra.jsp` to reload the QRA configuration file

CHAPTER 7

Building Page Flows

In This Section

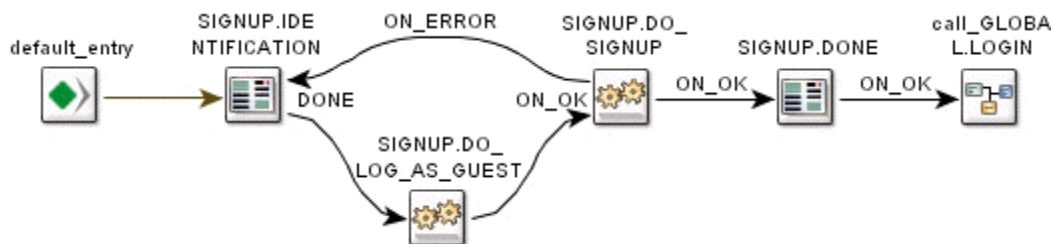
About Page Flows	84
About Page Flow Elements	87
Working With Page Flow References	88
Configuring Page Flows	90
Creating In Steps	93
Creating Out Steps	94
Working with Functional Steps	95
Connecting Steps	98

About Page Flows

A Page Flow is the foundation of your application. Page flows are sequences of events which carry out the functional tasks of your application. Similar to a flow chart, the page flow determines the path of users as they use the application.

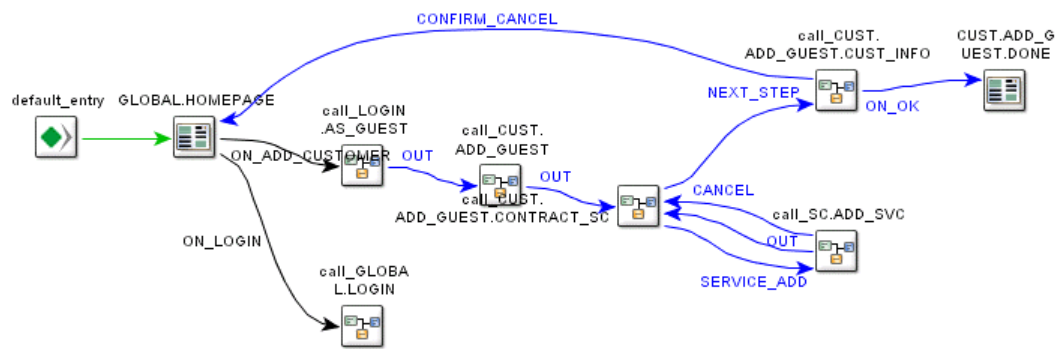
A page flow can be very simple and contain two or three simple steps and a linear flow. Page flows can also be complex, with many steps with the flow depending on several different factors.

A simple Page Flow look like this:



This page flow is the page flow for signing up for access to the application. In this page flow, you see the flow of the application start on the left in an entry point. The flow of the page flow then goes to a step called `SIGNUP.IDENTIFICATION`. The user interface code is in the JSP referenced by this step. When the user is finished, the flow then moves to processing steps. These steps process the information passed on from the first step and determine the next step based on the result. Depending on the result, either the `SIGNUP.IDENTIFICATION` step is called again whenever an error or an authentication problem occurs. If the results are processed without error, the page flow continues on to the next step, called `SIGNUP.DONE`. This page flow is complete at this step. From this point, the page flow calls another page flow called `GLOBAL.LOGIN`.

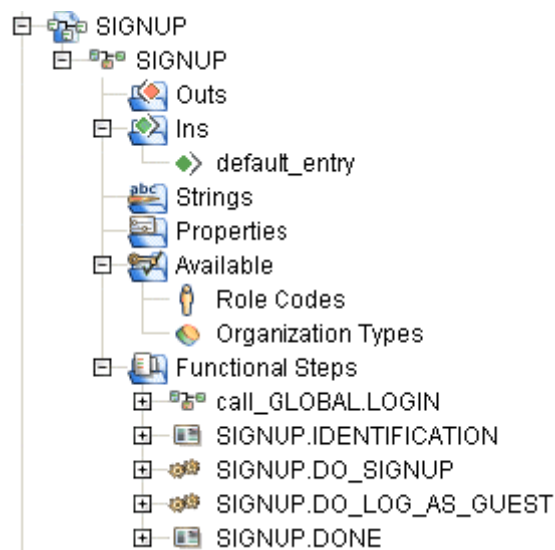
A more complex page flow looks like this:



This page flow illustrates the use of page flows as steps. This page flow is the starting page flow of an application. The `GLOBAL.HOMEPAGE` step is the home page of the application.

To illustrate how this type of page flow works, let's say the user is a guest user. At the `GLOBAL.HOMEPAGE` step, the page flow calls the `LOGIN.AS_GUEST` pageflow. When this pageflow is finished, it then calls the `CUST.ADD_GUEST` page flow. Again, the user goes through this page flow. When finished, this pageflow calls the `CUST.ADD_GUEST.CONTRACT.SC` pageflow. At this point, the page flow can call one of two different page flows. The user's choice determines which page flow is called. If the user wants to add a service to the contract they are signing up for, the page flow calls the `SC.ADD_SERVICE` page flow. This page flow is dedicated to adding services to a contract. This page flow can call one of two different page flows. The first call occurs when the user cancels the action. In this case, this page flow returns to the `CUST.ADD_GUEST.CONTRACT.SC` pageflow. If the user successfully adds a service to the contract, this page flow also returns the user by calling the the `CUST.ADD_GUEST.CONTRACT.SC` page flow.

In the explorer, this page flow looks like this:



Building Page Flows involves:

- Creating Page Flow References
- Configuring Page Flows
- Creating Out Steps
- Creating In Steps
- Creating Functional Steps
- Configuring Functional Steps
- Connecting Steps

About Page Flow Elements

Page flows have the following elements:

In Steps

In Steps are the start of a page flow.

Out Steps

Out Steps are the end of the page flow.

Properties

The Properties in a page flow are the properties of the page flow. They may override the Properties declared for the application.



Strings

The Strings in a page flow are strings which are associated with the page flow. They may override the Strings declared for the application.

Permissions

The page flow is available to users belonging to one of the specified groups.




The groups are:

-  Role Codes
-  Organization Types

Functional Steps

The Functional Steps are the individual steps in the data flow of a Page Flow.

The types of Functional Steps:

-  DisplayStep
This step handles the display of information
-  LogicStep
This step handles the processing of information.
-  PageflowStep
This step points to another page flow.

Working With Page Flow References

Working with Page Flows References involves:

- Creating Page Flow References
- Copying Page Flow References
- Renaming Page Flow References
- Editing Page Flows
- Refreshing Page Flows
- Closing Page Flow References
- Deleting Page Flow References

To create Page Flow References

- 1 Open the Application.
- 2 In the Explorer, select *Page Flows*.
- 3 Right click *Page Flows* then choose *Create New*. The *Choose File* dialog box appears.
- 4 Do one of the following:
 - Choose the page flow file of the reference
 - Enter the name of the file to create
- 5 Choose *OK*. The *Create New* dialog box appears.
- 6 Enter the code of the Page Flow Reference then choose *OK*. The new Page Flow Reference appears.

To copy Page Flow References

- 1 In the Explorer, expand *Page Flows*.
- 2 Select the Page Flow Reference to duplicate.
- 3 Right click the Page Flow then choose *Copy*.
- 4 Right click *Page Flows* then choose *Paste*. The Select File dialog box opens.
- 5 Enter the name of the new Page Flow Reference file then choose *OK*. The new Page Flow appears.

To rename Page Flow References

- 1 Open the application.
- 2 In the Explorer, expand *Page Flows*.

- 3 Select the Page Flow Reference to rename.
- 4 Slowly double-click the name. The name changes to a text field.
- 5 Enter the new name.
- 6 Press *Enter*. The new name appears.

To edit Page Flows

- 1 Open the application.
- 2 In the Explorer, expand *Page Flows*.
- 3 Expand the Page Flow Reference with the Page Flow to edit.
- 4 Double click the Page Flow. The page flow opens in the workspace.
- 5 Edit the Page Flow.
- 6 Save your changes.

To refresh the contents of Page Flows

When you modify elements in the Explorer, you may need to refresh the Page Flow.

- 1 On the Workspace, select the tab of the Page Flow to refresh. The Page Flow diagram appears in the Workspace.
- 2 Right click the tab then choose *Reload*. The Page Flow diagram in the Workspace refreshes and displays all of the components which currently make up the flow.

To delete Page Flow References

- 1 Open the application.
- 2 In the Browser, find the Page Flow to remove.
- 3 Right-click the Page Flow Reference then select *Delete*. A confirmation dialog box appears.
- 4 Choose one of the following:
 - Yes to remove the Page Flow Reference. This deletes only the reference and not the Page Flow file.
 - No to cancel

Configuring Page Flows

Configuring Page Flows involves:

- Specifying the Properties
- Specifying the Strings
- Specifying the Roles allowed to access the Page Flow
- Specifying the Organization Types allowed to access the Page Flow

To create Properties

- 1 In the Explorer, expand the Page Flow to display the components.
- 2 Right click Properties then select *Create New*. The *Create New* dialog box appears.
- 3 Enter the name of the property then choose OK. The property appears in the explorer.
- 4 Select the new property. The properties appear in the properties pane.
- 5 Enter the following properties:
 - Value
The value of the new property.
- 6 Save your changes.

To override Properties

- 1 Open the application.
- 2 In the Explorer, expand the Page Flows to display the components.
- 3 Find the Page Flow where you need to override the global property.
- 4 Right click Properties then select *Create New*. The *Create New* dialog box appears.
- 5 Enter the name of the property to override then choose OK. The property appears in the explorer.
- 6 Select the new property. The properties appear in the properties pane.
- 7 Enter the following properties:
 - Value
The value of the property to be used with this element instead of the global setting.
- 8 Save your changes.

To delete Properties

- 1 Open the application.
- 2 In the Explorer, expand Page Flows to display the components.
- 3 Expand Properties. The properties appear.
- 4 Right-click the property to delete then choose *Delete*. A confirmation dialog box appears.
- 5 Do one of the following:
 - Select *Yes* to permanently remove the element
 - Select *No* to cancel and return to the Explorer

To create Strings

- 1 In the Explorer, expand the Page Flow to display the components.
- 2 Right click Strings then select *Create New*. The *Create New* dialog box appears.
- 3 Enter the name of the string then choose OK. The string appears in the explorer.
- 4 Select the new string. The properties appear in the properties pane.
- 5 Enter the following properties:
 - Value
The value of the string.
- 6 Save your changes.

To override Strings

- 1 Open the application.
- 2 In the Explorer, expand Page Flows to display the components.
- 3 Find the Page Flow where you need to override the global String.
- 4 Right click Strings then select *Create New*. The *Create New* dialog box appears.
- 5 Enter the name of the String to override then choose OK. The property appears in the explorer.
- 6 Select the new String. The properties appears in the properties pane.
- 7 Enter the following properties:
 - Value
The value of the String to be used with this element instead of the global setting.
- 8 Save your changes.

To delete Strings

- 1 Open the application.
- 2 In the Explorer, expand Page Flows to display the components.
- 3 Expand Strings. The strings appear.
- 4 Right-click the string to delete then choose *Delete*. A confirmation dialog box appears.
- 5 Do one of the following:
 - Select *Yes* to permanently remove the element
 - Select *No* to cancel and return to the Explorer

To specify Role Codes

- 1 In the Explorer, expand the Page Flow to display the components.
- 2 Expand the Permissions element. The Role Codes and Organization Types elements appear.
- 3 To assign access based on Roles:
- 4 Click Role Codes. The properties pane displays the roles having permission to access the Page Flow.
- 5 Click Add. A new value appears in the properties pane.
- 6 Select an available Role Code.
- 7 Repeat as required.
- 8 Save your changes.

To specify Organization Types

- 1 In the Explorer, expand the Page Flow to display the components.
- 2 Expand the Permissions element. The Role Codes and Organization Types elements appear.
- 3 Click Organization Types. The properties pane displays the Organization Types having permission to access the Page Flow.
- 4 Click Add. A new value appears in the properties pane.
- 5 Select an available Organization Type.
- 6 Repeat as required.
- 7 Save your changes.

Creating In Steps

To create In steps

- 1 In the Explorer, expand the Page Flow to display the components.
- 2 Right click Ins then select *Create New*. The *Create New* dialog box appears.
- 3 Enter the code of the In step then choose OK. The step appears in the explorer.
- 4 Select the new In step. The properties appear in the properties pane.
- 5 Enter the following properties:
 - Functional Step
The name of the functional step which starts the page flow.
- 6 Save your changes.

Creating Out Steps

To create Out steps

- 1 In the Explorer, expand the Page Flow to display the components.
- 2 Right click Outs then select *Create New*. The *Create New* dialog box appears.
- 3 Enter the code of the Out step then choose OK. The step appears in the explorer.
- 4 Select the new Out step. The properties appear in the properties pane.
- 5 Enter the following properties:
 - Functional Step
The name of the default functional step called when the Page Flow has not defined a redirection of this Out step.
- 6 Save your changes.

Working with Functional Steps

To create LogicSteps

- 1 In the Explorer, expand the Page Flow to display the components.
- 2 Right click Functional Steps then select *Create New > LogicStep*. The *Create New* dialog box appears.
- 3 Enter the code of the LogicStep then choose OK. The Logic Step appears in the explorer.
- 4 Select the LogicStep. The properties appear in the property pane.
- 5 Enter the following properties:
 - JSP
Select one of the available JSPs.
 - Logic
Enter the name of the Logic Handler to use.
- 6 Save your changes.

To create DisplaySteps

- 1 In the Explorer, expand the Page Flow to display the components.
- 2 Right click Functional Steps then select *Create New > DisplayStep*. The *Create New* dialog box appears.
- 3 Enter the code of the DisplayStep then choose OK. The Display Step appears in the explorer.
- 4 Select the DisplayStep. The properties appear in the property pane.
- 5 Enter the following properties:
 - JSP
Select one of the available JSPs.
 - Form
Enter the name of the Form Handler to use.
- 6 Save your changes.

To create PageFlowSteps

- 1 In the Explorer, expand the Page Flow to display the components.

- 2 Right click Functional Steps then select *Create New > PageFlowStep*. The *Create New* dialog box appears.
- 3 Enter the code of the PageFlowStep then choose OK. The Display Step appears in the explorer.
- 4 Select the PageFlowStep. The properties appear in the property pane.
- 5 Enter the following properties:
 - PageFlow
Select one of the available Page Flows.
 - Entry
Enter the name of the Page Flow In step to use.
- 6 Save your changes.

Configuring Functional Steps

Configuring Functional Steps involves:

- Specifying the Properties
- Specifying the Strings
- Specifying the Role Code
- Specifying the Organization Type

To create Strings

To create Properties

To specify Role Codes

To specify Organization Types

Connecting Steps

Connecting Steps involves:

- Creating override connections
- Creating redirect connections
- Deleting connections

To connect steps

- 1 Open the Page Flow in the Workspace.
- 2 Find the Step you want to connect.
- 3 Click the Step then drag the pointer to the destination Step. Depending on the type of destination Step, the following may occur:
 - An arrow appears connecting the two Steps.
 - A dialog box appears informing you that you cannot connect the Steps. This occurs when:
 - The Step is already connected
 - The Steps are not compatible

To disconnect steps

- 1 Open the Page Flow in the Workspace.
- 2 Find the step you want to disconnect.
- 3 Select the arrow connecting then press *Del*. The arrow disappears and a message appears in the message pane.

CHAPTER 8

Building Menus

In This Section

About Menus	100
Working with Menus	101
Configuring Menus	103
Specifying Display Order	104

About Menus

A Menu is a list of choices presented to the user.

You use the PLS to create and configure the layout of your menus.

Working with Menus

To create menus

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Right click *Menus* then select *Create New*. The *Create New* dialog box appears.
- 4 Enter the code of the menu then choose OK. The menu appears in the explorer.
- 5 Select the new menu. The properties appear in the properties pane.
- 6 Enter the following properties:
 - Name
The name of the menu. This name is the default display value and is displayed in the menu bar.
 - Description
A description of the menu.
 - Pageflow
The page flow associated with this menu.
 - In
The In step of the specified page flow
- 7 Save your changes.

To create submenus

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Under menus, expand the menu to modify.
- 4 Right click the menu then select *Create New*. The *Create New* dialog box appears.
- 5 Enter the code of the submenu then choose OK. The menu appears in the explorer.
- 6 Select the new submenu. The properties appear in the properties pane.
- 7 Enter the following properties:
 - Name
The name of the submenu. This name is the default display value.
 - Description
A description of the submenu.
 - Pageflow

The page flow associated with this submenu.

- In

The In step of the specified page flow

8 Save your changes.

Configuring Menus

Configuring menus involves:

- Specifying the Properties
- Specifying the Strings

Specifying Display Order

The order of menu items in the Explorer corresponds to how your application displays menus.

Specifying the display order involves:

- Specifying the display order of menu items

To specify the display order

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Expand *Menus*. The list of menus appears.
- 4 Open the menu to modify.
- 5 Drag and drop the menus until they appear in the correct order.
- 6 If required, modify the submenu items.
- 7 Save your changes.

CHAPTER 9

Localizing Applications

In This Section

About Localizing Applications	106
Working with Strings	107
Working with Languages	109
Localizing Strings	111

About Localizing Applications

The strings you declare in the PLS are strings which are used by various elements of the presentation layer application. These elements include menus, steps, and page flows.

The PLS lets you centralize the presentation logic strings in a single location. The tool also allows you to configure strings for specific components or override the global strings when certain conditions are met.

By default, the Strings you enter in the PLS are the strings corresponding to the default language declared in the CID. If your application has more than one available language, the PLS generates lists of strings for each declared language and automatically configures the application to use these lists to determine the value of the strings.

Localizing Applications involves:

- Declaring strings
- Declaring languages
- Localizing strings

Working with Strings

You can create global strings which remain the same throughout the application. You can also create strings associated with the following elements:

- Page Flows
- Functional Steps
- Menus
- JSPs

You can also override the global strings in these elements.

Working with strings involves:

- Creating Strings
- Removing Strings
- Overriding Strings

To create Strings

- 1 Open the application.
- 2 In the Explorer, do one of the following:
 - To declare a global string, right click *Strings* then select *Create New*. The *Create New* dialog box appears.
 - To declare a String for an element, find the element then right click *Strings* then select *Create New*. The *Create New* dialog box appears.
- 3 Enter the code of the string then choose *OK*. The string appears in the explorer.
- 4 Select the new string. The properties appear in the properties pane.
- 5 Enter the following properties:
 - Value
The value of the string.
- 6 Save your changes.

To override Strings

- 1 Open the application.
- 2 Find the element where you need to override the global String.
- 3 Right click *Strings* then select *Create New*. The *Create New* dialog box appears.
- 4 Enter the name of the global String to override then choose *OK*. The property appears in the explorer.
- 5 Select the new String. The properties appear in the properties pane.

6 Enter the following properties:

- Value

The value of the String to be used with this element instead of the global setting.

7 Save your changes.

To delete Strings

1 Open the application.

2 In the Explorer, locate the String to delete.

3 Right-click the string to delete then choose *Delete*. A confirmation dialog box appears.

4 Do one of the following:

- Select *Yes* to delete the string
- Select *No* to cancel and return to the Explorer

Working with Languages

The Languages declared in the PLS correspond to the application languages declared in the CID.

These languages are for reference only and must correspond to the languages configured and declared in the CID.

When you localize your application, the PLS uses these languages to create the files required to localize the strings.

Working with languages involves:

- Creating languages
- Removing languages

To add Languages

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Right click *Languages* then select *Create New*. The *Create New* dialog box appears.
- 4 Enter the code of the Language then choose *OK*. The language appears in the explorer.
- 5 Select the new Language. The properties appear in the properties pane.
- 6 Enter the following properties:
 - Name
The name of the language.
 - Description
A description of the language.
- 7 Save your changes.

To remove Languages

- 1 Open the application.
- 2 In the Explorer, expand the application to display the components.
- 3 Expand *Languages*. The language appears.
- 4 Right-click the language to delete then choose *Delete*. A confirmation dialog box appears.
- 5 Do one of the following:

- Select *Yes* to delete the element
- Select *No* to cancel and return to the Explorer

Localizing Strings

The PLS makes localizing the strings in your application very easy.

If your application has more than one available language, the PLS generates lists of strings for each declared language and automatically configures the application to use these lists to determine the value of the strings.

The files are located in `<home_dir>\channels\<application_name>\WEB-INF\classes\mycfg\jfn\strings`

The PLS generates the following files:

- `<application_name>_<language>.properties` file for the global strings
- `/jsp/<element_name>_<language>.properties` file for each JSP having declared strings
- `/pageFlows/<element_name>_<language>.properties` file for each page flow having declared strings
- a `.diff` file for each file that has changed since the last generation. This file contains the differences made to the strings since the previous generation for the default language.

The value of the strings in the `.properties` files have the format:

`<element_name>.<component_name>.<string_name>=<value>`

An example of the string declaration:

```
APP.STR.subscription_fee_text=Signup Fee
APP.STR.card_number_label=Card number:
APP.MENU.INFO.Name=User Information
```

To generate the String files

- 1 Choose *Application > Generate String Bundles*. When finished a message appears in the message pane.

The files are located in `<home_dir>\channels\<application_name>\WEB-INF\classes\mycfg\jfn\strings`

To localize the strings

- 1** Go to `<home_dir>\channels\<application_name>\WEB-INF\classes\nmycfg\jfn\strings`
- 2** If present, open and read the diff file listing the changes since the last generation.
- 3** Open the file corresponding to the element and language to localize.
- 4** Change the value of the string.
- 5** Save your changes.

Index

A

Application Properties
 about • 57
 complex parameter settings • 59
 creating • 60
 JSPF settings • 58
 overriding • 61
 removing • 60
 report processing settings • 60
 search settings • 59
 setting • 60

Applications
 closing • 55
 creating • 55
 generating documentation • 56
 opening • 55
 overview • 52
 saving • 55
 validating • 56

C

Channels
 about • 38
 adding • 62
 contents • 37
 menus • 37
 Web channel • 38, 40
 CID (Customer Interaction Datastore)
 adding sample data • 46
 removing sample data • 46
 cidAdminTool Administration Tool
 create_demo_cid_test command • 46
 drop_demo_cid_structure command • 46

D

Display Pages
 about • 33
 and the JSPF • 32, 33

F

Form Handler JSPs

about • 34
framework_head.inc Framework JSP
 about • 32
 location • 32
 using • 33
framework_menuend.inc Framework JSP
 about • 32
 location • 32
framework_menustart.inc Framework JSP
 about • 32
 location • 32
framework_start.jsp Framework JSP
 about • 32
 location • 32
 using • 33
framework_tail.jsp Framework JSP
 about • 32
 location • 32
 Functional Steps
 about • 87
 connecting • 98
 creating PageFlowSteps • 95
 DisplaySteps • 95
 PF - creating LogicSteps • 95
 specifying Organization Types • 97
 specifying Properties • 97
 specifying Role Codes • 97
 specifying Strings • 97

I

In Steps
 connecting • 98
 creating • 93

J

jfnApplication.properties Configuration
 File
 about • 28, 29
 and channels • 62
 location • 29

using • 29, 62

JSP

about • 70

adding to JSPF • 74, 75, 76, 77, 78

configuring references • 72

creating references • 71

debugging • 80

declaring in JSPF Framework

configuration file • 75

editing • 72

framework JSPs • 32, 33

home directory of • 71

JSP - mandatory includes • 76, 77

JSPF variables • 75

removing references • 72

specifying page • 71

JSPF Classes

about • 31

jar file • 31

JSPF Framework

about • 26

channels • 22, 38

components • 28

display pages • 33

features • 27

form handler JSPs • 34

Framework JSPs • 32, 33

location • 28

logic handler JSPs • 35

mandatory includes • 76

reloading configuration file • 82

variables • 75

L

Languages

adding • 109

and strings • 106

Localizing

about • 106

and languages • 109

creating strings • 107

generating strings • 111

localizing • 112

Logic Handler JSPs

about • 35

M

Media

adding • 62

and channels • 62

Menus

about • 37, 100

contents • 37

creating • 101

creating sub menus • 101

display order • 104

MyWeb Channel

display options • 40, 41, 42, 43

features • 38

O

Organization Type

about • 64

creating • 65

in Functional Steps • 97

in JSPs • 72

in Page Flows • 92

specifying • 67

Out Steps

connecting • 98

creating • 94

Overrides

about • 98

Overriding

application properties • 61

security • 64

strings • 107

P

Page Flow References

copying • 88

creating • 88

deleting • 89

editing page flows • 89

refreshing • 89

renaming • 88

Page Flows

about • 84

and properties • 90

components of • 87

configuring • 90

copying • 88

creating • 88

deleting • 89

In steps • 93

Out steps • 94

Properties • 90

refreshing • 89

renaming • 88

specifying Organization Types • 92

- specifying Role Codes • 92

- specifying Strings • 91

- Permissions

- about • 64

- and Organization Types • 64

- and User Roles • 64

- assigning • 67

- creating • 65

- Personalization Manager

- about • 22

- channels • 38

- components • 22

- sample data • 44, 46

- PLS Design File

- reloading • 82

S

- Sample Data

- about • 44, 46

- contents • 44, 45

- inserting into CID • 46

- reference • 44

- removing from CID • 46

- Sample Data - services • 44

- users • 45

- Steps

- about • 87

- Connecting • 98

- Functional Steps • 87

- In steps • 87

- Out Steps • 87

- Strings

- about • 107

- and languages • 109

- creating • 107

- generating** • 111

- localizing • 111

- overriding • 107

U

- User Roles

- about • 64

- creating • 65

- in Functional Steps • 97

- in JSPs • 72

- in LogicSteps • 92

- specifying • 67