



IBM Case Manager 5.2: Customize and Extend the Features

Course Code: F217

ERC: 2.0

Volume Number: 1/1

Participant Guide

Template Version#: 11.0/1.0

Trademarks

IBM® and the IBM logo are registered trademarks of International Business Machines Corporation.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

DB2®
System p™

HACMP™
System x™

System i™
System z™

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Other company, product, or service names may be trademarks or service marks of others.

September 2014 edition

Trademarks

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as is" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

© Copyright International Business Machines Corporation 2014.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

TABLE OF CONTENTS

Course Overview

■ Course Description	2-2
■ Agenda	2-4
■ Options for customizing IBM Case Manager Client	2-5

Unit 1: Customize the Case Manager Client User Interface

Lesson 1.1: Customize the banner and login page

■ Customize the Case Manager Client appearance	1-3
■ Customize the Banner appearance	1-4
■ Customize the login page	1-5
■ Exercise 1.1.1:Customize the banner	1-7
■ Exercise 1.1.2:Customize the login page	1-12

Lesson 1.2: Configure icons and labels

■ Change Icons for the client	1-16
■ Modify the labels in the Case Manager Client	1-18
■ Exercise 1.2.1:Associate a custom icon for a MIME type	1-19
■ Exercise 1.2.2:Modify labels in the IBM Case Manager Client	1-24

Lesson 1.3: Specify viewers for file types

■ Viewer Maps	1-28
■ Exercise 1.3.1:Create a Viewer Map for PDF files	1-29
■ Exercise 1.3.2:View the Microsoft Word documents in the FileNet Viewer	1-35

Lesson 1.4: Create and customize help topics

■ Add custom help topics	1-39
■ Creating a help plug-in	1-40
■ Exercise 1.4.1>Edit the existing help topics	1-41
■ Exercise 1.4.2>Create a help plug-in	1-43

Lesson 1.5: Customize toolbar and menu

■ Toolbar widgets	1-49
■ Exercise 1.5.1:Customize the toolbar to implement actions	1-51
■ Exercise 1.5.2:Add a custom action as a menu item	1-56

Unit 2: Use Scripts to Customize Case Manager Client

Lesson 2.1: IBM Case Manager JavaScript API Overview

■ IBM Case Manager Development Architecture	2-3
■ IBM Case Manager API toolkits	2-4
■ IBM Content Navigator APIs	2-5
■ IBM Case Manager JavaScript API	2-7

■ Collaborative editing of objects	2-11
■ Script Adapter widget	2-13
■ Exercise 2.1.1:Use Script Adapter to customize the Case client	2-15

Lesson 2.2: Create a script action to start a task

■ Adding discretionary tasks	2-23
■ Tips for the labs	2-24
■ IBM Case Manager classes and events that are used in this lesson	2-25
■ Exercise 2.2.1:Create a toolbar button to start a task	2-26

Lesson 2.3: Add a choice list to a case property

■ Adding a choice list to a case property	2-33
■ IBM Case Manager classes that are used in this lesson	2-34
■ Exercise 2.3.1:Dynamically add a choice list to a case property	2-35

Lesson 2.4: Validate the data based on a step response

■ Data Validation that is based on a step response	2-43
■ Exercise 2.4.1:Validate the data based on a step response	2-44

Lesson 2.5: Create a case custom workbench page

■ Case custom workbench pages	2-53
■ Widgets that are used for this lesson	2-55
■ Filter In-basket SA - Script Adapter	2-56
■ Filter Search SA - Script Adapter	2-58
■ Lab overview	2-61
■ Exercise 2.5.1:Configure your system for the workbench page	2-62
■ Exercise 2.5.2:Create a case custom workbench page	2-66
■ Exercise 2.5.3:Add a Script Adapter to filter In-baskets	2-68
■ Exercise 2.5.4:Add a Script Adapter to filter cases	2-74

Unit 3: Develop Custom Widgets

Lesson 3.1: Custom widget development overview

■ Developing case management applications	3-3
■ Creating a custom page widget and actions package	3-5
■ IBM Content Navigator plug-in for the custom widget package	3-7
■ Setting up the development environment for plug-ins	3-9
■ Exercise 3.1.1:Create a plug-in project in Eclipse	3-10

Lesson 3.2: Create catalog and widget definition files

■ Create registry files for the custom widget package	3-18
■ Catalog JSON file	3-19
■ Page widget definition JSON file	3-21
■ Exercise 3.2.1>Create catalog and widget definition JSON files	3-22

Lesson 3.3: Implement a custom widget

■ IBM Case Manager custom page widget development	3-29
■ Implement a page widget	3-31
■ Files that are used in this lesson	3-33
■ Exercise 3.3.1:Implement a custom widget	3-34

Lesson 3.4: Build and register a widget package

■ Register the custom widget package	3-43
■ Exercise 3.4.1:Build and register the widget package	3-44
■ Exercise 3.4.2:Test the custom widget	3-50
■ Exercise 3.4.3:Troubleshooting (if the widget does not work)	3-54

Lesson 3.5: Create a widget with a toolbar and a menu

■ Folder structure for the custom widget project in this lesson	3-58
■ Implement toolbar and menu for your widget	3-59
■ Exercise 3.5.1>Create a Java project in Eclipse	3-60
■ Exercise 3.5.2>Create the widget definition JSON file	3-63
■ Exercise 3.5.3:Implement your widget	3-68

Lesson 3.6: Build and deploy a widget as an EAR file

■ Widget package structure	3-77
■ Exercise 3.6.1:Build and deploy the widget package	3-78
■ Exercise 3.6.2:Troubleshooting	3-88

Lesson 3.7: Update an existing widget package

■ Custom case comment widget	3-90
■ Update a custom widget package	3-91
■ Exercise 3.7.1:Update an existing package with new widgets	3-92
■ Appendix: Uninstall a custom widget in IBM Case Manager	3-99

Unit 4: Implement External Data Services

Lesson 4.1: Implement sample external data service

■ Case data from an external data source	4-3
■ Sample external data service	4-5
■ Java code in the sample EDS service	4-6
■ External data configuration in the sample	4-7
■ Exercise 4.1.1:Check the sample EDS configuration	4-8
■ Exercise 4.1.2:Change the field status dynamically for a property	4-10
■ Exercise 4.1.3:Add an external choice list with EDS	4-14
■ Exercise 4.1.4>Create dependent choice lists with EDS	4-16
■ Exercise 4.1.5:Implement case property validation	4-19
■ Exercise 4.1.6:Prefill an initial value for a case property	4-21
■ Exercise 4.1.7:Set the case property field status as required	4-23
■ Appendix: Steps to set up the sample External Data Service	4-25

Unit 5: Appendix

Appendix A: Start and Stop System Components

Appendix B: Debugging and troubleshooting

Appendix C: Additional information and links

Course Wrap Up

- [Completed Objectives](#) [W-2](#)
- [Course Evaluation](#) [W-4](#)

Course Overview

This Course Overview provides you with an outline of the course, its objectives, agenda, course organization and other details specific to this course.

Course Description

Overview

This course describes various options that IBM Case Manager provides for developing the user experience for your application.

Course Objectives

After completing this course, you should be able to:

- **Customize the Case Manager Client User Interface**
 - Customize the Banner and Login Page
 - Associate a custom icon for a MIME type
 - Modify labels in the IBM Case Manager Client
 - Specify Viewers for file types
 - View the Microsoft Word documents in the FileNet Viewer
 - Create and Customize Help topics
 - Customize the toolbar to implement actions
 - Add a custom action as a menu item
- **Use Scripts to Customize Case Manager Client**
 - Use Script Adapter to customize the Case client
 - Create a toolbar button to start a task
 - Dynamically add a choice list to a case property
 - Validate the data based on a step response
 - Create a case custom workbench page
 - Add a Script Adapter to filter In-baskets
 - Add a Script Adapter to filter cases
- **Develop Custom Widgets**
 - Create an IBM Content Navigator plug-in project in Eclipse
 - Create catalog and widget definition JSON files
 - Implement a custom widget (Custom search)
 - Build and register the widget package
 - Test the custom widget
 - Create a Java project in Eclipse for a widget package
 - Implement toolbar and menu for your widget
 - Define widget properties, and add event handling for your widget
 - Build and deploy the widget package with an EAR file
 - Troubleshooting tips

- Update an existing widgets package with new widgets
(Custom case comment widget as an example)
- Uninstall a custom widget package in IBM Case Manager
- **Implement External Data Services (EDS)**
 - Check the sample EDS configuration
 - Change the field status dynamically for a property
 - Add an external choice list with EDS
 - Create dependent choice lists with EDS
 - Implement case property validation
 - Prefill an initial value for a case property
 - Set the case property field status as required
 - Appendix: Steps to set up the sample External Data Service

Audience

The intended audiences for this course are:

- Developers who are responsible for:
 - Customizing and extending the IBM Case Manager features by developing widgets.
 - Building a customized user interface for IBM Case Manager.
- Anyone who needs to know the IBM Case Manager customization.

Prerequisites

The prerequisites for taking this course are:

- Intermediate level expertise in the following technologies:
 - Java
 - Dojo
 - JavaScript
 - JSON
 - HyperText Markup Language (HTML 5)
 - Cascading Style Sheets (CSS3)
- Experience with Eclipse IDE for developing applications.
- Familiarity with deploying applications in WebSphere Application Server.
- Familiarity with Case Management concepts
- Recommended prerequisite courses:
 - F215 - IBM Case Manager 5.2 Introduction to Case Manager
 - F212 - IBM Case Manager 5.2: Build a Case Manager Solution
 - F120 - IBM Content Navigator 2.0.2: Customize and Extend the Features

Agenda

Duration

The duration of this course is 3 days.

Day 1

- Unit 0: Course Overview – Options for customizing IBM Case Manager Client
- Unit 1: Customize the Case Manager Client User Interface (lessons 1-5)
- Unit 2: Use Scripts to Customize Case Manager Client (lesson 1)

Day 2

- Unit 2: Use Scripts to Customize Case Manager Client (lessons 2-5)
- Unit 3: Develop Custom Widgets (lessons 1-2)

Day 3

- Unit 3: Develop Custom Widgets (lessons 3- 7)
 - Unit 4: Implement External Data Services (EDS) (lesson 1)
-

Options for customizing IBM Case Manager Client

Options

IBM Case Manager provides various options to develop the user experience for your application. Each option requires different development skill set and time to achieve the goal.

- Customize the Case Manager Client User Interface
- Use Scripts and Script Adapter widget to customize Case Manager Client
- Develop Custom widgets
- Implement the External Data Service (EDS) interface

The following sections describe each option. It also specifies the unit names in this course that provides information and lab exercises for each option.

Customize the Case Manager Client User Interface (Basic, no code required)

You can change the appearance of the Case Manager Client in the admin tool without writing any code. This option is helpful to make quick changes to comply with the visual standards of an organization.

- Configure visual aspects for the Case Manager Client banner and login page.
 - Specify the name for your solution.
 - Add a company logo to both the login page and the banner.
 - Alter the colors that are used in the banner.
- Configure visual aspects for the icons and labels:
 - Change icons for specific MIME types.
 - Modify the labels that are used in the desktop.
- Specify Viewers for file types.
- View the Microsoft Word documents in the FileNet Viewer.
- Add custom help topics.
- Extend the Toolbar and Menus.



Note

For more information, see “Unit 1- Customize the Case Manager Client User Interface” of this course.

Use Scripts to Customize Case Manager Client (Intermediate)

You can use the IBM Case Manager JavaScript model API in scripts and Script Adapter widget to customize the Case Manager Client.

The Script Adapter widget is a special widget that is used for simple customization of page behavior.

Following examples are used in this course:

- Document enumeration
- Create a toolbar button to start a task
- Dynamically add a choice list to a case property
- Validate the data based on a step response
- Create a case custom workbench page
 - Add a Script Adapter to filter In-baskets
 - Add a Script Adapter to filter cases



Note

For more information, see “Unit 2 - Use Scripts to Customize Case Manager Client” of this course.

Develop Custom Widgets (Advanced)

IBM Case Manager and IBM Content Navigator plug-in enable developers to add new functions, change existing behavior and appearance of the application, or create an application. It is a powerful mechanism and the most flexible option for customizing user experience within your solution.

A custom widget is required when you want to present information in a page or when your widget must react to page events.

Developing a custom widget involves the following high-level steps:

- Create an IBM Content Navigator plug-in
- Create catalog and widget definition JSON files for registering the widget.
- Implement custom widgets
 - Implement functions, toolbar, and menu items for your widget
 - Define widget properties, and add event handling for your widget
- Build, deploy, and register the widget package
- Test the custom widget.
- You can also update an existing widgets package with new widgets

Required skills

- Java
- Dojo
- JavaScript
- JSON
- HyperText Markup Language (HTML 5)
- Cascading Style Sheets (CSS3)



Note

For more information, see “Unit 3 - Develop Custom Widgets” of this course.

Implement the External Data Services (Intermediate)

The IBM FileNet P8 repositories that are used in your solution have specific options for how properties are defined in your object model.

External Data Services (EDS) allows you to change the behavior of property data fields dynamically during run time in a light-weight manner.

IBM Case Manager provided sample EDS implementation enables the following tasks:

- Change the field status (hide or show) dynamically for a property
- Add an external choice list with EDS
- Create dependent choice lists with EDS
- Implement case property validation
- Prefill an initial value for a case property
- Set the case property field status as required



Note

For more information, see “Unit 4 - Implement External Data Services” of this course.

Customize the Case Manager Client User Interface

This unit provides guidance for customizing the IBM Case Manager Client user interface in the admin tool without having to write any code.

LESSON 1.1: Customize the banner and login page

What this lesson is about?

This lesson describes how to customize the Case Manager Client appearance for the banner and login page.

What you should be able to do?

After completing this lesson, you should be able to:

- Customize the banner and logon page.

How you will check your progress?

- Hands on labs.

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Customize the Case Manager Client appearance

IBM Case Manager Client is based on IBM Content Navigator. Its visual characteristics are shown with a customized theme.

- The theme determines the overall appearance, such as fonts and colors.
 - You can make the following changes in the administration tool without changing the theme:
 - Add your company logo to both the login page and banner of the client.
 - Add useful notes to the login page.
 - Alter the color of the desktop banner.
 - Change icons.
 - Alter the labels that are used in the client.
-

Customize the Banner appearance

Banner configuration

You can customize the banner without any custom code in the administration tool.

Desktop: Custom CM Client

The screenshot shows the 'Appearance' tab selected in the navigation bar. Below it, a section titled 'Banner and login page' contains the following fields:

- Application name:** Custom CM Client
- Banner logo:** URL: EDUIImages/IBM_Nav_logo.png (with a note: "Enter the URL of the logo image file.")
- Banner background color:** Custom: #FFFFCC (with a note: "Enter the color as a 3- or 6-hexadecimal color.")
- Application name text color:** Custom: #006600 (with a note: "Enter the color as a 3- or 6-hexadecimal color.")
- Banner icon color:** Dark (with a note: "Select a color for the icon based on the background color")

URL for logos

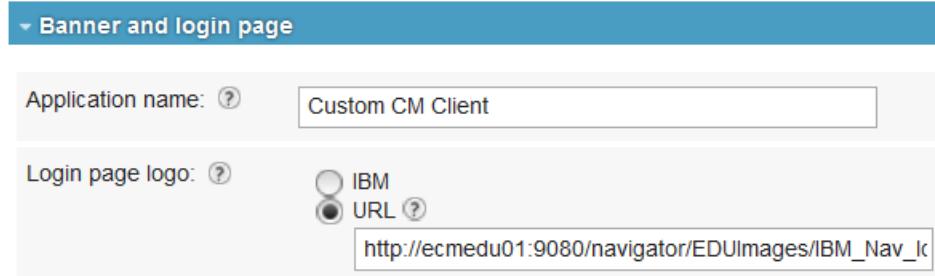
- A URL references the New logo.
 - If the image file is in the IBM Content Navigator web client application, enter a relative path.
 - Example: EDUIImages/IBM_Nav_logo.png.
 - You can also create a web application and add your logos to the Web Content folder.
 - If you deploy it in a highly available configuration, enter a fully qualified path that is available to all instances of IBM Content Navigator.
Example: `http://<server_name>/<application name> /<file name>`
- The logo must be no larger than 72 pixels wide and 32 pixels high.
 - The IBM banner logo in IBM Content Navigator is 72 x 24 pixels.
- The logo image must have a transparent background.

Customize the login page

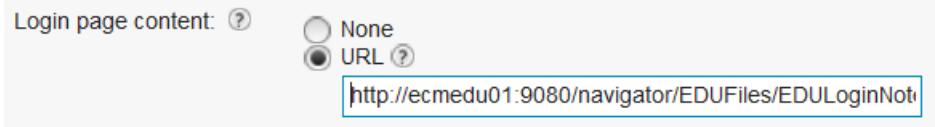
Login page configuration

You can customize the Login page logo and login page content in the administration tool.

- Enter the URL for the logo file.



- Enter the URL for the HTML file for the login page notes.



URL for login logo and login page content

- A URL references the new logo or the content file.
 - If the file is in the Content Navigator web client application, enter a relative path.
 - You can also create a web application and add your files to the Web Content folder.
- The size of the logo can be 150 x 112 pixels.
- The logo image must have a transparent background.

Login page content

- You can add content to the login page to provide information or the latest news to users when they access IBM Case Manager Case client.
- Example of information:
 - Guidance about the use of the system
 - Help desk contact procedures
 - Any planned downtime for system maintenance

- To add login page content, prepare an HTML file that contains the information that you want.
 - Add the file to a web application that is available to all systems where IBM Content Navigator is installed.
-

Exercise 1.1.1: Customize the banner

Introduction

In this lab, you make a copy of the existing Case Manager Case client and save it as another application. You customize the banner for your application.

User accounts

Type	User ID	Password
Operating system	administrator	passw0rd
IBM Case Manager Administrator	P8Admin	IBMFileNetP8

Passwords are always case-sensitive.

Procedures

Procedure 1: Start WebSphere Application Server, page. 1-7

Procedure 2: Make a copy of the existing Case Manager Client, page. 1-7

Procedure 3: Edit the new client to customize the banner, page. 1-9

Procedure 1: Start WebSphere Application Server

1. Click Start > All Programs > IBM WebSphere > IBM WebSphere Application Server V8.5 > Profiles > AppSrv01 > Start the server.
 - You can also use the Start Server1.bat file in the WebSphere Admin folder on the desktop.
2. Wait for the Start the server page to close.



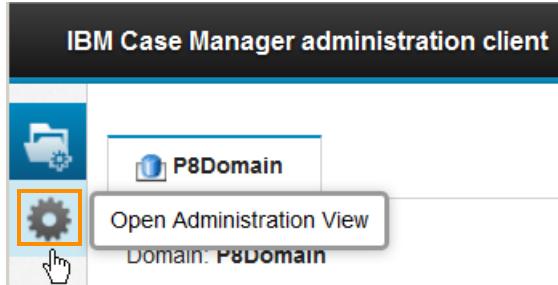
Note

For more information about “Start and stop System Components”, see the Appendix at the end of book.

Procedure 2: Make a copy of the existing Case Manager Client

1. Start the IBM Case Manager administration client.
 - URL: <http://ecmedu01:9080/navigator/?desktop=icmadmin>
 - User name: P8admin
 - Password: IBMFileNetP8

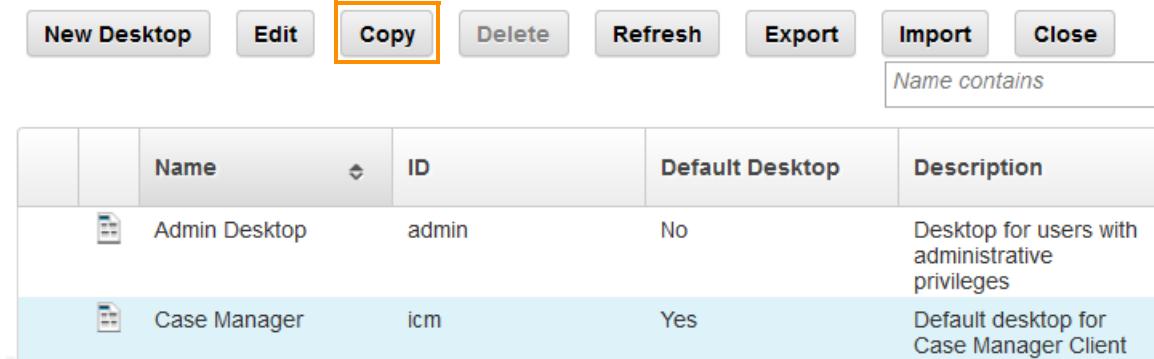
2. Copy the Case Manager Client application.
 - a. On the administration client, click the “Open Administration View” icon. If the Administration View icon is not visible, refresh the browser.



 Troubleshooting

You can also access the administration page with this URL:
<http://ecmedu01:9080/navigator/?desktop=admin>

- b. The Desktops tab opens. In the Desktops tab, select Case Manager and click Copy.



A screenshot of the "Desktops" tab in the administration client. At the top, there's a toolbar with buttons: New Desktop, Edit, Copy (which is highlighted with an orange border), Delete, Refresh, Export, Import, and Close. Below the toolbar is a search bar labeled "Name contains". The main area is a table with columns: Name, ID, Default Desktop, and Description. There are two rows: "Admin Desktop" (ID: admin, Default Desktop: No) and "Case Manager" (ID: icm, Default Desktop: Yes). The "Case Manager" row is currently selected.

	Name	ID	Default Desktop	Description
	Admin Desktop	admin	No	Desktop for users with administrative privileges
	Case Manager	icm	Yes	Default desktop for Case Manager Client

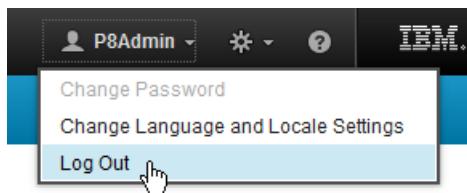
3. In the New Desktop tab, enter a name for the desktop (Example: Custom CM Client).
 - a. The ID field is automatically populated. Edit the ID value to custom.

 Important

The ID value is case-sensitive, and it must be unique. You reference this ID in the URL for this application.

4. Click the Appearance subtab.
 - a. Edit the Application name to Custom CM Client.
5. Click Save and Close.
6. Verify that the new application is listed in the Desktops tab.

7. Optional verification: Start the custom client application that you created.
 - a. In a new browser tab, go to <http://ecmedu01:9080/navigator/?desktop=custom>
 - b. Enter the login credentials.
 - User name: P8admin
 - Password: IBMFileNetP8
- The custom client application opens and the banner shows the new name.
8. Observe that the custom client application is identical to the default client and has the same Cases and Work tabs.
9. Log out of the custom client application and close the browser tab.



Procedure 3: Edit the new client to customize the banner

1. If it is not already opened, start the IBM Case Manager administration client.
 - URL: <http://ecmedu01:9080/navigator/?desktop=icmadmin>
 - User name: P8admin
 - Password: IBMFileNetP8
- a. On the administration client, click the “Open Administration View” icon on the left pane. If this icon is not visible, refresh the browser.
2. Edit the client.
 - a. In the Desktops tab, right-click Custom CM Client that you created and click Edit.
 - b. Click the Appearance subtab.
3. For the Banner and login page section, enter the values in the table.

Field Name	Option	Value
*Banner logo	URL	EDUIImages/IBM_Nav_logo.png
Banner background color	Custom	#FFFFCC
Application name text color	Custom	#006600
Banner icon color	Dark	

*An image for the banner logo is already copied into the C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\P8Node01Cell\navigator.ear\navigator.war\EDUIImages folder.

The completed page looks similar to the one in the following screen capture.

Desktop: Custom CM Client

The screenshot shows the 'Appearance' tab selected in a configuration interface. The tab bar includes 'General', 'Mobile', 'Repositories', 'Appearance' (selected), and 'Menus'. The 'Appearance' section contains four configuration items:

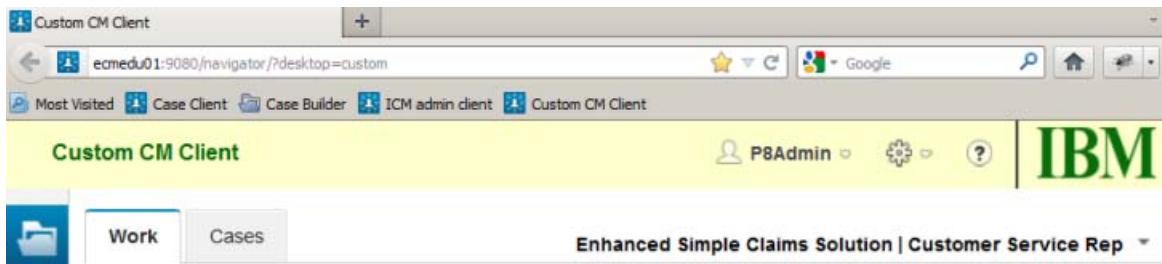
- Banner logo:** A radio button for 'IBM' is unselected, while 'URL' is selected. The URL value is 'EDUImages/IBM_Nav_logo.png'.
- Banner background color:** A radio button for 'Default' is unselected, while 'Custom' is selected. The custom color value is '#FFFFCC'.
- Application name text color:** A radio button for 'Default' is unselected, while 'Custom' is selected. The custom color value is '#006600'.
- Banner icon color:** A radio button for 'Light' is unselected, while 'Dark' is selected.

4. Click Save and Close.
 - a. Log out of the administration page.
 - b. Close the browser.

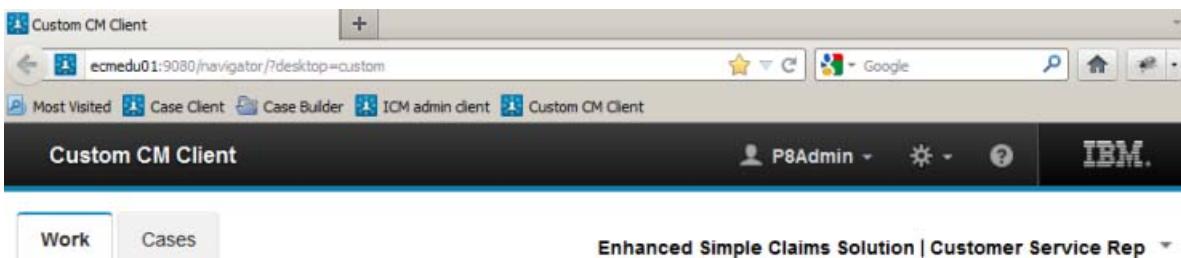
Procedure 4: Test the customized banner

1. Log in to the Custom CM Client in a new browser.
 - URL: `http://ecmedu01:9080/navigator/?desktop=custom`
 - User name: P8admin
 - Password: IBMFileNetP8
2. Verify the following changes (as you configured) in the banner for the client.
 - The banner background color
 - Banner logo
 - Application name text color

After customization:



Before customization:



3. Log out of the custom client and close the browser.

Troubleshooting

- If you are not able to access your custom desktop, check the `ID` value:
- When you created a copy of the case client, you edited the ID value to “custom” in Procedure 2: Make a copy of the existing Case Manager Client, page. 1-7.
 - Verify the `ID` value.
 - If you did not edit the ID value to “custom” at the time when you created a copy of the desktop, you can not edit it later.
 - Instead, you must use the default ID value that is shown in the IBM Case Manager admin client to access your custom client.

Exercise 1.1.2: Customize the login page

Introduction

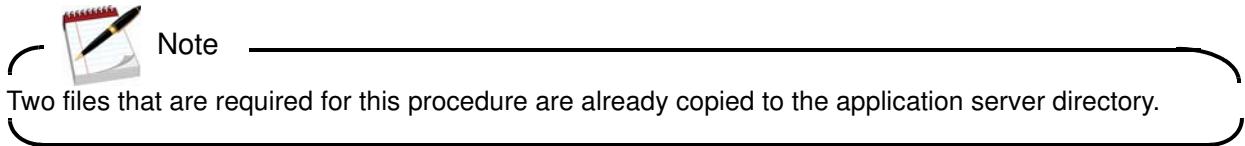
In this lab, you customize the login page for your application to add notes and a logo.

Procedures

Procedure 1: Edit the new client to customize the login page, page. 1-12

Procedure 2: Test the customized login page, page. 1-13

Procedure 1: Edit the new client to customize the login page



1. Verify the following files in the specified folder locations:

- An image file:

C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\P8Node01Cell\navigator.ear\navigator.war\EDUImages\IBM_Nav_logo.png

- An html file:

C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\P8Node01Cell\navigator.ear\navigator.war\EDUFiles\EDULoginNotes.html

2. Start the IBM Case Manager administration client.

- URL: <http://ecmedu01:9080/navigator/?desktop=icmadmin>
- User name: P8admin
- Password: IBMFileNetP8

- a. In the administration tool, click the “Open Administration View” (gear) icon on the left pane. If the Administration View icon is not visible, refresh the browser.



3. Edit the client:

- a. In the Desktops tab, right-click Custom CM Client and click Edit.
- b. Click the Appearance subtab.

- For the banner and login page section, enter the values in the following table

Field Name	Option	Value
Login page logo	URL	http://ecmedu01:9080/navigator/EDUImages/IBM_Nav_logo.png
Login page content	URL	http://ecmedu01:9080/navigator/EDUFiles/EDULoginNotes.html

**Note**

You can also enter a relative path for the files (Example: EDUFiles/EDULoginNotes.html) because these files are copied to the same web application directory. If you are using a file from a different web application, then you must enter a fully qualified URL address.

- Click Save and Close. If you are prompted, click Close in the Information window.
- Log out of the IBM Case Manager administration client and close the browser.

Procedure 2: Test the customized login page

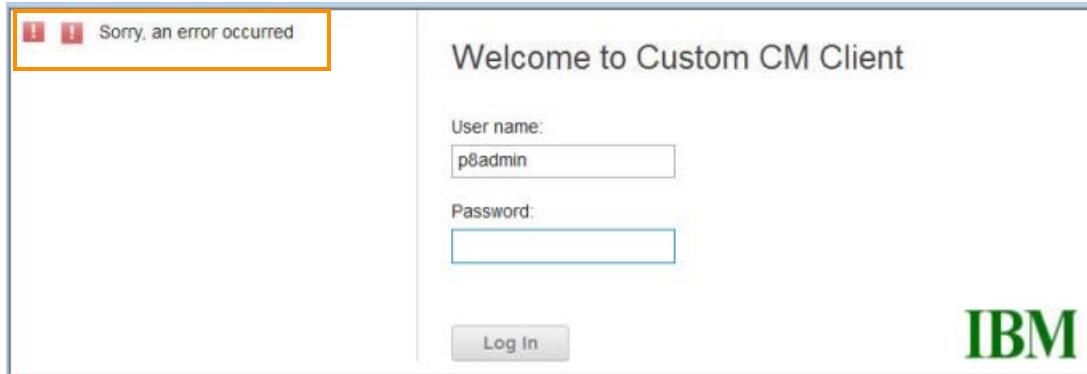
- Open the Log In page for the Custom CM Client in a new browser.
 - URL: http://ecmedu01:9080/navigator/?desktop=custom
- Verify the following changes to the login page of the custom client.
 - A new section on the left that allows the user to get help with logging in to the application.
 - A new logo in the lower right corner.

The screenshot shows the login interface for the 'Custom CM Client'. On the left, there's a sidebar with 'Terms of Use' information, including operating hours (07:30 to 19:00 Monday through Friday) and a help desk phone number (1-800-000-0000). Below that are links for 'Trouble Logging in?', 'Request Access', and 'Help'. The main login form has fields for 'User name:' (containing 'p8admin') and 'Password:', a 'Log In' button, and the 'IBM' logo.

- Close the browser.

Troubleshooting

- If the value for the “Login page client” is not correct, you get the following error:



- To correct the error, verify the value in Procedure 1: Edit the new client to customize the login page, page. 1-12 and test again.

A screenshot of a configuration dialog titled "Edit Client". Under the "General" tab, the "Login page content" section is shown. It has a dropdown menu with two options: "None" (radio button) and "URL" (radio button, which is selected). Below the dropdown is a text input field containing the URL "http://ecmedu01:9080/navigator/EDUFiles/EDULoginNot".

LESSON 1.2: Configure icons and labels

What this lesson is about?

This lesson describes how to customize the icons and labels for a custom Case Manager Case client.

What you should be able to do?

After completing this lesson, you should be able to:

- Change the icons.
- Modify labels.

How you will check your progress?

- Hands on labs.

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Change Icons for the client

Configure the icons that are shown in the web client

- You can change the icons that are displayed for document MIME types without any code.
 - MIME types indicate the type of document, such PDF or HTML.
 - The changes that you make apply to all of the clients in your environment.
- IBM Content Navigator provides predefined icons for MIME types.
 - You can map MIME types to the predefined icons.
 - You can also customize the icons to match icons that are used in your organization.
 - Example: Microsoft Office applications.

Create an Icon Mapping for MIME Types

You can create Icon Mappings to associate an icon to a MIME type.

New Icon Mapping

You can map a list of MIME types to a predefined icon that is provided with the web client or to a custom icon.

What icon do you want to use?

<input checked="" type="radio"/> Predefined icon: ?	 Presentation File Icon <input type="button" value="▼"/>
<input type="radio"/> Custom icon: ?	Specify the URL of the icon <input type="text"/>

New MIME type:

Available MIME Types application/x-tracker application/x-vnd.oasis.opendocument.spreadsheet application/x-vnd.oasis.opendocument.text application/x-workitem	Selected MIME Types application/x-vnd.oasis.opendocument.presentation
---	---

Buttons:  

- You can associate more than one icon for a MIME type.
 - If a MIME type has more than one Icon Mapping, the first icon in the list is used.

Edit an existing Icon Mapping for MIME Types

- You can edit an existing Icon Mapping to associate a new icon to a MIME type.
 - In the administration tool for the web client, click Settings.
 - In the Settings tab, click the Icon Mapping subtab.
 - Select an existing MIME type from the MIME Type Icons section.
 - Click Edit to map a MIME type to a predefined icon or to a custom icon.
-

Modify the labels in the Case Manager Client

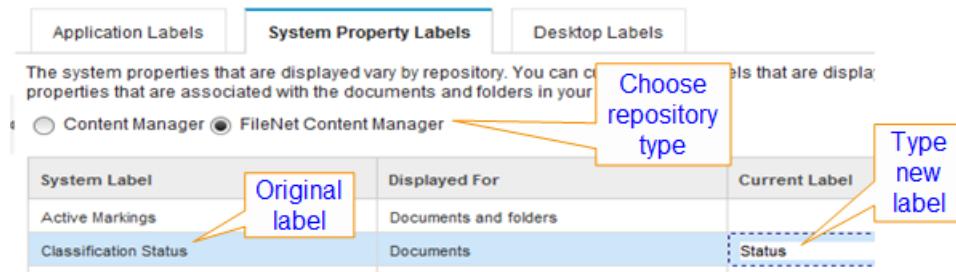
Customize labels

Customizing the labels are useful when users wants to use their own terminology.

- Changes that are made to Application Labels and System Property Labels apply to all the desktops in your environment.
- Each locale can have its own label text.
- The repositories use System Property Labels that show on desktops.

System Property Labels

The diagram shows how to change the default System Property Labels.



- Customize the labels in the Labels > System Property Labels tab.
- You can customize the labels for the system properties that are associated with the documents and folders in your repositories.
- Any changes that you make to the system property display names in the web client do not affect the property names and values that are configured on your repository.

Exercise 1.2.1: Associate a custom icon for a MIME type

Introduction

In this lab exercise, you associate a custom icon for the text/plain and application/rtf document MIME types that are stored in the repository.

Procedures

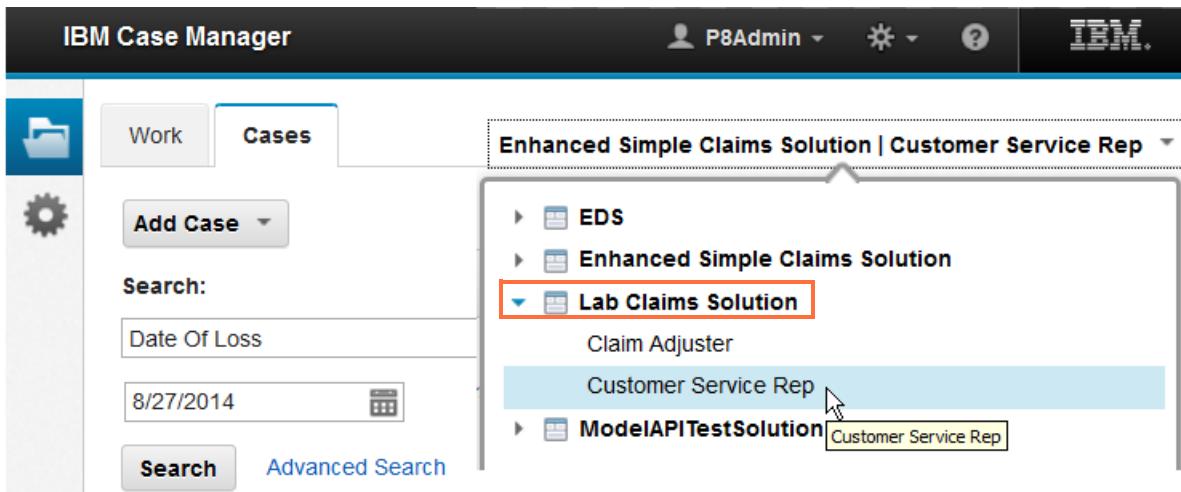
Procedure 1: Check the default icon, page. 1-19

Procedure 2: Customize the icon for text MIME type, page. 1-21

Procedure 3: Test the new custom icon, page. 1-23

Procedure 1: Check the default icon

1. Start the Custom Case Manager Client.
 - URL: <http://ecmedu01:9080/navigator/?desktop=custom>.
 - User name: P8admin
 - Password: IBMFileNetP8
2. Select Lab Claims Solution > Customer Service Rep on the upper right of the page from “Solution and Roles” list.



3. Open a case.
 - a. Click the Cases tab.
 - b. In the Search section, select Policy Family Name from the list.
 - c. Enter % in the second text box of the Search widget and click the Search button.

Search:

Policy Family Name

%

Search [Advanced Search](#)

- d. Available cases are listed in the Case List widget.
4. Open the Case details for a case.
 - a. Click the link for the case with Smith as the Policy Family Name value from Title column.
 - b. The Case details are shown in a separate tab.
5. Open the documents list for this case.
 - a. To see the complete list of documents, collapse the Timeline Visualizer widget by clicking the down arrow in the middle of the page.



- b. In Documents subtab, click Correspondence and forms folder.
6. Check the default icon that is associated with the text document MIME type.
 - a. Observe the different icons for each MIME type.
 - b. Check the icon for the Notes.txt document.

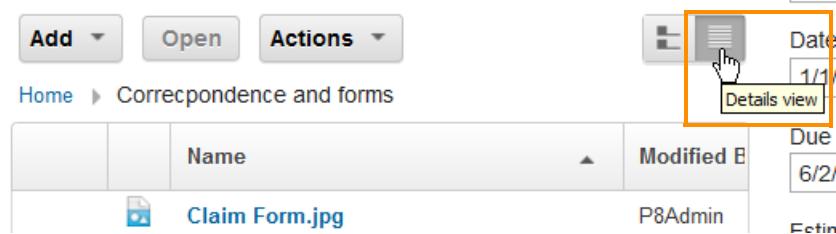
Home > Correspondence and forms

Name	Modified
Claim Form.jpg	P8Admin
Notes.txt	P8Admin
User Manual.pdf	P8Admin

7. You are going to replace this icon with a custom image.
 - a. Click Close.
8. Logout of the Case client and close the browser.

Notes - Magazine View and Details View

The document list in the screen capture in step 5 shows the Details View. Depending on the view that you have in your student system, the document list display might be different.



Procedure 2: Customize the icon for text MIME type

1. Start the IBM Case Manager administration client.
 - URL: <http://ecmedu01:9080/navigator/?desktop=icmadmin>
 - User name: P8admin
 - Password: IBMFileNetP8
2. On the administration client, click the “Open Administration View” icon on the left pane. If this icon is not visible, refresh the browser.

Troubleshooting

You can also access the administration page with this URL:
<http://ecmedu01:9080/navigator/?desktop=admin>

3. In the Administration page, click Settings on the left pane.
4. In the Settings tab > Icon Mapping subtab, select text/plain, application/rtf... MIME type.

MIME Type	Predefined Icon
application/pdf	PDF File Icon
text/plain, application/rtf, application/x-rtf, text/richtext, application/dca-rtf	Plain Text File Icon
text/html, text/htm	Web Document Icon



Hint

Observe that the list has a similar item with the MIME type “text/html” for Web Document Icon. For this lab, select the “Plain Text File Icon” item with the MIME type “text/plain”.

5. Click Edit.
 - a. In the Icon Mapping window, select the “Custom icon” option.
 - b. Enter `EDUImages/TXTIcon.png` as the value. This image file is already copied to your student system.

Icon Mapping

You can map a list of MIME types to a predefined icon that is provided with the web client or to a custom icon.

What icon do you want to use?

Predefined icon: `Plain Text File Icon` Custom icon: `EDUImages/TXTIcon.png`

New MIME type:

Available MIME Types	Selected MIME Types
<code>application/afp</code>	<code>text/plain</code>
<code>application/csbundled</code>	<code>application/rtf</code>

- c. Click OK.



Note

You can include more MIME types to associate with this icon by moving items from the Available MIME Types list to the Selected MIME Types list. To add an item that is not in the Available list, you can enter the name in the “New MIME type” field and click Add.

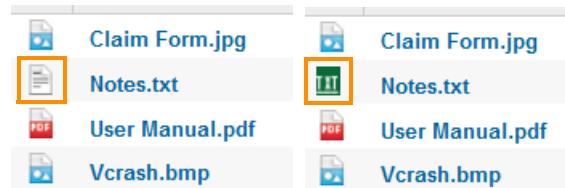
6. Back in the Icon Mapping tab, verify that the text/Plain MIME type shows a new icon that you edited.
 - a. The Custom Icon column shows the relative URL (path) for the image file.

	MIME Type	Predefined Icon	Custom Icon
	application/pdf	PDF File Icon	
	text/plain, application/rtf, application/x-rtf, text/richtext, application/dca-rtf		EDUImages/TXTIcon.png

7. Click Save and Close.
8. Log out of the administration tool and close the browser.

Procedure 3: Test the new custom icon

1. Repeat Procedure 1: Check the default icon, page. 1-19 to test the custom icon that is associated with the text document MIME type.
 - a. Verify the new icon for the Notes.txt document.
 - b. For comparison, the default icon before the customization is also shown.



2. Log out of the client and close the browser.

Troubleshooting

If the new icon is not shown, log out, close the browser and log in back. The Firefox browser in the student image is configured to clear the browser cache when you close the browser.

Exercise 1.2.2: Modify labels in the IBM Case Manager Client

Introduction

In this lab, you edit an Application Label name and customize it.

Procedures

Procedure 1: Check the default Application Label, page. 1-24

Procedure 2: Customize Application Labels, page. 1-25

Procedure 3: Test the new label, page. 1-26

Procedure 1: Check the default Application Label

1. Start the Custom Case Manager Client.
 - URL: <http://ecmedu01:9080/navigator/?desktop=custom>
 - User name: P8admin
 - Password: IBMFileNetP8
2. Open a case.
 - a. Click the Cases tab.
 - b. In the Search section, select Policy Family Name from the list.
 - c. Enter % in the second text box of the Search widget and click the Search button.

Search:

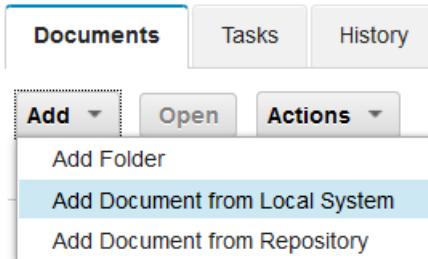
Policy Family Name

%

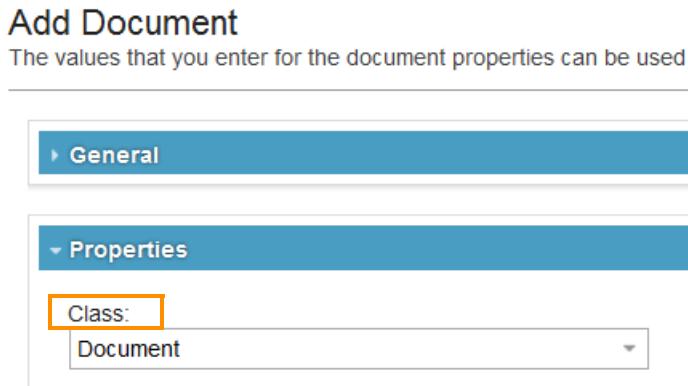
Search **Advanced Search**

- d. Available cases are listed in the Case List widget.
3. Open the Case details for a case.
 - a. Click the link for the case with Smith as the Policy Family Name value from Title column.
You can select any case from the list.
 - b. The Case details are shown in a tab.

4. Add a document to the case.
 - a. In Documents subtab, click Add > Add Document from Local System.



5. Check the label for Class.
 - a. In the Add Document wizard > Properties section, check that the label name for the document class is Class.



- b. You are going to change this label name to Object Type.
- c. Click Cancel to close the wizard.
- d. Logout of the Case client and close the browser.

Procedure 2: Customize Application Labels

1. Start the IBM Case Manager administration client.
 - URL: <http://ecmedu01:9080/navigator/?desktop=icmadmin>
 - User name: P8admin
 - Password: IBMFileNetP8
2. On the administration client, click the “Open Administration View” icon on the left pane. If this icon is not visible, refresh the browser.
3. In the admin tool, click Labels on the left pane.
4. In the Labels tab on the right > Application Labels subtab, select Class in the Default Label column.

5. Double-click in the space next to that label in the Current Label column.
 - a. Enter Object Type as the new label.

Default Label	Current Label
Class	Object Type

- b. Click Save and Close.
6. Log out of admin tool and close the Browser.

Procedure 3: Test the new label

1. Repeat Procedure 1: Check the default Application Label, page. 1-24 to test the new label.
2. Check the label for Class.
 - a. In the Add Document wizard > Properties section, check that the label name for the document class is changed to Object Type.

Add Document
The values that you enter for the document properties can be used

General

Properties

Object Type:
Document

- b. Click Cancel to close the wizard.
- c. Logout of the Case client and close the browser.

 Troubleshooting

If the new label is not shown, log out, close the browser and log in back. The Firefox browser in the student image is configured to clear the browser cache when you close the browser.

LESSON 1.3: Specify viewers for file types

What this lesson is about?

This lesson describes how to create a Viewer Map and associate it with the Case Manager custom client. Viewer Maps specify which viewer to use to open a file type.

What you should be able to do?

After completing this lesson, you should be able to:

- Create a Viewer Map and associate it with the Case Manager custom client.

How you will check your progress?

- Hands on labs.

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Viewer Maps

Viewer maps specify which viewer is used to open each file type.

- The Case Manager administration tool includes a default Viewer Map.
 - It includes Viewer Mappings for each repository type.
 - Cannot edit the default Viewer Map.
 - Can create copies of the default Viewer Map and customize.
- You can create a custom Viewer Map for the following tasks.
 - Use a different viewer for a file type.
 - Add more file types to the mapping.
- You can create one or more Viewer Maps.
 - Can associate only one Viewer Map with each custom client.

Fallback Viewers

- Each viewer supports a specific set of file types.
 - Some viewers are not supported on every server or client platform that IBM Content Navigator supports.
- IBM Content Navigator can automatically select a different viewer to use if another viewer is included in your Viewer Map.
- This behavior is called the Fallback behavior, and the second viewer is called a Fallback viewer.
- You can include multiple Fallback Viewers in your Viewer Map.

Custom Viewers

- You can create custom viewers programmatically as a plug-in and associate it with IBM Content Navigator.

Viewers that support all MIME types

- Some viewers support only specific MIME types (Adobe Reader supports PDF type).
- The following viewers in IBM Content Navigator support all MIME types:
 - AJAX Viewer
 - Applet Viewer
 - HTML Conversion
 - PDF Conversion
 - Web Browser

Exercise 1.3.1: Create a Viewer Map for PDF files

Introduction

In this lab, you create a custom Viewer Map and associate it with your custom case client. In your custom Viewer Map, you replace the default viewers for the PDF files with Adobe Reader.

Procedures

Procedure 1: Check the existing viewer for Adobe PDF files, page. 1-29

Procedure 2: Create a Viewer Map, page. 1-31

Procedure 3: Assign the new Viewer Map to your case client, page. 1-33

Procedure 4: Test the new Viewer Map, page. 1-33

Procedure 1: Check the existing viewer for Adobe PDF files

1. Start the custom Case Manager Client.
 - URL: <http://ecmedu01:9080/navigator/?desktop=custom>.
 - User name: P8admin
 - Password: IBMFileNetP8
2. In the Case Manager Client, if it is not already selected, select Lab Claims Solution > Customer Service Rep from the upper right of the page from “Solution and Roles” list.
3. Search for cases.
 - a. Click the Cases tab.
 - b. In the Search section, select Policy Family Name from the list.
 - c. Enter % in the second text box of the Search widget and click the Search button.

Search:

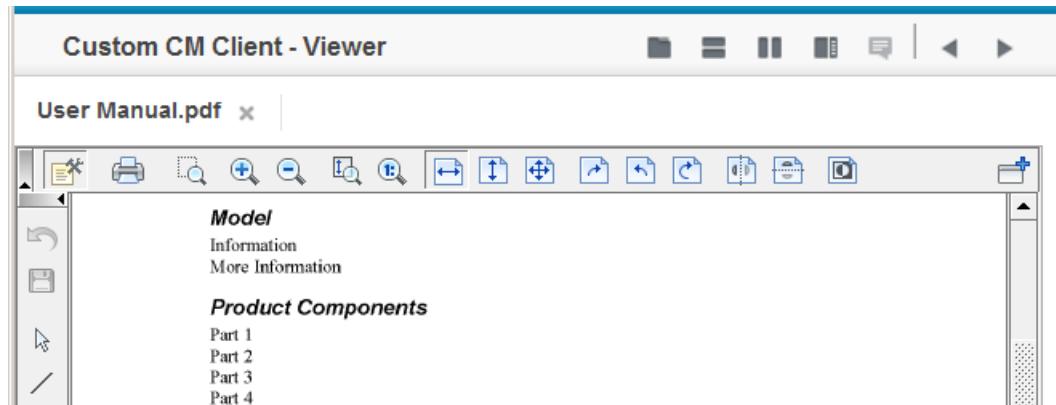
Policy Family Name

%

Search [Advanced Search](#)

- d. Available cases are listed in the Case List widget.
4. Open the documents list for a case.
 - a. Click the link for the case with Smith as the Policy Family Name value from Title column.
 - b. The details for that case are shown in a separate tab.

- c. In Documents subtab, click Correspondence and forms folder.
 - d. To see the complete list of documents, collapse the Timeline Visualizer widget.
5. Check the viewer that is available to open a PDF document.
- a. Scroll down and double-click a PDF document to open it in the default viewer.
 - b. Click Run if you are prompted.
 - c. The Document opens in the default viewer.

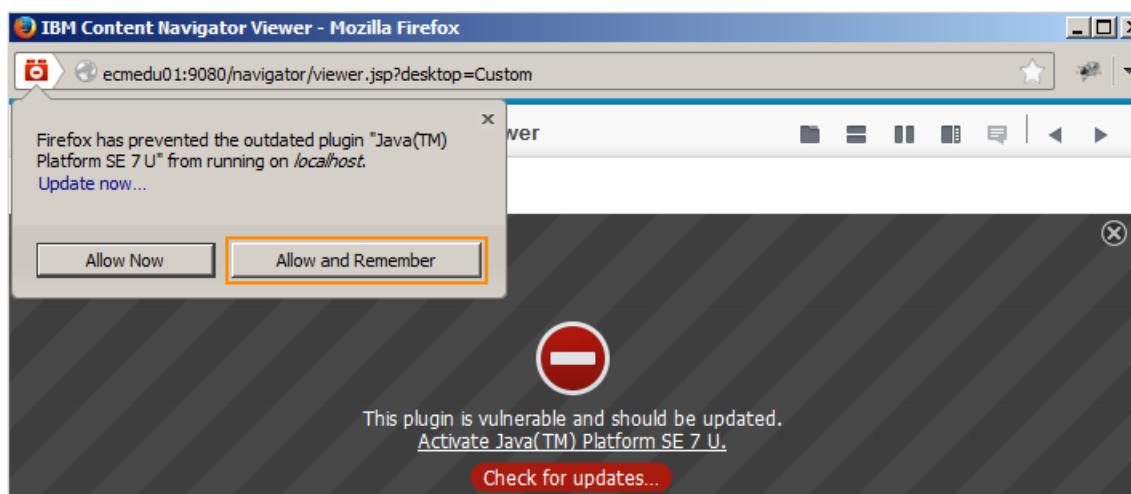


6. You are going to change the default viewer that is assigned to the PDF documents.
- a. Close the Viewer.



Troubleshooting

If you get the following error when you try to open a document in the viewer, do Step 7. Otherwise, skip to Procedure 2.



7. Click the red icon at the upper left corner.
 - a. Click “Allow and Remember” in the window.
 - b. Do not update to new Java version.
 - c. In the “Java Update Needed” window, select “Don not ask again until the next update is available” and click Later.
8. Log out and close the browser.

Procedure 2: Create a Viewer Map

1. Start the IBM Case Manager administration client.
 - URL: `http://ecmedu01:9080/navigator/?desktop=icmadmin`
 - User name: P8admin
 - Password: IBMFileNetP8
2. On the administration client, click the “Open Administration View” icon on the left pane. If this icon is not visible, refresh the browser.
3. Click **Viewer Maps** on the left pane.
4. Create a Viewer Map.
 - a. In the Viewer Maps tab on the right pane, click **New Viewer Map**.
The New Viewer Map contains a copy of all the mappings from the default Viewer Map.
 - b. In the New Viewer Map tab, enter `EDU Viewer Map` in the Name field.
The ID value is populated automatically.
 - c. Edit the description. (Example: It maps Adobe Reader for the PDF files).
5. Check the mappings for the FileNet Content Manager repository.
 - a. There are two mappings in the list. Observe the application/pdf MIME type

Repository Type	Viewer	MIME Type
FileNet Content Manager	FileNet Viewer	image/jpeg, image/jpg, image/jpeg, image/bmp, image/gif, image/tiff, image/x-png, application/pdf , application/x-cold, application/vnd.filenet.im-image, application/vnd.filenet.im-cold, application/vnd.filenet.im-other, image/png

[FileNet Content Manager](#) [Web Browser](#) [All MIME types](#)

6. Remove the application/pdf MIME type from the FileNet Viewer mapping.
 - a. Select the FileNet Viewer row for FileNet Content Manager and click **Edit**.
 - b. In the Mapping page > Selected MIME Types pane, scroll down and select the application/pdf.
 - c. Move it from the Selected MIME Types pane to the Available MIME Types pane by clicking the arrow. Click **OK**.

This step removes the option to view the PDF files in the FileNet Viewer.

- d. In the New Viewer Map, verify that the change is updated. The application/pdf MIME type is removed from the list.

 Note

The Case client selects Web Browser as the first Fallback Viewer. For the case client to use the new mapping, you must remove the application/pdf MIME type from the Web Browser mapping.

7. Remove the application/pdf MIME type from the Web Browser mapping for the FileNet Content Manager.
 - a. Select the Web Browser Viewer row for the FileNet Content Manager and click Edit.

Repository Type	Viewer	MIME Type
FileNet Content Manager	Web Browser	All MIME types

- b. In the Mapping page, clear the “All MIME types” option.
8. Move all the items in the Available MIME Types pane except application/pdf.
 - a. Select all the items in the Available MIME Types pane by pressing shift and click.
 - b. Move them to the Selected MIME Types pane by clicking the forward arrow.
 - c. In the Selected MIME Types pane, select the application/pdf.
 - d. Move it from the Selected MIME Types pane back to the Available MIME Types pane by using the arrow.
 - e. Click OK.
- This step removes the option to view the PDF files in the Web Viewer.
- f. In the New Viewer Map, verify that the change is updated. A list of MIME types is shown.
9. Create a Mapping for pdf.
 - a. Click New Mapping.
 - b. In the New Mapping page, select FileNet Content Manager for the “Repository type” field.
 - c. Select Adobe Reader for the Viewer field.
 - d. Move application/pdf from the Available MIME Types pane to the Selected MIME Types pane by using the arrow. Click OK.
 - e. In the New Viewer Map tab, click Save and Close.
10. In the Viewer Maps, verify that the newly created map is listed.

	Name	ID	Description
	EDU Viewer Map	EDUViewerMap	It maps Adobe Reader for the PDF files

11. Leave the application open for the next procedure.

Procedure 3: Assign the new Viewer Map to your case client

You replace the default Viewer Map that is assigned to your case client with the new one that you created.

1. Click the Desktops tab. Select Custom CM Client and click Edit.
2. In the General tab > Desktop configuration section, select the new Viewer Map from the list.

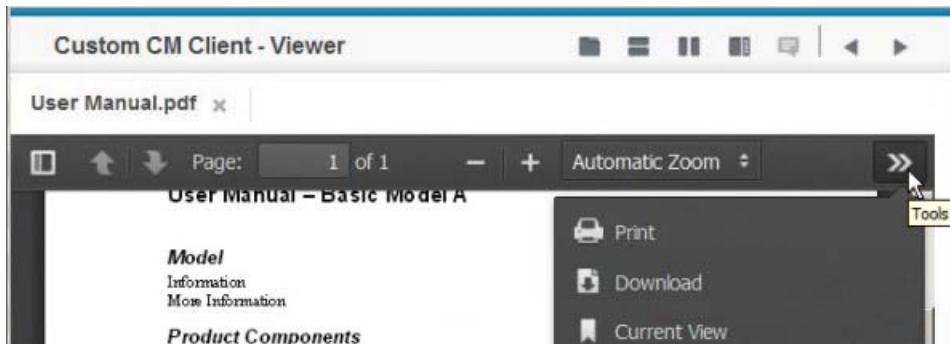
The screenshot shows a user interface for managing desktop configurations. A blue header bar says 'Desktop configuration'. Below it, there's a field labeled 'Viewer map:' with a question mark icon. To the right of the field is a button labeled 'EDU Viewer Map' with a blue border, indicating it is the selected option.

3. Click Save and Close.
4. Log out of the Case client and close the browser.

Procedure 4: Test the new Viewer Map

In this procedure, you test the new Viewer Map by opening a PDF file.

1. Repeat Procedure 1: Check the existing viewer for Adobe PDF files, page. 1-29, to open a PDF document.
 - a. Verify that the PDF file is opened in the Adobe Reader that is associated with Firefox.



2. Close the viewer, log out of the case client and close the browser.



Troubleshooting

If the changes that you made are not reflected, and you are not able to see the pdf document in Adobe Reader, stop and start the WebSphere Application Server and test again. Use the Stop_Server1.bat file in the WebSphere Admin folder on the desktop to stop the server. Use the Start_Server1.bat file in the same folder to start the server.

Exercise 1.3.2: View the Microsoft Word documents in the FileNet Viewer

Introduction

The default setting for the Word documents is to open them in Microsoft Word or Word Viewer from a web browser. You can configure a viewer map to open the Microsoft Word documents in the default FileNet Viewer for a quick reviewing and adding annotations.

In this lab, you modify the custom Viewer Map that you created in the previous exercise to view the Microsoft Word documents in the FileNet Viewer.

Procedures

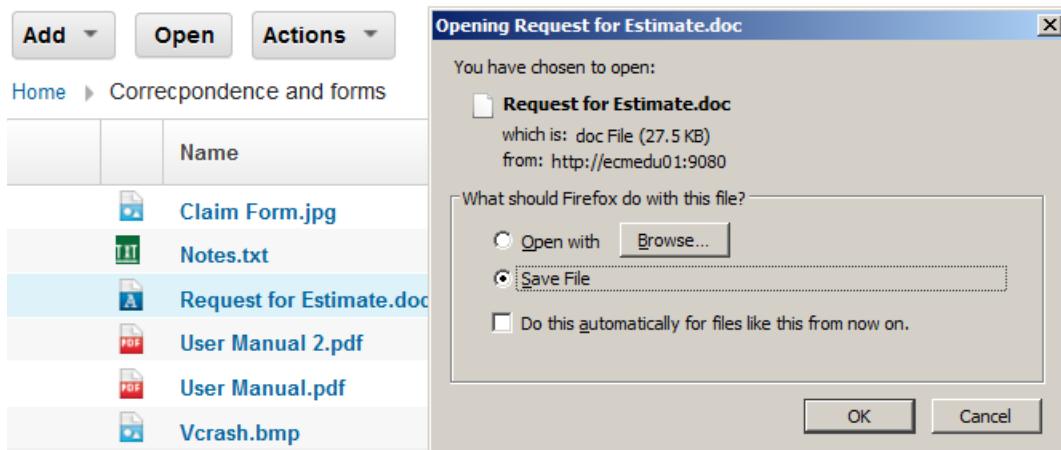
Procedure 1: Check the available viewers for Microsoft Word documents, page. 1-35

Procedure 2: Edit the custom Viewer Map, page. 1-36

Procedure 3: Test the modified custom Viewer Map, page. 1-37

Procedure 1: Check the available viewers for Microsoft Word documents

1. Check the viewer that is available to open a Microsoft Word document for a case.
 - a. Repeat Procedure 1: Check the existing viewer for Adobe PDF files, page. 1-29 from the previous exercise to open a Word document.
 - b. Verify that the browser prompts you to select an external viewer for the Word document. If the system has Microsoft Word, you are prompted to open the document in it.



2. Click Cancel to close the dialog, log out, and close the browser.

You are going to configure a default viewer for the Word documents.

Procedure 2: Edit the custom Viewer Map

1. Start the IBM Case Manager administration client.
 - URL: `http://ecmedu01:9080/navigator/?desktop=icmadmin`
 - User name: `P8admin`
 - Password: `IBMFleNetP8`
2. On the administration client, click the “Open Administration View” icon on the left pane. If this icon is not visible, refresh the browser.
3. Click `Viewer Maps` on the left pane.
4. Edit the custom Viewer Map that you created earlier.
 - a. In the `Viewer Maps` tab on the right pane, select `EDU Viewer Map` and click `Edit`.
 - b. In the `EDU Viewer Map` tab, edit the description. (Example: `Viewer map for the PDF and Word files`).
5. Check the mappings for the FileNet Content Manager repository.
 - a. Observe the `application/msword` MIME type for the following mappings in the list.
 - Web Browser (it has the `application/msword` MIME type)
 - FileNet Viewer (it does not have the `application/msword` MIME type)
6. Add the `application/msword` MIME type to the FileNet Viewer mapping.
 - a. Select the FileNet Viewer row (FileNet Content Manager) and click `Edit`.
 - b. In the Mapping page > Available MIME Types pane, select the `application/msword`.
 - c. Move it from the Available MIME Types pane to the Selected MIME Types pane by using the arrow.
 - d. Click `OK`.

This step adds the option to view the Microsoft Word documents in the FileNet Viewer.
7. Remove the `application/msword` MIME type from the Web Browser mapping for the FileNet Content Manager.



Note

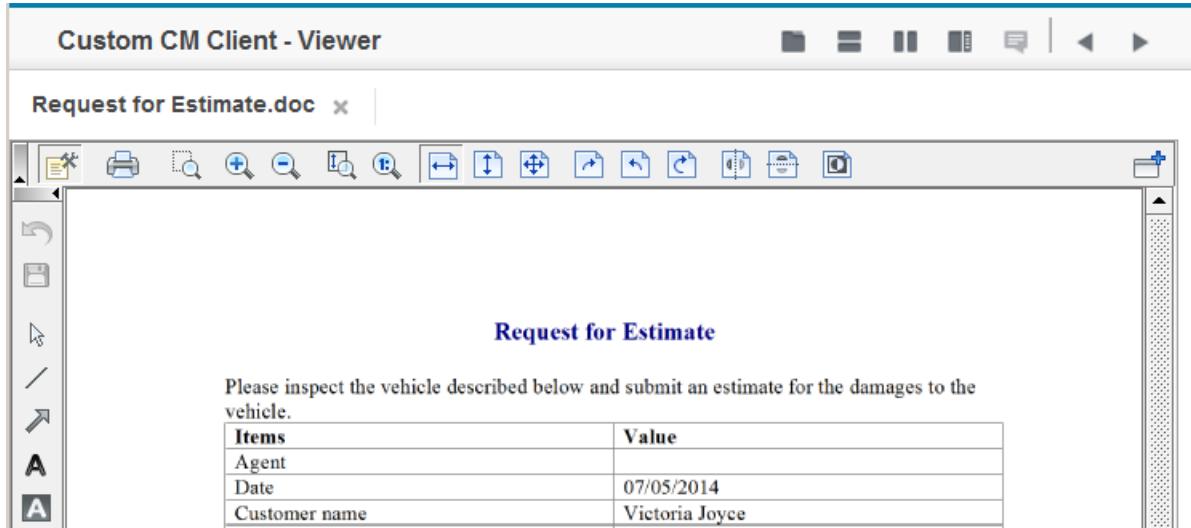
The IBM Content Navigator (Case client) selects Web Browser as the first Fallback Viewer. For the Case Manager Client to use the new mapping, you must remove the `application/msword` MIME type from the Web Browser mapping.

- a. Select the Web Browser Viewer row for the FileNet Content Manager click `Edit`.
- b. In the Mapping page > Selected MIME Types pane, select the `application/msword`.
- c. Move it from the Selected MIME Types pane to the Available MIME Types pane by using the arrow.
- d. Click `OK`. This step removes the option to view the Microsoft Word documents in the Web Viewer.

- e. In the EDU Viewer Map, verify that the application/msword MIME type is removed.
8. Click Save and Close.
9. Log out of the application and close the browser.

Procedure 3: Test the modified custom Viewer Map

1. Repeat Procedure 1: Check the existing viewer for Adobe PDF files, page. 1-29 from the previous exercise to open a Word document.
2. Verify that the Word document is now opened in FileNet Viewer as you configured.



3. Close the Viewer.
4. Log out of Case client and close the browser.

 Troubleshooting

If the changes that you made are not reflected and you are not able to see the Microsoft Word document in the Viewer, stop and start the WebSphere Application Server and test again. Use the Stop Server1.bat file in the WebSphere Admin folder on the desktop to stop the server. Use the Start Server1.bat file in the same folder to start the server.

LESSON 1.4: Create and customize help topics

What this lesson is about?

This lesson describes how to edit or add custom help topics to the IBM Case Manager help system and how to create a help plug-in.

What you should be able to do?

After completing this lesson, you should be able to:

- Edit the existing help topics.
- Create a help plug-in.
- Add custom help topics to the IBM Case Manager help system.

How you will check your progress?

- Hands on labs.

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Add custom help topics

Help system for IBM Case Manager

You can modify the online help files or add your own help files for your users in IBM Case Manager.

- The help system for IBM Case Manager uses the formatting of Eclipse plug-ins to display content.
 - The help files are created in Darwin Information Typing Architecture (DITA) XML.
 - They are converted to XHTML and use a cascading style sheet named `swg_info_common.css`.
- Help files are installed in the following location: IBM Case Manager_installation_directory /help/content.
 - They are then copied to the network shared directory when you configure IBM Case Manager.
 - You must modify the files in the network shared directory for changes to be reflected in the help system.
- The location for the network shared directory is specified on the “Set properties for development environment” page in the Create Profile wizard.
 - This wizard is run from the IBM Case Manager administration client.
 - The default value for network shared directory on a Windows server: C:\Program Files(x86)\IBM\CaseManagement\configure\properties\help\content\.

Create new help topics

You can create new help files to add to the IBM Case Manager help system, if you want to add custom instructions in a separate file.

If you introduce a new cascading style sheet (CSS) for the help files, modify the HTML to use the new CSS file.



Note

Any new files are overwritten in this location (network shared directory that you configured for IBM Case Manager) if you upgrade to a new version of IBM Case Manager.

Creating a help plug-in

Help plug-in

You can create a new help plug-in for the IBM Case Manager help system to add a section of custom content for your users.

- To create your own help plug-in:

1. Create a subdirectory for your new or modified help in the following directory: IBM Case Manager_installation_directory/network_shared_directory/help/content/.

Example location: C:\Program Files
(x86)\IBM\CaseManagement\configure\properties\help\content



Important

Placing your topics in a separate subdirectory ensures that when an upgrade or fix pack is applied, your help are not overwritten.

2. Create all content files in the Extensible Hypertext Markup Language (XHTML) format.
3. Create table of contents files that describe the navigation for your topics. The following example shows a simple table of contents in an XML file called toc.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<toc label="My Custom Help">
    <topic label="Help Topic 1" href="Help1.htm"></topic>
    <topic label="Help Topic 2" href="Help2.htm"></topic>
    <topic label="Help Topic 3" href="Help3.htm"></topic>
</toc>
```

4. Create the plugin.xml file that extends the org.eclipse.help.toc extension point and specifies table of contents files.

Every plug-in requires a file that is called plugin.xml to identify the plug-in contents to the system. The id for the <plugin> element should match the subdirectory name.

```
<?xml version="1.0" encoding="utf-8"?>
<plugin id="com.mycustomhelp.doc" name="My Custom Help"
provider-name="CompanyName" version="1.0">
    <extension point="org.eclipse.help.toc">
        <toc file="toc.xml" primary="true"/>
    </extension>
</plugin>
```

5. Restart the application server.

Exercise 1.4.1: Edit the existing help topics

Introduction

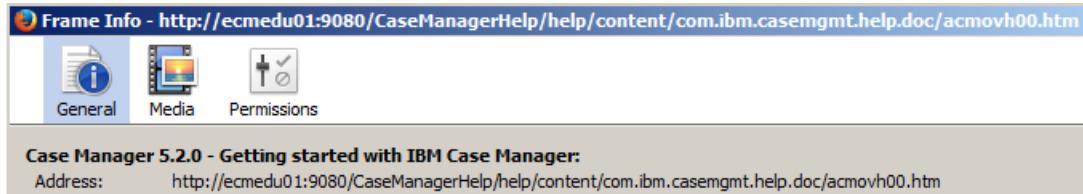
In this lab, you edit the existing help topics in the IBM Case Manager help system.

Procedures

Procedure 1: Edit the existing help topics, page. 1-41

Procedure 1: Edit the existing help topics

1. View the default help system for IBM Case Manager.
 - a. In a Firefox browser, go to the URL:
<http://ecmedu01:9080/CaseManagerHelp/index.jsp>
 - b. Expand the IBM Case Manager help node in the left pane and click the “Getting Started with IBM Case Manager” node.
2. Select a page in help system that you want to edit.
 - a. For this exercise, you use “Getting Started with IBM Case Manager” page.
 - b. Get the URL for this Frame by right-clicking on the page in the right pane, and select This Frame > View Frame Info.



- c. From the Address field, note down the package and file name to edit:
/com.ibm.casemgmt.help.doc/acmovh00.htm
 - d. Close the Frame Info page.
 - e. Leave the help system open in the browser to test the help after editing.
3. Go to the following directory and open the acmovh00.htm file in a text editor (Notepad++):
C:\Program Files (x86)\IBM\CaseManagement\configure\properties\help\content\com.ibm.casemgmt.help.doc\
 4. Edit the file to add a section for a custom topic.
 - a. For this section, you can add any changes that you want or do the Steps 4b.
 - b. Enter the text on Line 44 and 45 as shown in the following screen capture.

```
44 <li class="link nlchildlink"><span class="nlchildlinktext"><a href="acmsdh00.htm">
45   Adding a Custom Help Topic </a></span><br />
46 <li class="link nlchildlink"><span class="nlchildlinktext"><a href="acmsdh00.htm">
47   You can use the <span class="keyword">instructions</span> in the Student Notebook.
      Adding and deploying a case management solution</a></span><br />
      You can use the <span class="keyword">Case Manager Builder</span> to
```

**Hint**

Optionally, copy the text from C:\ICM\CustomHelpTopics\acmovh00_Solution.htm and paste it.

- c. Save your changes and close the file.
5. Test the “Getting Started with IBM Case Manager” page.
 - a. Repeat Step 1 to access the help page and refresh the browser.
6. Verify that the edits that you added is reflected on the page and it looks like the following screen capture.

Getting started with IBM Case Manager

Start here to learn how IBM® Case Manager enables you to create solutions, including defining roles, document classes, case types, case properties, tasks, and human oriented processes.

Adding a Custom Help Topic

You can use the instructions in the Student Notebook.

Adding and deploying a case management solution

You can use the Case Manager Builder to add a solution that case workers will access from the Case

7. Close the browser.

**Note**

You can also use Eclipse as the text editor.

Exercise 1.4.2: Create a help plug-in

Introduction

IBM Case Manager installation provides a default help plug-in for the Case Manager Client in the installation directory. They are not yet copied to the network shared directory.

In this lab, you use the default plug-in as a base for your plug-in. You explore, edit, and deploy the plug-in.

Procedures

Procedure 2: Creating a help plug-in, page. 1-44

Procedure 3: Add more topics to the custom help system, page. 1-46

Procedure 1: Stop the help application

1. Start the WebSphere Integrated Solutions Console:
 - URL: <http://ecmedu01:9043/ibm/console/logon.jsp>
 - User name: P8Admin
 - Password: IBMFileNetP8
2. In the left pane, expand Applications > Application Types and select WebSphere enterprise applications.
3. In the right pane, select the Case Manager Help application from the list and click Stop.

The screenshot shows the 'Enterprise Applications' interface. At the top, there's a header with 'Enterprise Applications' and a note: 'Use this page to manage installed applications. A single application can be deployed onto multiple servers.' Below the header are buttons for 'Start', 'Stop', 'Install', 'Uninstall', 'Update', 'Rollout Update', 'Remove File', and 'Export'. Underneath these buttons is a toolbar with icons for file operations. The main area has a table with columns for 'Select', 'Name', and 'Application Status'. The table lists several applications: CaseBuilder, CaseEDS, CaseForms, CaseManager, and CaseManagerHelp. The 'CaseManagerHelp' row is selected, indicated by a checked checkbox in the 'Select' column and highlighted with an orange border around the entire row. The 'Application Status' column shows a blue circular icon with a white question mark.

Procedure 2: Creating a help plug-in

1. To create a subdirectory for your new or modified help:
 - a. Copy the com.ibm.casemgmt.client.doc directory from the following location:
C:\Program Files (x86)\IBM\CaseManagement\help\content
 - b. Paste it to the following location:
C:\Program Files(x86)\IBM\CaseManagement\configure\properties\help\content
 - c. Rename the package to com.ibm.casemgmt.client.**custom**.doc
2. Sample content files are already created in the Extensible Hypertext Markup Language (XHTML) format.
3. Edit the table of contents file:



Note

You must have a file for table of contents to show as a navigation tree in the left pane. A table of contents file (acmrt.xml) that describe the navigation for the topics is provided.

- a. Rename the acmrt.xml file to toc.xml in the following directory.
C:\Program Files (x86)\IBM\CaseManagement\configure\properties\help\content\com.ibm.casemgmt.client.custom.doc
- b. Open the toc.xml file and view the content.
- c. Change the top-level label “Managing cases” in line 4 to “IBM Case Manager Custom Client Help”.

```
toc.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <?NLS TYPE="org.eclipse.help.toc"?>
3  <!--@ Copyright IBM Corporation 2014-->
4  <toc label="IBM Case Manager Custom Client Help" topic="acmrth00.htm">
5    <criteria name="prodname" value="Managing a case"/>
```

- d. Save your changes and close the file.
4. Edit the plugin.xml file (from the same directory specified in Step 3a) as shown in the following text. Specify the subdirectory and toc file. Required changes are shown in bold font.



Note

Every help plug-in requires a file that is called plugin.xml to identify the plug-in contents to the system. The id for the <plugin> element should match the subdirectory name. This plugin.xml file must extend the org.eclipse.help.toc extension point and specify the table of contents files.

```
<?xml version="1.0" encoding="utf-8"?>
<?xml eclipse version="3.0"?>
<plugin id="com.ibm.casemgmt.client.custom.doc" name="My Custom Help"
provider-name="IBM" version="5.2">
    <extension point="org.eclipse.help.toc">
        <toc file="toc.xml" primary="true"/>
    </extension>
    ...

```

**Hint**

Optionally, copy the text from C:\ICM\CustomHelpTopics\plugin_Solution.xml and paste it.

- a. Save your changes and close the file.
5. Edit the MANIFEST.MF file in the following directory to provide the custom bundle name:
C:\Program Files (x86)\IBM\CaseManagement\configure\properties\help\content\com.ibm.casemgmt.client.custom.doc\META-INF

```
MANIFEST.MF
1 Manifest-Version: 1.0
2 Bundle-ManifestVersion: 2
3 Bundle-Localization: plugin
4 Bundle-Name: %name
5 Bundle-Vendor: %providerName
6 Eclipse-LazyStart: true
7 Bundle-SymbolicName: com.ibm.casemgmt.client.custom.doc; singleton=true
8 Bundle-Version: 5.2.0
```

- a. Save your changes and close the file.
6. Start the help application in the Web Application Server:
 - a. Refer to the steps in Procedure 1: Stop the help application, page. 1-43 and start the application.
7. Verify that the help plug-in is deployed and the new help topic is shown in the navigation.
 - a. In a Firefox browser, go to the URL:
<http://ecmedu01:9080/CaseManagerHelp/index.jsp>
 - b. Click the IBM Case Manager Custom Client Help node. Check that the help topic is shown in the right pane.
 - c. For comparison, the help system before the customization is also shown.



8. Close the browser.

Procedure 3: Add more topics to the custom help system

In this procedure, you add more topics to your custom help system that you deployed in the previous procedure.

1. Create a custom help topic (in XHTML format as `htm` file) in a text editor and save it in the following directory: `C:\Program Files (x86)\IBM\CaseManagement\configure\properties\help\content\com.ibm.casemgmt.client.custom.doc`
 - a. For this exercise, name the file as `MyCustomHelpTopic.htm`
 - b. You reference this file name in the following step.



Hint

Optionally, copy the `MyCustomHelpTopic.htm` file from `C:\ICM\CustomHelpTopics` folder to the directory in Step 1.

2. Declare the new help file in the `toc.xml` file.

- a. Open the `toc.xml` file in the following directory:

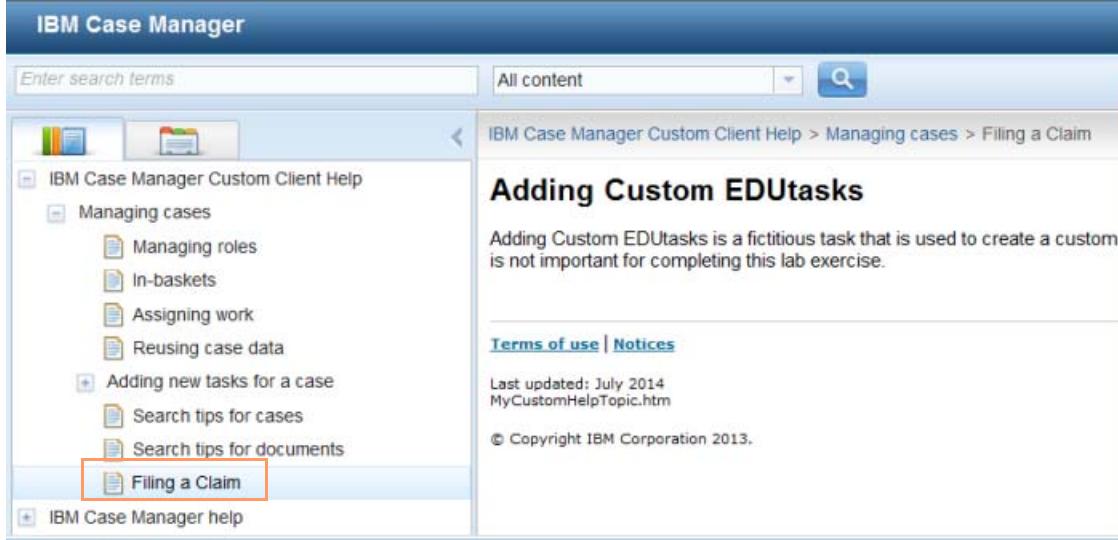
`C:\Program Files (x86)\IBM\CaseManagement\configure\properties\help\content\com.ibm.casemgmt.client.custom.doc`

- b. Add a line with your custom help file name (towards the end of the file) as shown in the following screen capture. You can provide any text for the `label` field.

```
15      </topic>
16      <topic label="Search tips for cases" href="acmrth04.htm"/>
17      <topic label="Search tips for documents" href="acmrth05.htm"/>
18      <topic label="Filing a Claim" href="MyCustomHelpTopic.htm"/>
19      </topic>
20  </toc>
```

- c. Save your changes and close the file.

3. Verify that the new topic is added to the custom help system.
 - a. In a Firefox browser, go to the URL:
`http://ecmedu01:9080/CaseManagerHelp/index.jsp`
 - b. Expand the IBM Case Manager Custom Client Help > Managing Cases node.
 - c. Check that the “Filing a Claim” help topic is shown in the left pane navigation.
 - d. Click that link and the custom page that you added is shown in the right pane.



The screenshot shows the IBM Case Manager Custom Client Help interface. The left pane displays a tree view of topics under "IBM Case Manager Custom Client Help". The "Managing cases" node is expanded, showing "Filing a Claim" which is highlighted with a red box. The right pane shows the content for "Adding Custom EDUtasks", which includes a brief description, links to "Terms of use" and "Notices", and update information.

If the changes that you made are not reflected and the new topic does not show up in the help system, restart the WebSphere Application Server, and test again: Use `Stop Server1.bat` to stop the server. Wait for the server to stop, and use `Start Server1.bat` to start the server. Wait for the Command Prompt window to close.

LESSON 1.5: Customize toolbar and menu

What this lesson is about?

This lesson describes how to implement actions by adding buttons to the toolbar, and add custom actions to the Menu.

What you should be able to do?

After completing this lesson, you should be able to:

- Customize the toolbar to implement actions.
- Add a custom action as a menu item.

How you will check your progress?

- Hands on labs.

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Toolbar widgets

Case Toolbar widget

You use the Case Toolbar widget to specify the actions that case workers can take for cases.

- These actions can be implemented as buttons or menu items in the toolbar.
- The buttons and menu items that are included by default in the Case Toolbar widget depend on the type of page that has the widget.
- You can edit the settings for the Case Toolbar widget to add, remove, and edit the buttons and menu items that are available to case workers.

Toolbar widget events

- The Case Toolbar widget handles events to display and process actions that the user can perform for a case.
 - Example: The Case Toolbar widget handles an incoming event to display the properties for a case type so that the user can create a case.
- The Case Toolbar widget provides incoming events to handle the data that is received from other widgets.
- The Case Toolbar widget publishes events if certain actions are configured for the widget.
- Example: If the Add Case action is configured for the Toolbar widget, the widget publishes the Open new case page event for that action.

IBM Case Manager pages with toolbar widget

The following table shows example of the Case Manager pages that includes the Toolbar widget by default. The table also shows the available toolbar buttons for some of the default pages.

Page Name	Toolbar buttons
Cases page	Add Case
Work page	- Add Case - Manage Roles
Add Case page	- Save Case and Close Page - Close Case Page

Case Details page	- Add Comment to Case - Add Task - Close Case Page - Save Case - Split Case
Split Case page	- Save Case and Close Page - Close Case Page
Work Details page	- Add Comment to Work Item - Close Work Details Page - Dispatch Work Item - Open Next Work Item - Reassign Item - Save Work Item

Exercise 1.5.1: Customize the toolbar to implement actions

Introduction

In this exercise, you customize the toolbar to implement actions as buttons. You add a button to show the link to a case, and another button to open a web page.

Procedures

Procedure 1: Create a custom page, page. 1-51

Procedure 2: Edit the page to customize the toolbar, page. 1-52

Procedure 3: Assign the custom page to a role, page. 1-53

Procedure 4: Redeploy the solution, page. 1-54

Procedure 1: Create a custom page

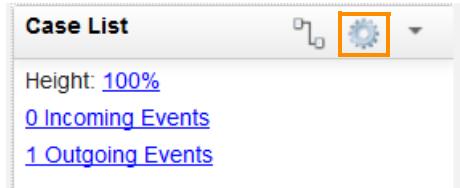
In this procedure, you copy the default Cases page and modify it. A solution for this lab is already created.

1. In Firefox, log on to IBM Case Manager Builder as an administrative user.
 - URL: <http://ecmedu01:9080/CaseBuilder>
 - User name: P8admin
 - Password: IBMFileNetP8
2. In Case Manager Builder, open Lab Claims Solution by clicking the Edit link.
Hover the mouse over the solution to see the links.
3. Create a custom page:
 - a. Open the Pages tab and expand the Solution Pages.
 - b. Click and hover the mouse over the Cases page name.
 - c. Select the copy icon on the right side of the page.
 - d. In the resulting page, edit the name to Custom Toolbar Menu for your new page and click OK to create the copy.
 - e. Save your work by clicking Save at the top of the page.
 - f. Leave the page open for the next procedure.

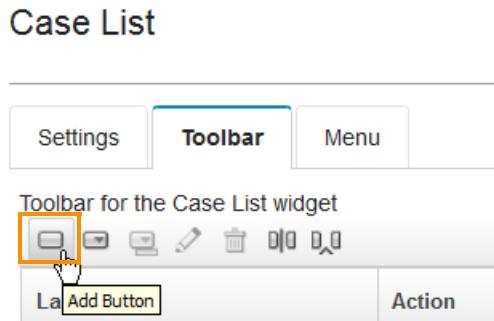
Procedure 2: Edit the page to customize the toolbar

In this procedure, you implement actions as buttons in the toolbar.

1. In the Pages tab, click the Custom Toolbar Menu page to edit it in Page Designer.
 - a. Expand the middle pane to see the tools for the Case List widget.
 - b. Click the “Edit Settings” icon for the Case List widget in the middle pane.



2. Click the Toolbar tab and then click the Add Button.



3. Select “Show Link to Case” from the list for the Action field.
 - a. Verify that “Show Link to Case” is shown for the label.
 - b. Click OK.



4. Validate that you see your newly added toolbar action.
5. Repeat step 3-5 to add another button with the following values:
 - Action: Open Web page
 - URL: <https://ecmedu01:9043/ibm/console/logon.jsp>

6. Verify that the list has the two actions that you added as shown in the screen capture.

Case List

The screenshot shows the 'Case List' settings window. At the top, there are three tabs: 'Settings', 'Toolbar' (which is selected), and 'Menu'. Below the tabs is a toolbar labeled 'Toolbar for the Case List widget' with icons for back, forward, search, and other functions. A table below the toolbar lists two actions:

Label	Action	Alignment
Show Link to Case	Show Link to Case	Left
Open Web Page	Open Web Page	Left

7. Click OK at the end of the dialog to exit the Case List settings window.
a. Click Save and then click Close at the upper right of the Page Designer window.
b. Leave the solution open for the next procedure.

Procedure 3: Assign the custom page to a role

1. Your solution is already opened in the Case Manager Builder. Open the Roles tab.
2. Click the Customer Service Rep role link.
 - a. Open the Pages subtab.
 - b. Click Assign Page.
 - c. Select the Custom Toolbar Menu Page.

The screenshot shows the 'Role Settings' window for the 'Customer Service Rep' role. The 'Pages' subtab is selected. In the 'Assign Page' dropdown, 'Custom Toolbar Menu' is selected. Below the dropdown are 'Select All' and 'Clear All' buttons. A table lists the assigned page with its name and unique ID:

Name:	Unique ID:
Custom Toolbar Menu	CustomToolbarMenu

- d. Click OK to close the dialogue window.

3. Verify that your page is listed in the Pages tab.
 - a. Click OK All to accept the changes to the role.
 - b. Click the “Save and Close” button on the solution to exit the solution editor.
 - c. Leave the Case Manager Builder open for the next procedure.

Procedure 4: Redeploy the solution

1. In the Manage Solutions page, select Lab Claims Solution and hover the mouse over.
 - a. Click Deploy.
 - b. In the Confirmation page, make sure that “Commit my changes and make them available for deployment” is selected, and click Deploy.
 - c. Wait for the green check mark to appear next to the solution.
2. Select Lab Claims Solution and hover the mouse over again.
 - a. Click Test to open Case Manager Client.

Procedure 5: Test the customized toolbar

1. In the Case Manager Client, if it is not already selected, select Lab Claims Solution > Customer Service Rep from the upper right of the page from “Solution and Roles” list.
2. Click the Custom Toolbar Menu tab to open the custom page.
3. Search for cases.
 - a. In the Search section, select Policy Family name from the list.
 - b. Enter % in the next text box of the Search widget and click the Search button.

Search:

Policy Family Name

%

Search Advanced Search

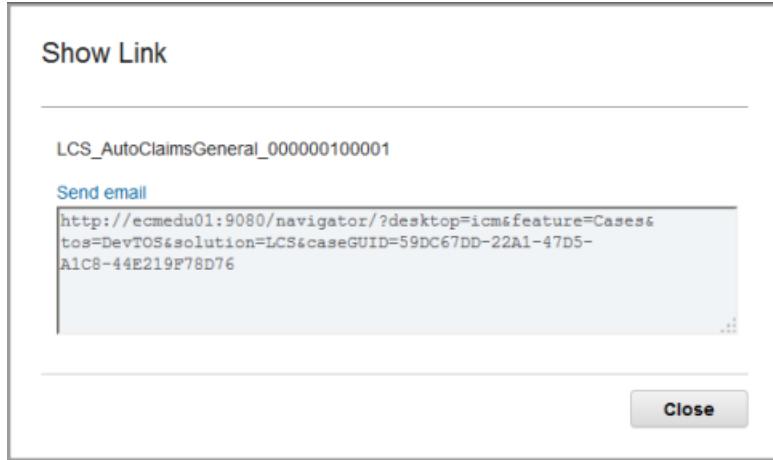
- c. Available cases are listed in the Case List widget.
4. Validate the Show Link to Case button.
 - a. Select a case in the Case List widget and click the Show Link to Case button.



Hint

If you click the Title link of a case, the case opens in the Case Details page. Select a case by clicking other columns.

- b. Verify that you get a dialog page with the direct link to the case.



5. Optionally, check the link.
 - a. Copy the URL and close the “Show Link” page.
 - b. Open a new browser tab.
 - c. Paste the URL into a new browser tab.
 - d. Verify that the Case Details page opens for the case that you selected.
 - e. Close the browser tab.
6. Validate the Open Web Page button.
 - a. Back in the Custom Toolbar Menu tab, click the Open Web Page button.
 - b. Verify that the WebSphere Integrated Solutions Console opens in a new browser tab.
7. Log out of the applications and close the browser windows.

Exercise 1.5.2: Add a custom action as a menu item

Introduction

In this exercise, you add a custom script action to a menu item on the default Case List widget. This script checks against the Case Title that you entered to see whether it matches with a case in the set of available cases.

Procedures

Procedure 1: Open the custom page, page. 1-56

Procedure 2: Edit the page to add the menu item, page. 1-56

Procedure 3: Test the menu item, page. 1-58

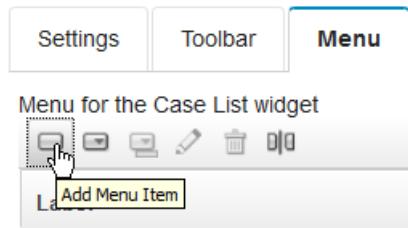
Procedure 1: Open the custom page

For this exercise, you reuse the custom page that you created in the previous exercise, and edit the page.

1. Refer to the steps in Procedure 1: Create a custom page, page. 1-51 to open the solution in Case Manager Builder.
 - a. If the page is not created earlier, create the page to use it in this exercise.
2. Open the Pages tab.
 - a. Expand the Solution Pages.
 - b. Click the Custom Toolbar Menu page to edit it in Page Designer.

Procedure 2: Edit the page to add the menu item

1. In Page Designer > Custom Toolbar Menu page, click the “Edit Settings” icon for the Case List widget in the middle pane.
2. Click the Menu tab.
 - a. Click the “Add Menu Item” button.



3. Select Script Action from the list for the Action field.

4. Edit the value for your label (Example: Flag Special Case).
5. Enter the following JavaScript for the execute section.

**Hint**

Optionally, copy the text from the C:\ICM\Menu Item Script\MenuItemScript.txt file and paste it.

```
var x = this.getActionContext("CaseReference");
var c = "LCS_AutoClaimsGeneral_000000100001";
if(c == x[0].getCaseTitle()){
    alert("This is a special case, please review this case.");
}
else{
    alert("This is a normal case, please proceed.");
}
```

**Note**

A case with the case title value "LCS_AutoClaimsGeneral_000000100001" is added to your system and used as an example in the code. You can optionally create a case and use the Title of your case.

How does this code work?

- This script checks against the Case Title to see whether it matches a case in the variable set (A list of available cases).
- You retrieve the value of a list of cases, as an array, from the Case List widget in the following line in this script: `this.getActionContext("CaseReference");`
- Then, you add an `if` statement to check against the first element of the case list array to see if it matches the name that is stored in the variable "c".

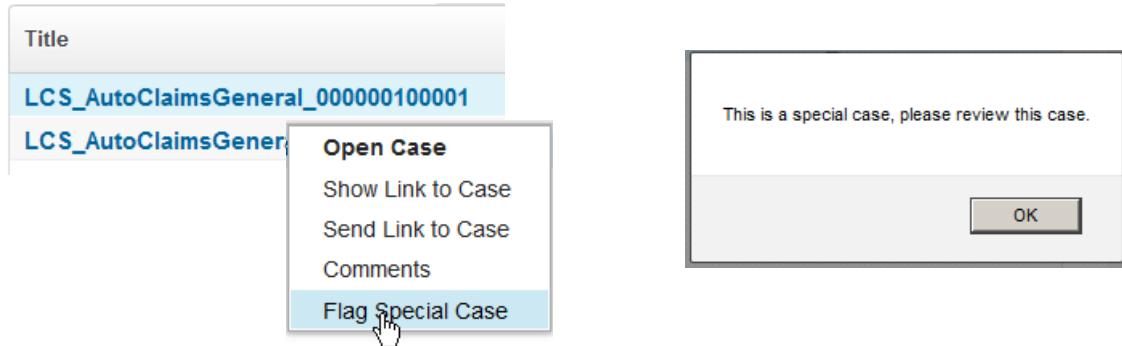
6. The OK button is not visible. Scroll down and click OK.
7. Verify that your Flag Special Case menu item is listed for the Case List widget.

Label	Action
Open Case	Open Case
Show Link to Case	Show Link to Case
Comments	Add Comment to Case
Flag Special Case	Script Action

8. Click OK to exit out of the Case List dialog window.
 - a. Click Save and then click Close at the upper right of the Page Designer window.
 - b. Click “Save and Close” on the solution to exit the solution editor.
9. Redeploy the solution and open Case Manager Client.
Refer to Procedure 4: Redeploy the solution, page. 1-54.

Procedure 3: Test the menu item

1. In Case Manager Client, click the Custom Toolbar Menu tab to open the custom page.
2. Search for cases.
 - a. In the Search section, select Policy Family name from the list.
 - b. Enter % in the next text box of the Search widget and click the Search button.
 - c. Available cases are listed in the Case List widget.
3. Validate your custom script action in the menu item.
 - a. Right-click the case with the Title value that you entered in the code (Policy Family Name; Smith).
 - b. Select Flag Special Case from the menu.
 - c. Verify that a dialog box is shown with the text that you added in the code: “This is a special case, please review this case.”



- d. Click OK to close the dialog box.
4. Verify that when you select another case, a different message is shown as you specified in the code.
 - a. Right-click the case with the Policy Family Name value “Beckner” and select Flag Special Case from the menu.
 - b. Validate that there is a different message.
 - c. Log out of the Case Manager Client and Case Manager Builder applications; Close the browser windows.

2

Use Scripts to Customize Case Manager Client

This unit provides guidance for customizing IBM Case Manager Client with the use of Script Adapter and scripts.

© Copyright IBM Corp. 2014

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

LESSON 2.1: IBM Case Manager JavaScript API Overview

What this lesson is about?

This lesson provides an overview of IBM Case Manager JavaScript API.

What you should be able to do?

After completing this lesson, you should be able to:

- Use IBM Case Manager JavaScript API.
- Use Script Adapter widget to customize the IBM Case Manager Case client.

How you will check your progress?

- Hands on labs.

Lab Solution files

- The solution files for this lesson are included in the following folder:
C:\ICM\Script Adapter Code

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

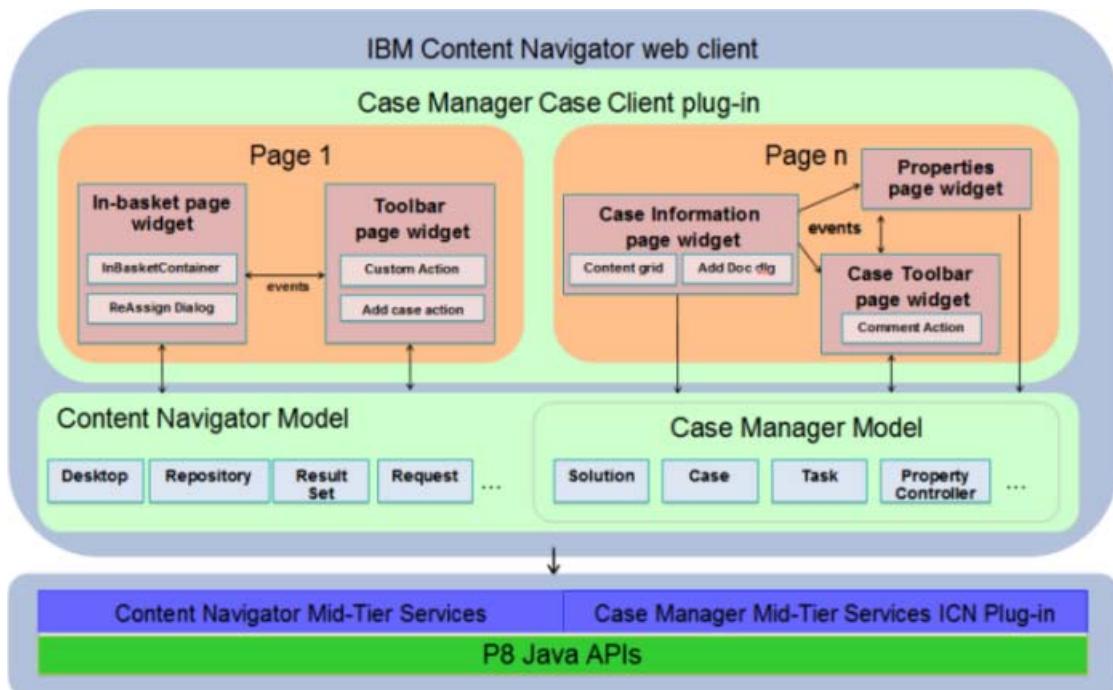
- <http://www.ibm.com/developerworks/library/>

IBM Case Manager Development Architecture

Architectural diagram

IBM Case Manager extends the IBM Content Navigator, and provides Case Manager model classes, page widgets, and utility classes (dialogs and actions).

It uses a Model-View-Controller (MVC) architecture.



- IBM Case Manager mid-tier services are built on IBM FileNet P8 Java APIs as an IBM Content Navigator plug-in.
- IBM Case Manager model layer consists of case management objects.
 - Example: Solution, Case, Task, and Property Controller objects.
- IBM Case Manager visual layer API provides classes that represent the following features:
 - Page widgets (example: Case Search, Case List, and In-basket)
 - Dialogs (example: Comments and Reassign)
 - Actions that are configured on toolbars and menus.

IBM Case Manager API toolkits

Overview of API toolkits

IBM Case Manager extends the IBM Content Navigator JavaScript Toolkit, and it provides the following sets of JavaScript APIs:

- IBM Content Navigator Model layer API
- IBM Content Navigator Visual widget API
- IBM Case Manager Model layer API
- IBM Case Manager Page widget API

Benefits of the APIs

- Enables custom widget developers to easily extend IBM Content Navigator and IBM Case Manager functionality.
 - Separation of business logic and User Interface (UI)
 - Model layer clearly partitions the business logic from the UI layer.
 - Shared model objects
 - Model objects that contain business/server side objects can be shared across the default and the custom widgets
 - Reusable components
 - Reuse visual widgets from the IBM Content Navigator widget library to build more complex custom UI widgets and dialogs
 - Reuse and extend IBM Case Manager page widgets for customizations.
-

IBM Content Navigator APIs

IBM Content Navigator JavaScript Toolkit summary

The following table lists the important IBM Content Navigator JavaScript Toolkit packages for model layer and visual widgets APIs. It also shows the important classes in the package, when to use and example scenarios.

Package	Example classes	When to use?	Example scenario
ecm.model Objects in the Content Server	Desktop Repository ContentItem WorkItem SearchTemplate SearchCriterion ResultSet	To access documents or folders (ContentItem objects) associated with a Case or a WorkItem. To retrieve information about the currently logged in user.	Use a script action to retrieve the document id of a document that is filed in a case.
ecm.widget Generic visual widgets	AddContentItemGeneralPane AddContentItemPropertiesPane AddContentItemSecurityPane ContentClassSelector	To build a custom dialog or page widget	Create a custom page widget that shows the selected document properties in the widget instead of in a dialog
ecm.widget.dialog Content operation dialogs Search dialogs Message dialogs	AddContentItemDialog CheckInDialog EditPropertiesDialog SearchBuilderDialog ErrorDialog StatusDialog	To create a custom dialog that extends an existing dialog with modifications	Create a custom Add Document dialog that allows you to select documents only from local file system.

IBM Content Navigator Model layer API

These APIs define the client-side model for IBM Content Navigator.

- The modeling classes do the following tasks:
 - Define communication with the mid-tier services
 - Provide the data server interaction APIs
 - Support caching.
- The data in the content servers is stored as model objects and used by the visual widgets.
 - The UI widgets interact with the content servers through the APIs on the model objects.

IBM Content Navigator Visual widget API

Visual widget library is based on Dojo and IDX. It provides generic User Interface widgets, panes, and dialogs for running content and workflow-related operations.

- Document and Folder-related widget components
- Workflow-related widget components
- User-related widget components
- A number of dialogs for document, search and workflow operations that can be used as is or extended.

 Note

For this unit, you use only model API for the Script Adapters.

IBM Case Manager JavaScript API

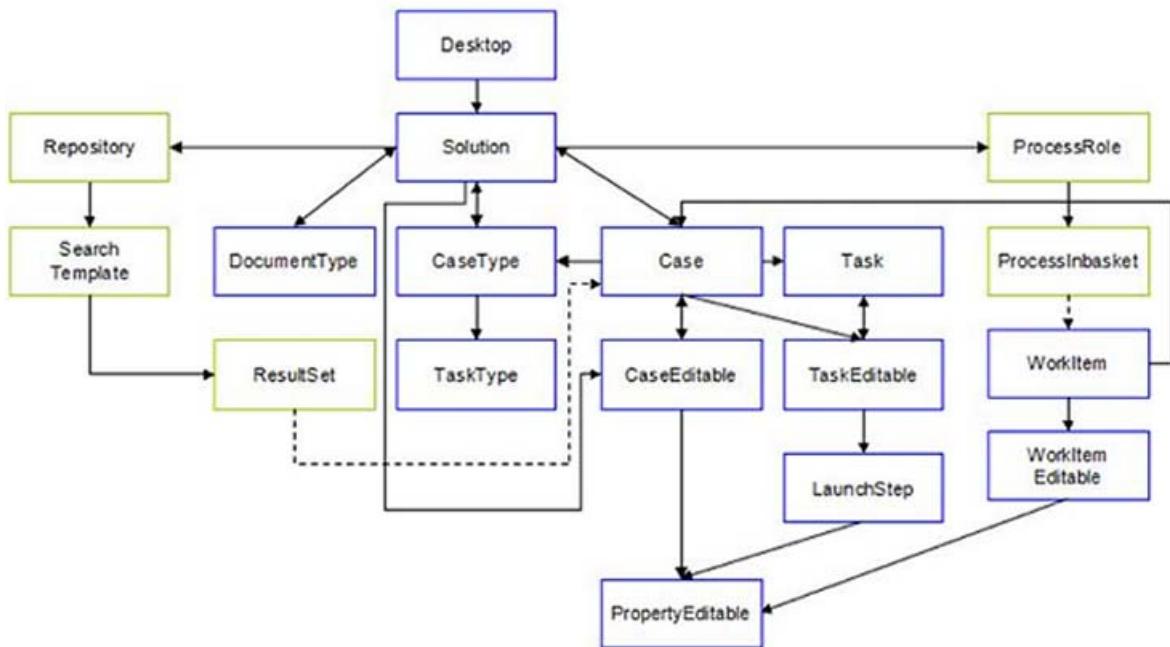
IBM Case Manager Model API

The Model API are JavaScript classes that are part of the larger Case Manager JavaScript Toolkit.

- They are non-UI components and represent Case Manager objects in the repository such as Solution, Case, and Task.
- They communicate with mid-tier services through a navigator plug-in.
- Similar to the navigator model API and services, the public API is the JavaScript Model API. The plug-in services are not public API.
 - Calling the plug-in services directly through HTTP by custom code is not supported.
 - The only supported use is a JavaScript client that calls the model classes.
- They extend and reuse some navigator model classes.
- They provide functions where it is not sufficient or possible to achieve through the navigator API alone.

IBM Case Manager Model layer objects

IBM Case Manager extends the IBM Content Navigator and you can retrieve information about the Solutions from the IBM Content Navigator Desktop object.



- The boxes that are represented in green are IBM Content Navigator objects.

IBM Case Manager Model API usage

The Model API can be used in the following scenarios.

- To develop a custom widget
 - Model object is passed as part of the payload of an event
- To use in a Script Adapter

IBM Case Manager Model API for Solutions

- Retrieve the deployed solutions
 - Solutions that are deployed to any target object store registered with the navigator desktop
- For a particular solution, retrieve
 - The case type or documents types
 - Information about the properties across all case types
 - The roles
 - The static pages to show for a particular role

IBM Case Manager Model API for a case type

- For a particular case type
 - Attributes include
 - What rights the user has cases of that case type
 - Whether dynamic (custom) tasks are supported
 - The default view definition to use when editing cases of this type
- Retrieve information about the properties of the case type
- Retrieve the discretionary task types that can be created
- Discover the particular page to use for a page type and role
- Search for dynamic tasks compatible with this case type (attached to cases of this type)

IBM Case Manager Model API for cases

- Retrieve a particular case or obtain a Case model object given a navigator ContentItem (for example from a folder-based search)
- Create a case
- Update the properties of a case
- Split a case

- Relate cases together
- Retrieve the cases that are related to a case (through splitting a case or generally relating cases together)
- Add and retrieve comments on cases
- Retrieve the case history
- Retrieve the tasks that are associated with a case
- Classes for the new case timeline history

IBM Case Manager Model API for tasks

- For task objects
 - Start a manual task
 - Enable and disable certain tasks
 - Stop and restart the workflow that is associated with a task
 - Create a discretionary task
- Dynamic (custom) tasks
 - Create a dynamic task
 - Update the data that describes the dynamic task and indicate if the data is valid (allowing the task to be started)
 - Start a dynamic task, causing dynamic workflow to be created and started

IBM Case Manager Model API for work items

Case Manager model shows a WorkItem object that encapsulates Case Manager specific behavior such as better integration with case properties in the API.

- Retrieve a work item
- Lock or unlock the work item
- Save or complete the work item
- Update the step parameters (properties) when saving or completing

IBM Case Manager JavaScript Toolkit summary

The following table lists the important IBM Case Manager JavaScript Toolkit packages for model layer and visual widgets APIs. It also shows the classes in the package, when to use a particular API, and example scenarios.

Package	Example classes	When to use?	Example scenario
icm.model Objects in the Case Manager system. No UI components	Case CaseComment CaseRelationship HistoryEvent PropertyController Solution Task	- To access ICM data when you create custom widgets and actions. These objects are used with Content Navigator model. - To access data for scripting events.	- Get the case identifier for the case displayed on case detail page. - Get editable model object. - Update a case property.
icm.action The default actions that are provided by Case Manager.	AddCustomTask SendLink ShowLink, AddDocumentfromLocal	To build a custom action: You can wrap an existing ICM action or create a new custom action.	Create a custom action for case document that publishes an event
icm.base Base classes that are used to create custom page widgets and actions	BasePageWidget BaseActionContext Constants WidgetAttributes _EventStub	When creating a custom widget or action, use these base classes to provide the infrastructure and fill in the implementation with custom behavior	Create a custom search widget Create an action to display data from an external system.
icm.dialog ICM-provided dialogs	AddCommentDialog, AddTaskDialog, DynamicTaskEditorDialog	To add a dialog to a custom widget or an action	Display dialog to add a comment when user clicks an icon in a content list
icm.pgwidget Classes that represent the page widgets	Attachment, CaseForm, CaseInfo, CaseList, CaseSearch, CaseToolbar, CaseVisualizer	To create a custom widget that includes a page widget To extend a page widget with more behavior	Extend the in-basket widget to change font to red for overdue items Add new tab in Case Info
icm.util Utility classes	Coordination, SearchPayload, In-basketFilterUtil	To participate in processing a page, such as dispatching a work item or saving a case. To build nested CE queries for flexible search.	Custom widget that saves data to an external system can hook into the dispatch of a work item
icm.widget.menu Toolbar and pop-up menu classes	ContextualMenu, Menu, MenuManager, Toolbar	To include a context menu or toolbar in a custom widget, and use the Page Layout Designer to enable configuring the menus	Implement retrieval of data from an external system with a dialog that is configured as a toolbar action in a custom widget

Collaborative editing of objects

IBM Case Manager editable objects

- Core objects in the API such as Case and WorkItem do not show scratchpad behavior.
 - State of the objects always reflects what was last persisted.
 - The behavior is similar to the Content Navigator objects such as ContentItem and ecm.model.WorkItem.
- Some Case Manager model objects show an editable object that provides scratchpad semantics.
 - Modify properties and other attributes of the object before saving.
 - Multiple widgets can share an Editable object. Each widget can apply its changes before saving.

Model objects with corresponding Editable objects

The following table lists the model objects and their corresponding editable objects. It also shows when these objects are used.

Model object	Corresponding Editable object	When is it used?
Case	CaseEditable	- To Modify case properties - To Create a new case.
WorkItem	WorkItemEditable	To modify step parameters when completing or saving a step
Task	TaskEditable / LaunchStep	Creating a discretionary task that is associated with a FileNet BPM workflow
<Property objects>	A collection of PropertyEditable objects	Represents the properties

Case object example for modifying the properties

```
//Get a copy of the caseEditable object for the current case.  
var caseEditable = caseObject.createEditable();  
//Create a property object mapped to "SOL_StringProp".  
var p = caseEditable.propertiesCollection["SOL_StringProp"];  
//Set the value of the property to "String Value".  
p.setValue("String value");  
//Save the modified property to the case.  
caseEditable.save(lang.hitch(this, this._saveComplete));
```

- When editable is saved, changes are reflected back on main non-editable.
 - Client can subscribe to change notification on editable or non-editable.
 - Client such as properties widget subscribes to change notifications at each property level
- Multiple editables can exist for a particular non-editable.
 - If any of them are saved, the non-editable and other editables are updated.
- Beside property value, other property attributes can change when External Data Service is involved.

Creating a new object

- A pending Editable object is obtained first.
- Main model object doesn't exist until pending Editable is saved.

```
solution.createNewCaseEditable(caseType,  
    lang.hitch(this, function(caseEdit){  
        var p = caseEdit.propertiesCollection[...];  
        p.setValue("String value");  
        caseEdit.save(lang.hitch(this, this._saveNewComplete));  
    }));  
    ...  
    _saveNewComplete: function(caseEdit) {  
        var caseObj = caseEdit.caseObject;  
    },
```

Script Adapter widget

Characteristics of Script Adapter

- A Script Adapter is a special widget that the user can place it on a page.
- The user enters JavaScript in the widget that runs when it is called.
- Script Adapters are wired to other widgets.
- Script Adapter widgets receive events from the other widgets to which they are wired.
 - Other widgets can receive events that are broadcast.

Script Adapter widget usage

You can use Script Adapter widget for the following tasks:

- To transform the data that one widget publishes into a different format for another widget.
 - Example: Transform the data that the Properties widget publishes into a format that a custom widget on a Case Details page can understand.
- To insert logic between widget event communication.
 - Example: Run a custom validation on data that the Properties widget publishes.
- Debug your solution with Script Adapter widget.
 - Configure the Script Adapter widget to display source event payload at run time for debugging.

How does the Script Adapter widget inserts logic?

- When the Script Adapter widget receives an event from a widget to which it is wired, it displays the event details in the Received Event section.
- The Script Adapter then runs a script that transforms the data as a function with a “payload” parameter.
- You can manipulate the incoming payload by implementing any type of logic in a script.
 - The value that your custom script returns is the payload of the outbound event of this Script Adapter widget.
 - The Sent Event section displays this information.
- Example: A Script Adapter receives a wired event with a payload of “test data” (a string value), which the Received Event section displays.
 - The Script Adapter has the following script:

```
alert("The value of the payload is: " + payload);
return "Event Payload: " + payload + "!";
```
 - The Sent Event section displays “Event Payload: test data!” as the payload for the outbound event.

- Notice that your script added an exclamation mark to the incoming string.



Hint

Optional: Hide the Script Adapter widget so that it is not visible to the user.

Debug the events with Script Adapter widget

You can use a Script Adapter to view event data to debug problems with wires between two widgets.

In Page Designer, edit the settings for the Script Adapter widget to select one or both of the following options:

- Show Script Text
 - Select this option to show the text of the script when the script runs.
- Block Outbound Event
 - Select this option to prevent the Script Adapter widget from sending the outgoing event.
 - If you wired the Send event payload event to another widget, you can select this option to temporarily stop the Script Adapter widget from sending the event while you are debugging the script.



Important

Your script must contain basic JavaScript only and must be viewed as the body of one single function. You can use the alert() statement to display information about the values in the script. You cannot use Dojo.

What is a payload?

- A payload is an object that is passed from, or received by, a widget when an event happens in the system.
 - Most payloads are JSON objects.
 - A payload can be a simple string.
- When you configure a Script Adapter for the first time:
 - You can create the event wiring with the other widget, and display the payload that is received by the Script Adapter to verify what is in the payload.
 - To display the payload use the following lines of code:

```
if(payload)  
alert("payload= " ,JSON.stringify(payload));
```

Exercise 2.1.1: Use Script Adapter to customize the Case client

Introduction

In this lab, you show the number of items (documents and folders) that are attached to a case by using the Script Adapter widget and wiring it to the Case List widget on the Cases page.

This is an example of the use of Script Adapter. Many examples are presented in this unit.

Procedures

Procedure 1: Start WebSphere Application Server, page. 2-15

Procedure 2: Create a custom page, page. 2-15

Procedure 3: Edit the page to add the Script Adapter, page. 2-16

Procedure 4: Assign the custom page to a role, page. 2-19

Procedure 5: Redeploy the solution, page. 2-20

Procedure 6: Test the Script Adapter, page. 2-20

Procedure 1: Start WebSphere Application Server

1. If it is not already started, start the WebSphere Application Server.
 - a. Click Start > All Programs > IBM WebSphere > IBM WebSphere Application Server V8.5 > Profiles > AppSrv01 > Start the server.
 - You can also use the Start_Server1.bat file in the WebSphere Admin folder on the desktop.
2. Wait for the Start the server page to close.



Note

For more information about “Start and stop System Components”, see the Appendix at the end of book.

Procedure 2: Create a custom page

A solution for this lab is already created. In this procedure, you copy the default Cases page and modify it.

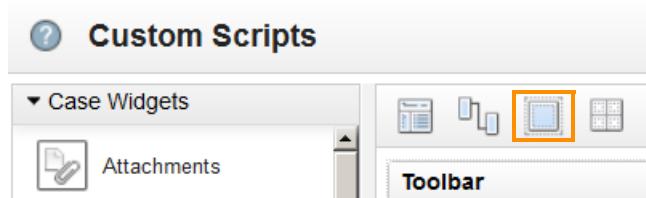
1. In Firefox, log on to IBM Case Manager Builder as an administrative user.
 - URL: <http://ecmedu01:9080/CaseBuilder>
 - User name: P8admin
 - Password: IBMFileNetP8

2. In Case Manager Builder, open Lab Claims Solution by clicking the Edit link.
Hover the mouse over the solution to see the links.
3. Create a custom page:
 - a. Open the Pages tab and expand the Solution Pages.
 - b. Click and hover the mouse over the Cases page name.
 - c. Select the Copy icon on the right side of the page.
 - d. In the resulting page, edit the name to Custom Scripts for your new page and click OK to create the copy.
 - e. Save your work by clicking Save at the top of the page.
4. Leave the Pages tab opened for the next procedure.

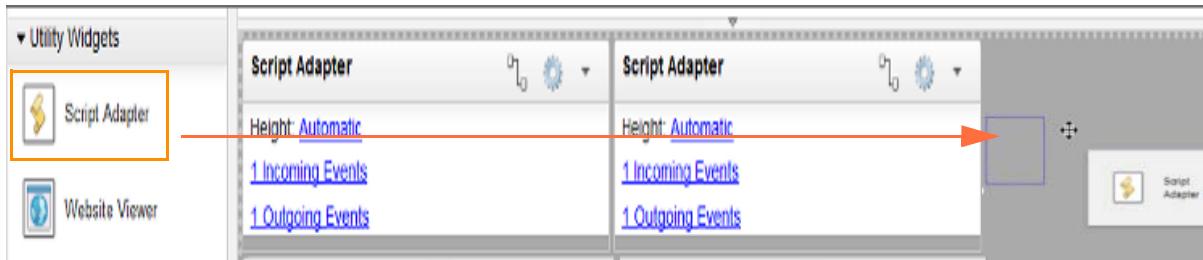
Procedure 3: Edit the page to add the Script Adapter

In this procedure, you edit the page to add the Script Adapter.

1. In the Pages tab, click your custom page (Custom Scripts) to edit it in Page Designer.
2. Click the “Show or Hide Hidden Widgets” button.
 - a. Notice that a gray section appears on the bottom of the main layout area.
 - b. The section already contains two Script Adapters.

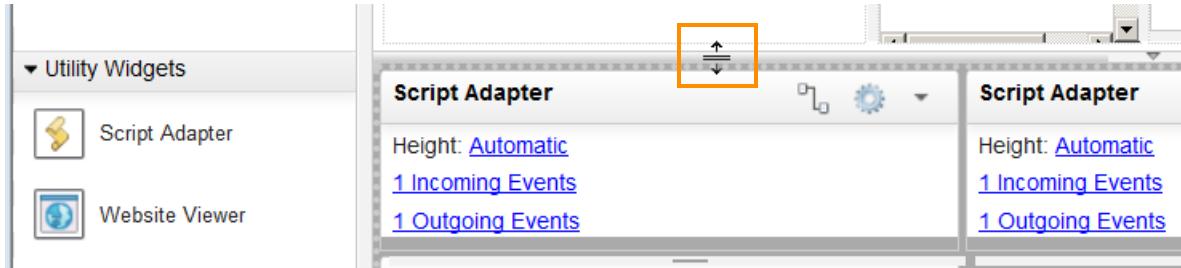


3. Drag and drop the Script Adapter widget (under the “Utility Widgets”) from widget palette on the left column to the bottom of the page on the right.



- a. The location of the widget does not matter.
- b. If your Script Adapter is not visible, scroll down the page (it might be below the existing Script Adapters).

- c. You can also expand the Script Adapter area with the control that is shown in the screen capture.



4. Rename the widget:

- a. Click the down-arrow, and select “Rename Widget”.

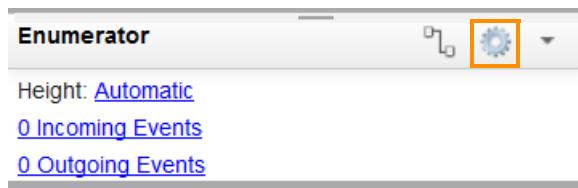


- b. In the “Rename Widget” dialog page, edit the Widget name as Enumerator.

Naming the Script Adapters helps to differentiate your widget.

- c. Click OK.

5. Click the “Edit Settings” icon of the Script Adapter widget.



- a. In the Script Adapter dialog window that opens, clear the text in the JavaScript text box.

6. Enter the following lines of code into the JavaScript text box.

```
var caseEditable = payload.caseEditable;
var caseObj = caseEditable.getCase();
var caseFolder = caseObj.caseFolder;
var actionsHandler = ecm.model.desktop.getActionsHandler();
if (actionsHandler) {
    actionsHandler.actionOpen(caseFolder, function(caseFolder,resultSet) {
```

```
        alert("The selected case folder has " + resultSet.items.length + " items.");
        for (var i = 0 ; i < resultSet.items.length; i++) {
            console.log("Document ID : " + resultSet.items[i].id + "\n" +
            "Document Title: " + resultSet.items[i].name);
            alert("Document ID : " + resultSet.items[i].id + "\n" + "Document
Title: " + resultSet.items[i].name );
        }
    });
}
return payload;
```

**Hint**

Copying the text from the PDF file might have hidden characters. Copy the text from the C:\ICM\Script Adapter Code\EnumerateDocs.txt file and paste it.

- a. Click OK to close the Script Adapter dialog page.

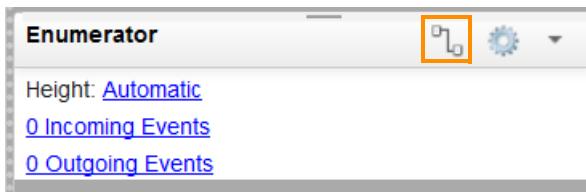
**Troubleshooting**

If you have any errors in the JavaScript, the OK button is disabled. Also, it shows an error message. Verify the code that you entered. If needed, clear the code, copy, and paste the code again.

How does the code works?

- The script gets case editable model objects from the payload.
- Calls the Content Navigator function ecm.model.desktop.getActionsHandler() to get the ActionsHandler for case folder to retrieve the contents of the folder.
- In a for loop, you then retrieve the document ID and document title.

7. Click the Edit Wiring icon for the Script Adapter widget.



8. Wire the Script Adapter to the Case List widget to receive the Select Case outgoing event.
 - a. In the “Wire Events” page > “Incoming Events for Enumerator” section, select the following values from the list for each field:

Field	Value
Source widget	Case List
Outgoing event	Select Case
Incoming event	Receive event payload

- b. Click Add Wire.
- c. Validate that the completed wiring looks like the following screen capture.



9. Click Save to save your work.
 - a. Scroll down and click OK at the bottom of the Wire Events dialog window.
 - b. Click Save and then Close to close Page Designer.
 - c. Leave the solution open for the next procedure.

Procedure 4: Assign the custom page to a role

1. Your solution is already opened in the Case Manager Builder. Open the Roles tab.
2. Click the Customer Service Rep role link.
 - a. Open the Pages subtab.
3. Optionally, remove the Custom Toolbar page that you created in the previous unit.
 - a. Select the page, hover over, and click the Remove (trash can) icon.



Note

Step 3, removes only the association of the page with a role. The actual page (Custom Toolbar) that you created is still available if you need to use it. The removal step helps to minimize the number of tabs in the Case Manager Client.

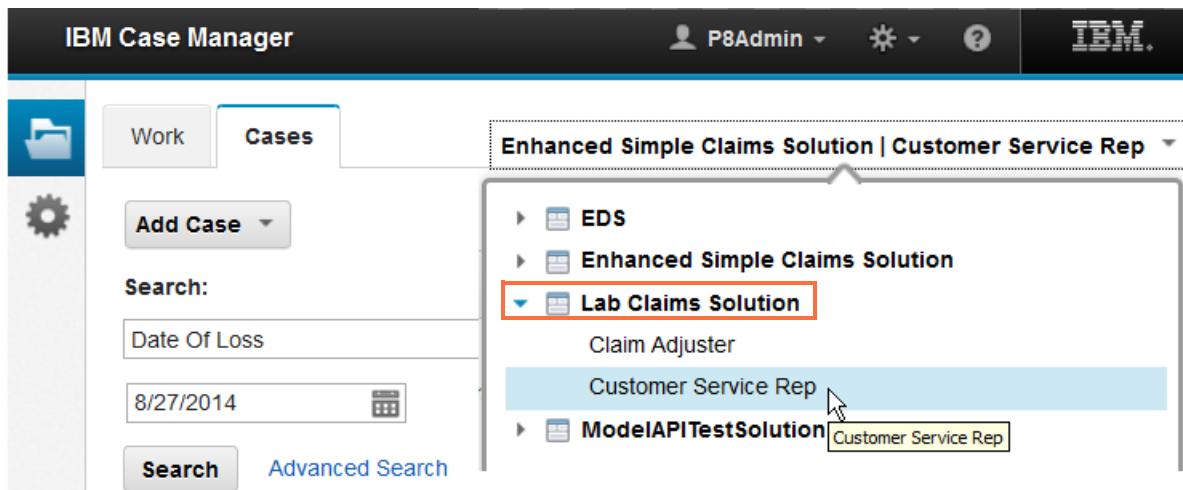
4. Assign the new page that you created.
 - a. Click Assign Page.
 - b. Select the Custom Scripts Page.
 - c. Click OK to close the dialog window.
5. Verify that your page is listed in the Pages tab.
 - a. Click OK.
 - b. Click OK All to accept the changes to the role.
 - c. Click “Save and Close” at the top of the page to exit the solution editor.
 - d. Leave the Case Manager Builder open for the next procedure.

Procedure 5: Redeploy the solution

1. In the Manage Solutions page, select Lab Claims Solution and hover the mouse over.
 - a. Click Deploy.
 - b. In the Confirmation page, make sure that “Commit my changes and make them available for deployment” is selected, and click Deploy.
 - c. Wait for the green check mark to appear next to the solution.
2. Select Lab Claims Solution and hover the mouse over again.
 - a. Click Test to open Case Manager Client.

Procedure 6: Test the Script Adapter

1. In the Case Manager Client, if it is not already selected, select Lab Claims Solution > Customer Service Rep from the upper right of the page from “Solution and Roles” list.



2. Select the Custom Scripts tab to open your custom page.

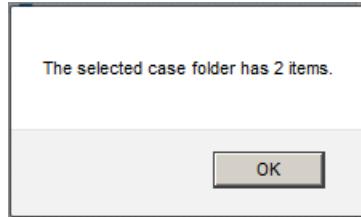
3. Search for cases.
 - a. In the Search section, select Policy Family name from the list.
 - b. Enter % in the next text box of the Search widget and click the Search button.
 - c. Available cases are listed in the Case List widget.
4. Validate the Script Adapter widget.
 - a. Select the case with the "Policy Family Name": Beckner in the Case List widget.



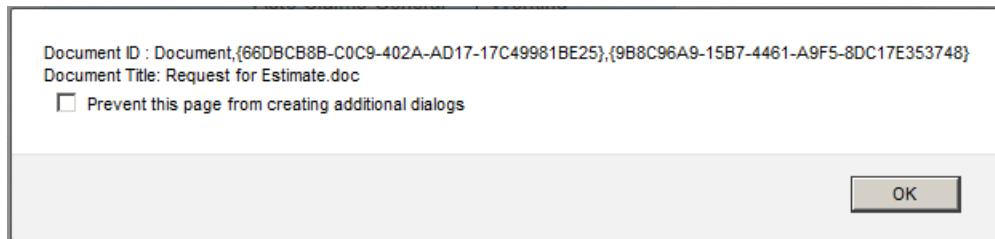
Hint

If you click the Title link of a case, the case opens in the Case Details page. Select a case by clicking other columns.

- b. Verify that you get a message with number of items in that case folder. Click OK.



- c. You get another message with the Document ID, Title as you specified in the code.
- d. The code loops through the result set for each item. When you click OK on the message, if there are more items, you get more messages.



- e. Close all the dialog windows.
- f. Log out of the Case Manager Builder and the client, and close the browsers.

LESSON 2.2: Create a script action to start a task

What this lesson is about?

This lesson describes how to create a toolbar item to start a specific discretionary task.

What you should be able to do?

After completing this lesson, you should be able to:

- Start a specific discretionary task from a toolbar button.

How you will check your progress?

- Hands on labs.

Lab Solution files

- The solution files for this lesson are included in the following folder:
C:\ICM\Script Adapter Code

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>
-

Adding discretionary tasks

Discretionary tasks

- Discretionary tasks are defined for specific types of cases.
 - These tasks allow caseworker to add more tasks to a case at the run time.
 - They are used for situations that happen frequently, but not as part of every case.
- Example:
 - A discretionary task might be used to initiate a fraud investigation for an insurance claim.
 - The work items that are required to complete the investigation are predefined in the solution for the discretionary task.

Start a discretionary task programmatically

- In the IBM Case Manager Client user interface, You do the following steps to start a task.
 1. From the Case or Work Details page, select the Tasks tab.
 2. Click the Add Task button to show the dialog to select the task.
 3. Select one of the discretionary tasks for a case from a list.
 4. Enter a unique name for the task.
 5. Click OK to get to the Add task page.
- For discretionary tasks that are commonly added to a case, you can minimize the steps that a user must do.
 - Example: Add a toolbar button to a case details page with Script Action.

Tips for the labs

Firefox browser

- In this course, the Firefox browser is used to work with IBM Case Manager.
 - If you use other browsers, you must take into account that they might interpret the JavaScript differently.
- Firebug is a helpful debugging tool when you are modifying the JavaScript.
 - It runs only in Firefox.
 - You must use a compatible debugging tool for other browsers.

Create new pages for customization

- When you customize a page, create a page by copying a working page.
 - You then modify the new page.
- The new page must be associated with a role or a workflow step or a case for it to display in Case Manager Client. The following table shows the type of pages and their association.

Type of page	Associate the page to a
Solution Pages	Role
Work Details	Specific activity steps
Case Details	Case Type

- This approach allows you to revert to the default configuration with minimal effort.

Script Adapter naming

- Since the default labels for all the Script Adapter widgets are the same, rename your Script Adapter to a unique name.
 - Optionally, append SA (or any other label) to the end of your Script Adapter name to make it easier to find it in the console log.
-

IBM Case Manager classes and events that are used in this lesson

AddTask event

This event opens an Add Task page so that the user can add a task.

- Event ID
 - icm.AddTask
- Payload
 - taskEditable
 - An `icm.model.TaskEditable` object that represents the task that is added.
 - caseEditable
 - An `icm.model.CaseEditable` object that represents the case to which the task is added.
 - coordination
 - An `icm.base.Coordination` object that is used internally by the widgets in the same page.

IBM Case Manager JavaScript classes

- `icm.base.Coordination`
 - This class represents an object that is used to coordinate the communication among widgets that are on a page.
- `icm.model.CaseEditable`
 - This class represents a case.
 - To obtain a `CaseEditable` object to create a case, call the `createNewCaseEditable` method on the `Solution` object.
 - To obtain a `CaseEditable` object to edit an existing case, call the `createEditable` method on the `Case` object.
- `icm.model.TaskEditable`
 - This class represents a task.
 - `TaskEditable` objects are primarily used to represent new discretionary tasks.
 - To obtain a `TaskEditable` object for a discretionary task, call the `createNewTaskEditable` method on the `Case` object.
 - To obtain a `TaskEditable` object to edit an existing task, call the `createEditable` method on the `Task` object.
 - For discretionary tasks, the model includes the `LaunchStep` class that represents the launch step of a workflow. The `propertiesCollection` attribute of this class provides a collection of `PropertyEditable` objects. Each `PropertyEditable` object represents a parameter for the launch step.

Exercise 2.2.1: Create a toolbar button to start a task

Introduction

In this lab, you add a toolbar button with a script to start the Add Task page in a single click.

Procedures

Procedure 1: Check the default method to start a discretionary task, page. 2-26

Procedure 2: Create a custom page to add a script action to start a task, page. 2-27

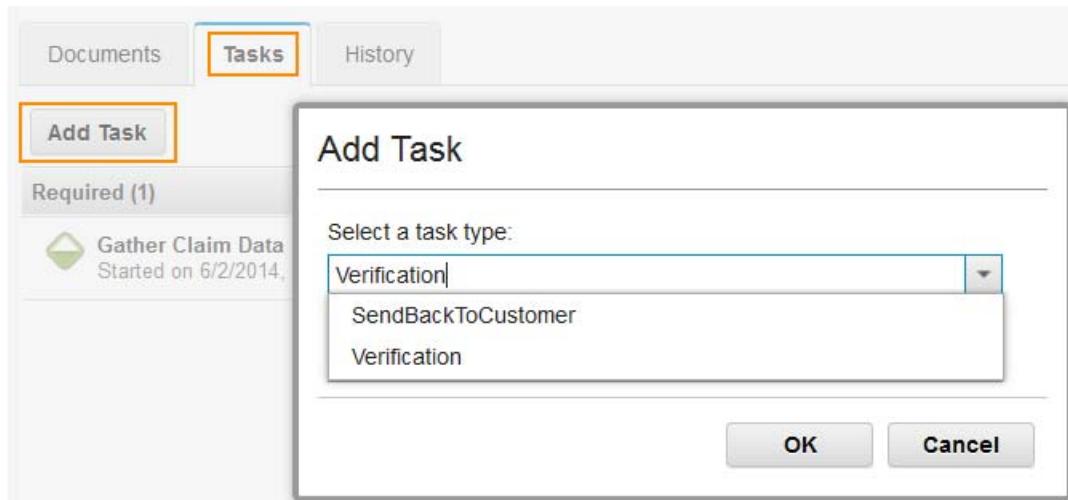
Procedure 3: Test the custom script action to add a task, page. 2-30

Procedure 1: Check the default method to start a discretionary task

The Lab Claim Solution contains discretionary tasks required for this lab exercise.

1. Log in to the Case Manager Client in a Firefox browser.
 - URL: <http://ecmedu01:9080/navigator/?desktop=icm>
 - User name: P8admin
 - Password: IBMFileNetP8
2. In the Lab Claims Solution > Customer Service Rep role, select the Cases tab.
3. Search for cases.
 - a. In the Search section, select Policy Family name from the list.
 - b. Enter % in the next text box of the Search widget and click the Search button.
 - c. Available cases are listed in the Case List widget.
4. Open case details for a case.
 - a. Click the Title link of the case with the “Policy Family Name”: Smith in the Case List widget.

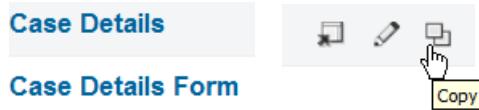
The case opens in the Case Details page.
5. Start a discretionary task.
 - a. Select the Tasks subtab.
 - b. Click “Add Task”.
 - c. For the “Select a task type” field, observe that you can have several tasks in the list from which you choose a task.
6. Optionally, select a task (Example: Verification) and click OK.



7. The “Add Task” tab opens. If there is a required field (Example: estimate), enter a value.
 - a. Click Add.
 - b. Back in the “Case Details” tab, click Save and then Close.
 - c. The work item from your task is listed in the Work tab > CSR Tasks subtab. Other work items are also listed. These were created when cases were added.
8. Log out of the Case Manager Client and close the browser.

Procedure 2: Create a custom page to add a script action to start a task

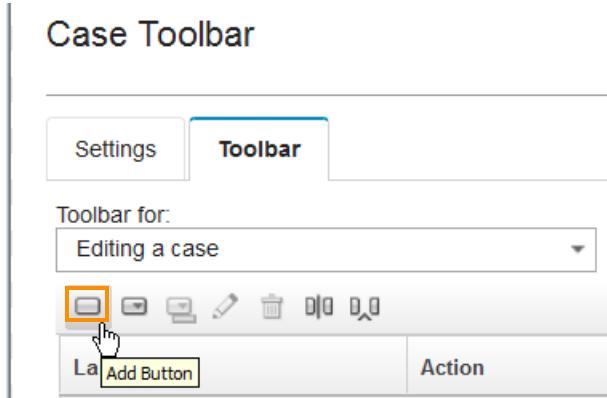
1. In Firefox, log on to IBM Case Manager Builder as an administrative user.
 - URL: <http://ecmedu01:9080/CaseBuilder>
 - User name: P8Admin
 - Password: IBMFileNetP8
2. In Case Manager Builder, open Lab Claims Solution by clicking the Edit link. Hover the mouse over the solution to see the links.
3. Create a custom page:
 - a. Open the Pages tab and expand the Case Details Pages.
 - b. Hover the mouse over the Case Details page name.
 - c. Select the Copy icon on the right side of the page.



- d. In the resulting page, edit the name to **Script Case Details** for your new page and click **OK** to create the copy.
- e. Save your work by clicking **Save** at the top of the page.
4. Edit the page in Page Designer to add the custom widget.
 - a. In the Pages tab, double-click **Script Case Details**.
The **Script Case Details** page opens in Page Designer.
5. In the **Script Case Details** page, add a **Script Action** toolbar button.
 - a. Click the “Edit Settings” icon for the **Case Toolbar** widget at the top of the page.



- b. Select the **Toolbar** tab and click the “Add Button” icon.



6. For **Action**, select **Script Action** from the list.
7. Edit the value for **Label** with the name of your discretionary task. (Example: **Verification**).

Action:	Script Action
Alignment:	Left
Label:	Verification
Execute:	<pre>var caseEditable = this.getActionContext("Case")[0]; var solution = caseEditable.getCase().caseType.getSolution();</pre>

8. Enter the following JavaScript in the Execute input field.



Hint

Optionally, copy the text from the C:\ICM\Script Adapter Code\StartDiscretionaryTaskFromToolBarButton.txt file and paste it.

```
var caseEditable = this.getActionContext("Case")[0];
var solution = caseEditable.getCase().caseType.getSolution();
var prefix = solution.getPrefix();
var theCaseID = caseEditable .getCaseTitle();

/* Must match the task type name in your solution that appears in the task
list of the Add Task page */
var taskType = prefix + "_Verification";

callback=function(taskEditable){

/* Provide a task name that shows in the list after you add the task */
taskEditable.setTaskName("VerifyDocs " + theCaseID);

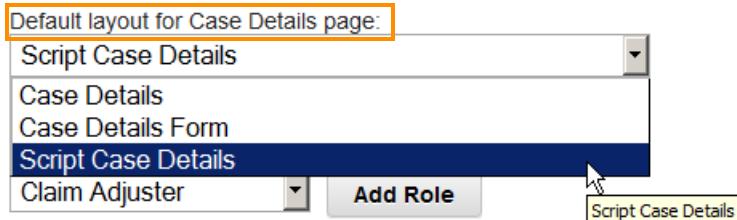
var addTaskPagePayload = {
    "taskEditable": taskEditable,
    "coordination": new icm.util.Coordination()
};
this.broadcastEvent("icm.AddTask", addTaskPagePayload);
};

caseEditable.getCase().createNewTaskEditable(taskType,
dojo.hitch(this,callback));
```

How does the script work?

- Get a CaseEditable object for the case, to which the task is added, from the action context.
- Create an instance of a TaskEditable class for your task type.
- Create a payload that contains taskEditable and the coordination objects.
- Broadcast the AddTask event.
- The Page Container widget receives the AddTask event with the payload.
 - It handles the event to open an Add Task page so that the user can add a task.

9. The OK button for the action is not visible. Scroll down and click OK.
10. Verify that your Verification item is listed for the Case Toolbar.
11. Click OK in the Case Toolbar page.
12. Click Save to save your changes and Close to close Page Designer.
13. Assign the new page to a case type.
 - a. Select the Case Types tab.
 - b. Select Auto Claims General.
 - c. In the Case Type page, select the “Script Case Details” page for “Default layout for Case Details page”.

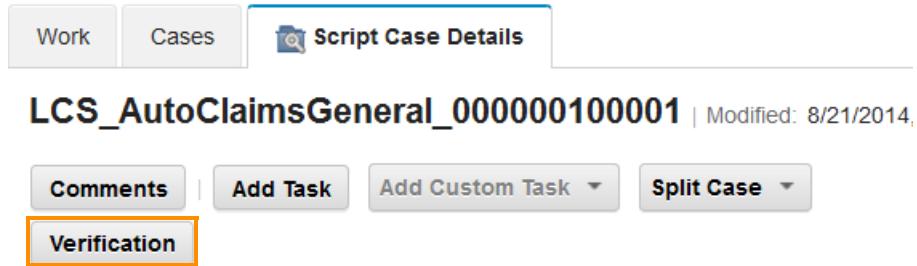


14. Click Save and Close to save the changes to the solution.
15. Redeploy the solution.
 - a. In the Manage Solutions page, select Lab Claims Solution and hover the mouse over.
 - b. Click Deploy.
 - c. In the Confirmation page, make sure that “Commit my changes and make them available for deployment” is selected, and click Deploy.
 - d. Wait for the green check mark to appear next to the solution.
16. Select Lab Claims Solution and hover the mouse over again.
 - a. Click Test to open Case Manager Client.

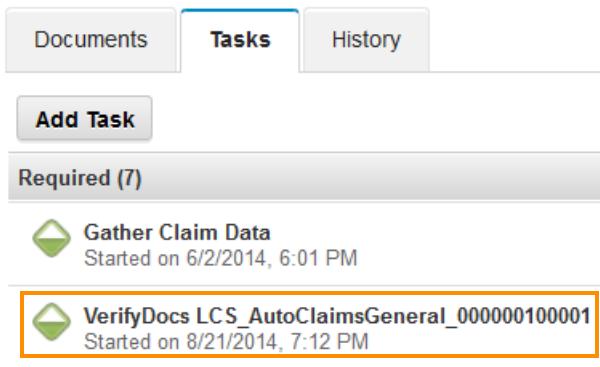
Procedure 3: Test the custom script action to add a task

1. The Case Manager Client is opened.
 - a. In the Lab Claims Solution > Customer Service Rep role, select the Cases tab.
2. Search for cases.
 - a. In the Search section, select Policy Family name from the list.
 - b. Enter % in the next text box of the Search widget and click the Search button.
 - c. Available cases are listed in the Case List widget.

3. Open case details for a case.
 - a. Click the Title link of the case with the “Policy Family Name”: Smith in the Case List widget in the right pane.
The case opens in the Script Case Details page.
4. In the Script Case Details tab, verify that the toolbar button (Verification) that you added is shown at the top of the page.



5. Click Verification. The Add Task page is opened.
You are able to access the user interface to add a discretionary (Verification) task with a single click.
 - a. Enter a value for the Estimate field (Example: 123). It is a required field.
6. Click Add to add the task.
 - a. Back in the Script Case Details page, the task that you added is listed in the Tasks subtab.
 - b. To see the complete list of tasks, collapse the Timeline Visualizer widget.
 - c. Check that the name contains the string that you entered in the script (VerifyDocs) and the case name.



7. Log out of the applications and close the browser windows.

LESSON 2.3: Add a choice list to a case property

What this lesson is about?

This lesson describes how to dynamically add a choice list to a case property with a Script Adapter.

What you should be able to do?

After completing this lesson, you should be able to:

- Add a choice list to a case property with a Script Adapter.

How you will check your progress?

- Hands on labs.

Lab Solution files

- The solution files for this lesson are included in the following folder:
C:\ICM\Script Adapter Code

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Adding a choice list to a case property

Adding a choice list dynamically

- The choice lists provide preconfigured values for the data fields so that users can avoid any errors from typing the data.
- IBM Case Manager provides the following methods to associate a choice list with a case property.
 - Assign a choice list in IBM Case Manager Builder when you design a Case Solution.
 - Implement IBM Case Manager External Data Services (EDS) to assign it during run time.



Note

The EDS topic is covered in the “Implement External Data Services” unit.

- Use Client-side scripts with JavaScript Model APIs.



Note

This lesson describes the client-side scripting method.

IBM Case Manager classes that are used in this lesson

taskEditable.getLaunchStep()

- `taskEditable.getLaunchStep()`
 - An `icm.model.TaskEditable` object that represents the task that is created.
 - The editable properties are contained in the `icm.model.LaunchStep` object that is associated with the `icm.model.TaskEditable` object.
 - To obtain the editable properties, call the `taskEditable.getLaunchStep()` method.

Property Controller

- The Property Controller classes are provided under the `icm.model.properties.controller` package.
- These classes provide the controller for the view that is rendered in the Properties Widget.
- They provide a mechanism for binding and unbinding the controller to a case model object.
 - You can use these classes to customize the functionality of the properties view.

Constants.CoordTopic.LOADWIDGET

- `icm.base.Constants` is helper class that defines the constants for IBM Case Manager.
 - Constants are divided into different sub categories.
 - The `CoordTopic` field represents the constants for coordination topics.
 - `LOADWIDGET` is the topic for a coordination step for page widgets to indicate that it is in ready state.
 - It means that all initialization is completed.
-

Exercise 2.3.1: Dynamically add a choice list to a case property

Introduction

In this lab exercise, you add a choice list to a case property that is shown in the Properties widget on an Add Task page. You use the discretionary task that you added in the previous exercise.

Procedures

Procedure 1: Check the default behavior for a step response, page. 2-44

Procedure 2: Customize the Add Task page, page. 2-36

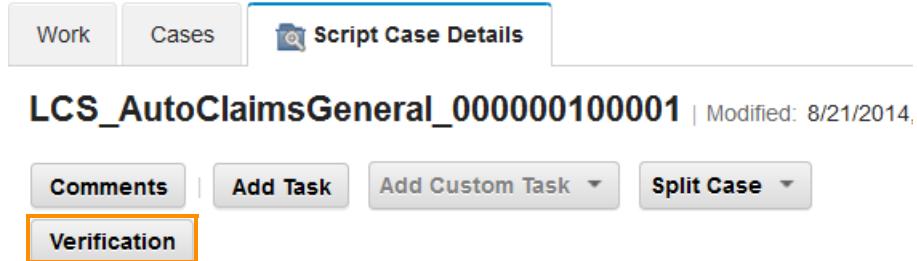
Procedure 3: Redeploy the solution, page. 2-40

Procedure 4: Test the choice list, page. 2-41

Procedure 1: Check the default behavior for a step response

The Lab Claim Solution contains discretionary tasks that are required for this lab exercise.

1. Log in to the Case Manager Client in a Firefox browser.
 - URL: <http://ecmedu01:9080/navigator/?desktop=icm>
 - User name: P8admin
 - Password: IBMFileNetP8
2. In the Lab Claims Solution > Customer Service Rep role, select the Cases tab.
3. Search for cases.
 - a. In the Search section, select Policy Family name from the list.
 - b. Enter % in the next text box of the Search widget and click the Search button.
 - c. Available cases are listed in the Case List widget.
4. Open case details for a case.
 - a. Click the Title link of the case with the “Policy Family Name”: Smith in the Case List widget in the right pane.
The case opens in the Script Case Details page.
5. In the Script Case Details tab, click the Verification toolbar button at the top of the page.
The Add Task page is opened.



6. Check that the “Case Stage” property does not have a choice list that is attached to it.

VerifyDocs LCS_AutoClaimsGeneral_000000100001 |

The screenshot shows the 'VerifyDocs' page for the same case. On the left, there's a 'Case Stage' field which is highlighted with an orange box. Below it is a field for 'Estimate' containing '1,237'. On the right, there's a section titled 'Attachments' with a message 'No attachments to display.' and three buttons: 'Add', 'Open', and 'Actions'.

7. For this procedure, it is not necessary to complete the task. Click Cancel.
 - a. Log out of the client.

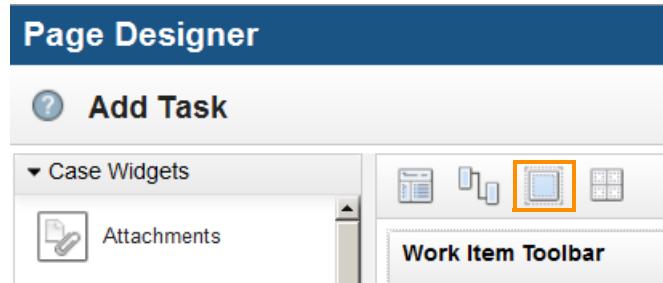
 Note —————
In the following procedures, you add associate a choice list to the “Case Stage” field.

Procedure 2: Customize the Add Task page

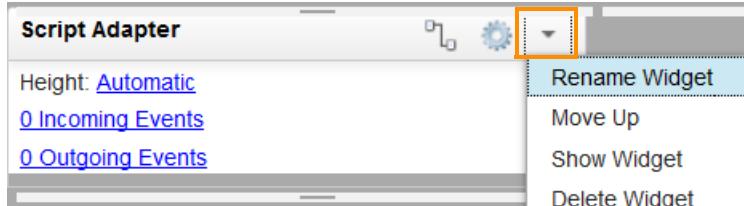
In this procedure, you edit the Add Task page to add the Script Adapter.

1. In Firefox, log on to IBM Case Manager Builder as an administrative user.
 - URL: <http://ecmedu01:9080/CaseBuilder>
 - User name: P8Admin
 - Password: IBMFileNetP8
2. In Case Manager Builder, open Lab Claims Solution by clicking the Edit link. Hover the mouse over the solution to see the links.

3. Edit the “Add Task” page in Page Designer:
 - a. Open the Pages tab and expand the Add Task Pages.
 - b. Click Add Task.
4. In Page Designer > Add Task page, add a Script Adapter widget.
 - a. Click the “Show or Hide Hidden Widgets” button.
A gray section appears on the bottom of the main layout area.



- b. Drag the Script Adapter widget (under the “Utility Widgets”) from widget palette on the left column to the bottom of the page on the right.
5. Rename the widget:
 - a. Click the down-arrow, and select “Rename Widget”.



- b. In the “Rename Widget” dialog page, edit the Widget name as ChoiceList.
 - c. Click OK.
6. Click the “Edit Settings” icon of the Script Adapter widget.



- a. In the Script Adapter dialog window, clear the text in the JavaScript text box.

7. Enter the following lines of code into the JavaScript text box.



Hint

Optionally, copy the text from the C:\ICM\Script Adapter Code\DynamicallyAddChoicesTask.txt file and paste it.

```
/* Specify the property to associate a choice list */
var theProp = this.solution.getPrefix() + "_CaseStage";
console.log("Got the property: " + theProp);
/* From the add task payload, retrieve the workItemEditable with the
getLaunchStep method. */
var workItemEdt = payload.taskEditable.getLaunchStep();
var coord = payload.coordination;
/* Create an array of choices and assign them to a variable */
var choices = [
    {
        label: "Assignment",
        value: "1"
    },
    {
        label: "In Process",
        value: "2"
    },
    {
        label: "Approval",
        value: "3"
    }
];
/* Load the controller and constants */
require(["icm/model/properties/controller/ControllerManager",
"icm/base/Constants"],function(ControllerManager, Constants){
    /* Use the LOADWIDGET coordination topic handler to bind the controller to
the editable.
Get the property and set the choices */
    coord.participate(Constants.CoordTopic.LOADWIDGET, function(context,
complete, abort){
        var propsController = ControllerManager.bind(workItemEdt);
        var propController = propsController.getPropertyController(theProp);
        if (propController){
            propController.set("choices", choices);
        }
    })
});
```

```
        else
        /* If the specified property does not exist, show an error */
        console.log("ERROR - No such property: " + theProp);
        complete();
    });

});
```



Troubleshooting

If you have any errors in the JavaScript, the OK button is disabled. Also, it shows an error message. Verify the code that you entered. If needed, clear the code, copy, and paste the code again.



Note

You can also assign choice lists to workflow data fields using this method. Use the data field name without the solution prefix.

How does the script works?

- Specify the property to associate a choice list.
- From the add task payload, retrieve the workItemEditable with the getLaunchStep method.
- Create an array of choices and assign them to a variable.
- Load the controller and constants.
- Use the LOADWIDGET coordination topic handler to bind the controller to the editable.
- Get the property and set the choices
 - If the specified property does not exist, show an error.

-
8. Click OK to close the Script Adapter dialog page.
 9. Click the Edit Wiring icon for the Script Adapter widget.



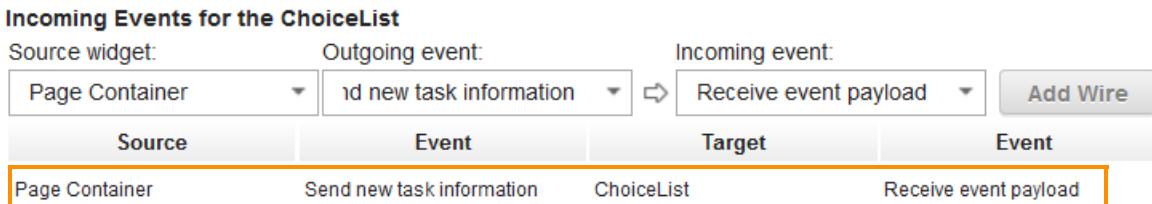
10. Wire Script Adapter to Page Container to receive the “Send work Item” outgoing event.

- In the “Wire Events” page > “Incoming Events for ChoiceList” section, select the following values from the list for each field:

Field	Value
Source widget	Page Container
Outgoing event	Send new task information
Incoming event	Receive event payload

- Click Add Wire.

- Validate that the completed wiring looks like the following screen capture.



11. Click Save to save your work.

- Scroll down and click OK at the bottom of the Wire Events dialog window.
- Click Save and then Close to close Page Designer.
- Click “Save and Close” at the top of the page to exit the solution editor.
- Leave the Case Manager Builder open for the next procedure.

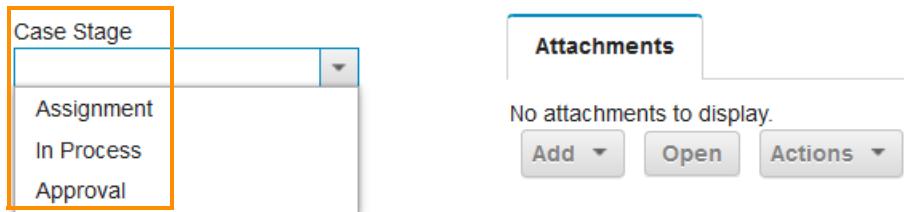
Procedure 3: Redeploy the solution

- In the Manage Solutions page, select Lab Claims Solution and hover the mouse over.
 - Click Deploy.
 - In the Confirmation page, make sure that “Commit my changes and make them available for deployment” is selected, and click Deploy.
 - Wait for the green check mark to appear next to the solution.
- Select Lab Claims Solution and hover the mouse over again.
 - Click Test to open Case Manager Client.

Procedure 4: Test the choice list

1. The Case Manager Client is opened.
 - a. In the Lab Claims Solution > Customer Service Rep role, select the Cases tab.
2. Search for cases.
 - a. In the Search section, select Policy Family name from the list.
 - b. Enter % in the next text box of the Search widget and click the Search button.
 - c. Available cases are listed in the Case List widget.
3. Open case details for a case.
 - a. Click the Title link of the case with the “Policy Family Name”: Smith in the Case List widget in the right pane.
The case opens in the Script Case Details page.
4. In the Script Case Details tab, click the Verification toolbar button at the top of the page.
5. The Add Task page is opened.
6. Check that the “Case Stage” property has a choice list that is attached to it because of the Script Adapter code.

VerifyDocs LCS_AutoClaimsGeneral_000000100001 |



7. For this procedure, it is not necessary to complete the task. Click Cancel.
 - a. Log out of the client.
8. Log out of the applications and close the browser windows.

LESSON 2.4: Validate the data based on a step response

What this lesson is about?

This lesson describes how to validate the data with a Script Adapter when a specific step response is selected.

What you should be able to do?

After completing this lesson, you should be able to:

- Validate the data based on a step response.

How you will check your progress?

- Hands on labs.

Lab Solution files

- The solution files for this lesson are included in the following folder:
C:\ICM\Script Adapter Code

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Data Validation that is based on a step response

Data Validation

- For many case solutions, certain step data is required for a particular step response.
- But another step response might not require the same step data.
- Instead of making all the step fields as required, you can add a data validation that is based on the step response that is selected.

IBM Case Manager JavaScript classes that are used in this lesson

Constants.CoordContext.WKITEMRESPONSE

- `icm.base.Constants` is helper class that defines the constants for IBM Case Manager.
- Constants for the context of coordination topics carry through all steps of a coordination.
- `WKITEMRESPONSE` the response name the user selected.

Constants.CoordTopic.VALIDATE

- The `CoordTopic` field represents the constants for coordination topics.
 - `VALIDATE` is the topic for a coordination step for page widgets to do data validation.
-

Exercise 2.4.1: Validate the data based on a step response

Introduction

For many case solutions, certain step data is required when a step response is clicked. But another step response might not require the same step data. Instead of making that step field as required, you can add a data validation that is based on the step response.

In this lab exercise, you add a data validation that is based on a step response selection. You use the discretionary task that you added in the previous exercise.

Procedures

Procedure 1: Check the default behavior for a step response, page. 2-44

Procedure 2: Customize the work details page, page. 2-45

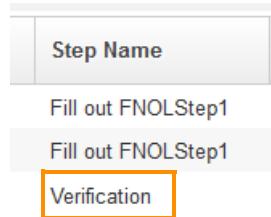
Procedure 3: Redeploy the solution, page. 2-49

Procedure 4: Test the data validation, page. 2-49

Procedure 1: Check the default behavior for a step response

The Lab Claim Solution contains discretionary tasks required for this lab exercise.

1. Log in to the Case Manager Client in a Firefox browser.
 - URL: <http://ecmedu01:9080/navigator/?desktop=icm>
 - User name: P8admin
 - Password: IBMFileNetP8
2. In the Lab Claims Solution > Customer Service Rep role, select the Work tab.
3. Open Work details for the Verification step work item.
 - a. In the “CSR Tasks” subtab, click the link in the “Date of Loss” column for a row item with the “Step Name”: Verification.
If you have many items with the “Step Name”: Verification, you can use any item.
 - b. You need to scroll the side bar to see the “Step Name” column.



The step opens in the “Verification” work details page.

**Hint**

This work item is created when you added a discretionary task in Procedure 3: Test the custom script action to add a task, page. 2-30. If the work item does not exist, repeat the procedure to create this work item.

4. Complete the work item with the “Need Clarification” step response.
 - a. Notice that the “Reason” case property field in the left pane is empty.

Policy Phone Number	<input type="text" value="111-222-4444"/>
Reason	<input type="text"/>

- b. Click the “Need Clarification” button.
- c. This action completes the step and the Verification page is closed.
- d. Back in the Work tab > “CSR Tasks” subtab, the work item that you completed is no longer listed.
- e. Log out of the client.

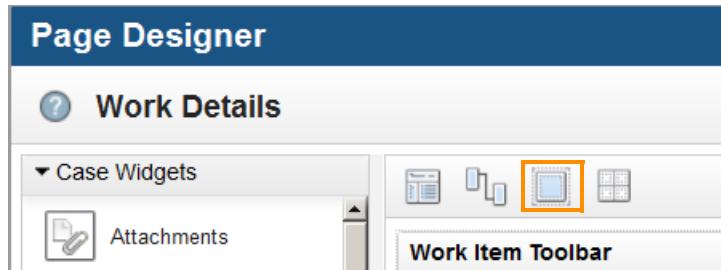
Procedure 2: Customize the work details page

In this procedure, you add a data validation to check that the “Reason” field has a value before you are able to complete the step with the “Need Clarification” option.

In this procedure, you edit the work details page to add the Script Adapter.

1. In Firefox, log on to IBM Case Manager Builder as an administrative user.
 - URL: <http://ecmedu01:9080/CaseBuilder>
 - User name: P8Admin
 - Password: IBMFileNetP8
2. In Case Manager Builder, open Lab Claims Solution by clicking the Edit link.
Hover the mouse over the solution to see the links.
3. Edit the work details page in Page Designer:
 - a. Open the Pages tab and expand the Work Details Pages.
 - b. Click the “Work Details” link to open the page.
4. In Page Designer > Work Details page, add a Script Adapter widget.
 - a. Click the “Show or Hide Hidden Widgets” button.
A gray section appears on the bottom of the main layout area.

- b. The section already contains a Script Adapter.



- c. Drag the Script Adapter widget (under the "Utility Widgets") from widget palette on the left column to the bottom of the page on the right.



5. Rename the widget:

- a. Click the down-arrow, and select "Rename Widget".



- b. In the "Rename Widget" dialog page, edit the Widget name as DataValidation.

- c. Click OK.

6. Click the "Edit Settings" icon of the Script Adapter widget.



- a. In the Script Adapter dialog window, clear the text in the JavaScript text box.

7. Enter the following lines of code into the JavaScript text box.



Hint

Optionally, copy the text from the C:\ICM\Script Adapter Code\DataValidationStepResponse.txt file and paste it.

```
var coord = payload.coordination;
/* Retrieve the work editable object from the payload */
var workitemEdit = payload.workItemEditable;
var solution = this.solution;
var prefix = solution.getPrefix();
// Load the constants and controller manager
require(["icm/base/Constants",
"icm/model/properties/controller/ControllerManager"], function(Constants,
ControllerManager){
if(coord){
/*Participate in the VALIDATE coordination step*/
coord.participate(Constants.CoordTopic.VALIDATE, function(context, complete,
abort){
/*Check the specific response, an empty string ("") stands for the Complete
button*/
if(context[Constants.CoordContext.WKITEMRESPONSE] === "Need Clarification"){
/*Check work item property attribute's value*/
var propId = prefix + '_Reason';
/*Retrieve prop value using property controller API*/
var theController = ControllerManager.bind(workitemEdit);
var propController = theController.getPropertyController(propId);
var value = propController.get("value");
/*Unbind is necessary to prevent mem leak.*/
ControllerManager.unbind(workitemEdit)
if(value === null || value === undefined || value === ""){
/*Abort the workitem completion with passing a message to user*/
abort({"message":"Please enter a reason before sending it for more
clarification."});
}else{
complete();
}
}else{
}
}})
```

```
        complete();
    }
});
}
}
});
```



Troubleshooting

If you have any errors in the JavaScript, the OK button is disabled. Also, it shows an error message. Verify the code that you entered. If needed, clear the code, copy, and paste the code again.

How does the script works?

- Retrieve the work editable object from the payload.
- Load the constants and controller manager.
- Participate in the VALIDATE coordination step.
- Check the specific response, an empty string (" ") stands for the “Complete” option.
- Specify the value for the work item property attribute.
- Retrieve the property value with property controller API.
- Add Unbind to prevent the memory leak.
- If the value is empty, stop the work item completion, and show a message to user.

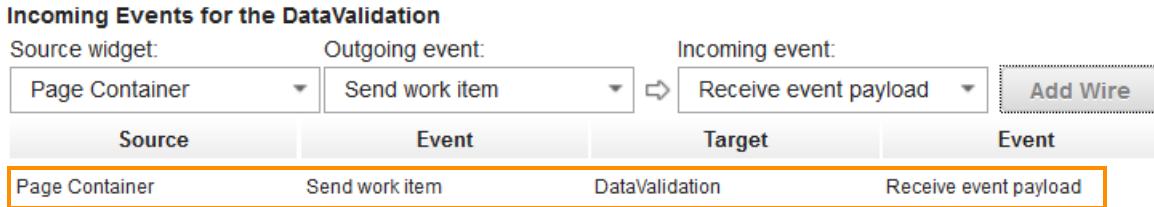
-
8. Click OK to close the Script Adapter dialog page.
 9. Click the Edit Wiring icon for the Script Adapter widget.



10. Wire Script Adapter to Page Container to receive the “Send workitem” outgoing event.
 - a. In the “Wire Events” page > “Incoming Events for DataValidation” section, select the following values from the list for each field:

Field	Value
Source widget	Page Container
Outgoing event	Send Work Item
Incoming event	Receive event payload

- b. Click Add Wire.
- c. Validate that the completed wiring looks like the following screen capture.



11. Click Save to save your work.
 - a. Scroll down and click OK at the bottom of the Wire Events dialog window.
 - b. Click Save and then Close to close Page Designer.
 - c. Click "Save and Close" at the top of the page to exit the solution editor.
 - d. Leave the Case Manager Builder open for the next procedure.

Procedure 3: Redeploy the solution

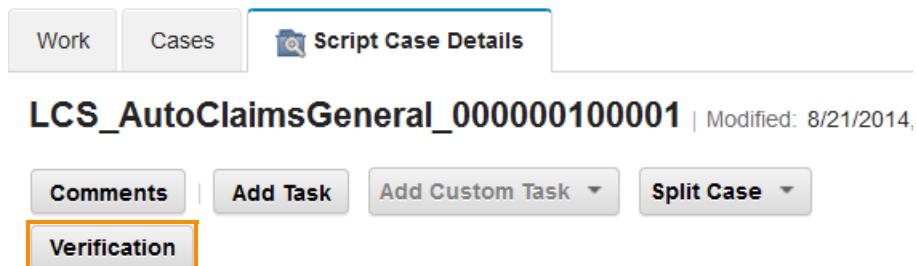
1. In the Manage Solutions page, select Lab Claims Solution and hover the mouse over.
 - a. Click Deploy.
 - b. In the Confirmation page, make sure that "Commit my changes and make them available for deployment" is selected, and click Deploy.
 - c. Wait for the green check mark to appear next to the solution.
2. Select Lab Claims Solution and hover the mouse over again.
 - a. Click Test to open Case Manager Client.

Procedure 4: Test the data validation

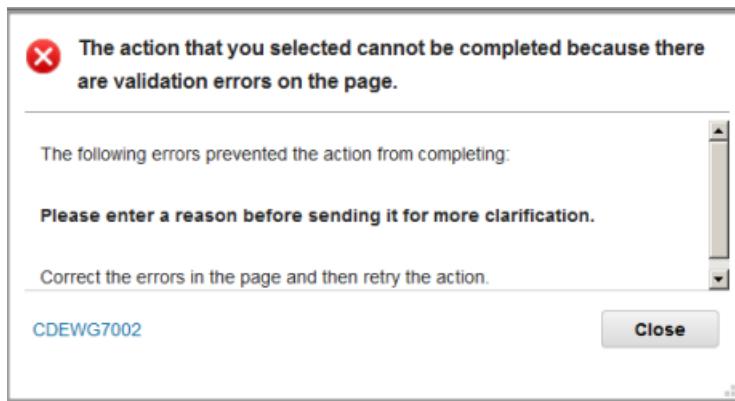
In Procedure 1, you completed the work item for the Verification task. For the data validation test, you are going to start another task.

1. The Case Manager Client is opened.
 - a. In the Lab Claims Solution > Customer Service Rep role, select the Cases tab.
 2. Search for cases.
 - a. In the Search section, select Policy Family name from the list.
 - b. Enter % in the next text box of the Search widget and click the Search button.
 - c. Available cases are listed in the Case List widget.
 3. Open case details for a case.
 - a. Click the Title link of the case with the "Policy Family Name": Smith in the Case List widget in the right pane.
- The case opens in the Script Case Details page.

4. In the Script Case Details tab, click the Verification toolbar button at the top of the page.



5. The Add Task page is opened.
 - a. If required, enter a value for the Estimate field.
 - b. Click Add to add the task.
 - c. Back in the Script Case Details page, click Save and then Close.
6. Select the Work tab.
7. Open work details for the Verification step work item (for the task that you just launched).
 - a. In the "CSR Tasks" subtab, click the link in the "Date of Loss" column for the row item with the "Step Name": Verification.
 - b. You need to scroll the side bar to the right to see the "Step Name" column.
 - c. The Verification tab is opened.
8. Test the data validation.
 - a. Leave the "Reason" case property field in the left pane is empty.
 - b. Click the "Need Clarification" button.
 - c. Verify that an error message as you entered in the script is shown.



9. Complete the work item with the "Need Clarification" step response.
 - a. Close the error message window.
 - b. Enter some text in the "Reason" case property field on the left.

- c. Click the “Need Clarification” button.
 - d. Verify that this time, the action completes the step and the Verification page is closed.
 - e. Back in the Work tab > “CSR Tasks” subtab, the work item that you completed is no longer listed.
10. Log out of the applications and close the browser windows.
-

LESSON 2.5: Create a case custom workbench page

What this lesson is about?

This lesson describes how to create a case workbench in IBM Case Manager with Script Adapters. From the workbench, the users view their assigned cases and the associated work.

What you should be able to do?

After completing this lesson, you should be able to:

- Create a case workbench.

How you will check your progress?

- Hands on labs.

Lab Solution files

- The solution files for this lesson are included in the following folder:
C:\ICM\Script Adapter Code

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

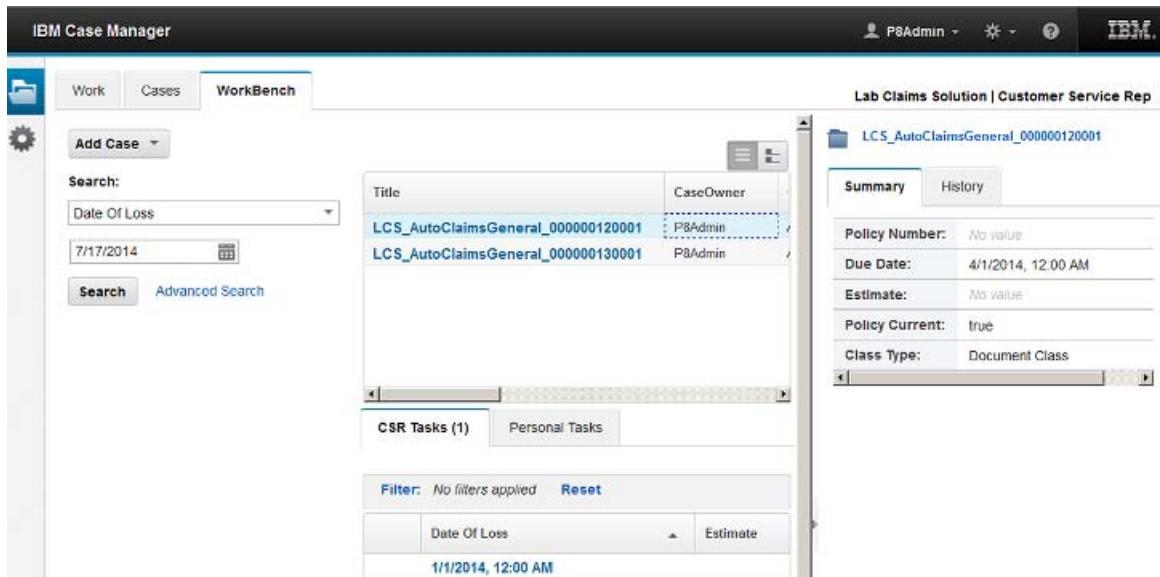
IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Case custom workbench pages

Introduction

- When a user logs in to the default IBM Case Manager Client, the client opens with two tabs initially:
 1. **Work** - The Work page shows a list of work items in the in-baskets.
 2. **Cases** - The Cases page allows users to run searches that are based on case properties.
- Work is assigned to users in Case Manager but it does not implement case ownership by default.
- The customization in this lesson provides a case-oriented view of case management.
 - In situations, where users own all or part of a case, you can configure Script Adapters and set up a case workbench.
 - From the workbench, the users can view only their assigned cases and the associated work.



A Case property that indicates the case ownership

The case owner has the responsibility for all of the activities that are associated with a case.

- In this lesson, you have a user who is assigned to work on a set of cases.
- A case property that identifies the case owner for a specific case.
- You use that property to determine what is shown on the custom page.

- When the Case List widget is populated, it displays only the cases for which the current user is assigned as the case owner.
- The customization in this lesson provides a workbench that is based on the association of a specific user with a case.

**Note**

You can create workbench pages that are based on any case property.

In-baskets

- An in-basket displays IBM Case Manager work items.
- Use an in-basket to view, open, and work with work items (activity) that are assigned to users.
 - If an activity is assigned to you, it is shown in your personal in-basket.
 - If the work items are assigned to a group, they are shown in one or more role in-baskets.
- The in-basket widget contains a separate tab for your personal in-basket and for each role.
- The “all assigned work in-basket” shows a list of all the open work items that are assigned to users.

In-baskets Filters

You can filter the work items list to display only the work that is assigned to a specific user or based on any other criteria.

- Define the In-baskets filter criteria in Case Manager Builder.
 - For System Properties based filters, use the Process Designer > Configuration tool.
- The filter restricts the items that are shown for faster access, and minimizes the time that you spend locating items in your inbox.
 - Example: filter for high priority items

Widgets that are used for this lesson

Introduction

In this lesson, you reuse the default page widgets; you also add code to the generic Script Adapter (SA) widgets to create a customized workbench page for the Case Manager user.

Script Adapters

- **Filter In-basket SA**
 - This widget filters only the work items that are specific for the selected case.
- **Filter Search SA**
 - This widget builds a custom search query so that active cases only for the current user are presented.
 - You can modify this sample for an administrative case view that shows the active cases for a list of specific users instead of the currently logged in user.

Default widgets that are available in the Case Manager Client

- **Case List**
- **In-Baskets**
 - The In-Basket widget lists the active work items. The Filter In-basket SA is used to ensure that only work items for the selected case are visible.
- **Search widget**
 - This widget is optional. It allows the users to further filter the cases that are listed in the Case List widget.
 - Without this widget, the users select only the cases from the Case List widget.
 - The Search widget is visible by default.

How the page is presented?

- When the user first selects the page, only the Case List widget and the Case Search widget are visible.
- When the user selects a case, the Case Summary widget and the In-basket widget are presented.
 - The case summary displays the summary information about the selected case.
 - The In-basket widget displays the active steps that are assigned to users for that case.

Filter In-basket SA - Script Adapter

Introduction

- This widget filters only the work items that are specific for the selected case.
- It is wired to the In-basket widget and
 - In-basket widget shows the work items that are filtered.

Filter In-basket SA wiring and how it works

- Filter In-basket SA is wired to the Case List and In-basket widgets.
 - The custom widget receives the “Select Case” or the “In-basket selected” outgoing event and the incoming event payload.
 - When the user selects a case in the Case List widget, a payload is sent to Filter In-baskets SA that includes the GUID of the selected case.
- The script that you add to the custom widget filters the work items from the payload that is based on the filter criteria.
 - This widget then sends the event payload (containing only the filtered work items) back to the In-basket widget.
 - This widget also causes the In-basket widget to refresh the work items list.



Note

In-basket filters can be created for any case property. Example: Work items that are based on user ID.

How does the script work?

You create an in-basket dynamic filter in your script.

- Include the In-baskets names, that you use it for your custom page, in an array in JSON format.

```
var inbasketNames = [ {  
    "queueName" : "Inbox",  
    "inbasketName" : solutionPrefix + "_" + "All Assigned Work"  
}, {  
    "queueName" : solutionPrefix + "_" + "CustomerServiceRep",  
    "inbasketName" : "CSR Tasks"  
} ];
```

- The script creates a dynamic filter that is based on the in-basket names that you provided.
 - It loops through the array, and adds each in-basket to the filter.
 - It also includes filtering so that only work items that are associated with the selected case are shown.

```
for (var i = 0; i < inbasketNames.length; i++) {  
    var data = {  
        "queueName" : inbasketNames[i].queueName,  
        "inbasketName" : inbasketNames[i].inbasketName,  
        "hideFilterUI" : false,  
        "queryFilter" : "(F_CaseFolder = :A)",  
        "queryFields" : [ {  
            "name" : "F_CaseFolder",  
            "type" : "xs:guid",  
            "value" : "{" + payload.caseEditable.id + "}"  
        } ],  
        "hideLockedByOther" : false  
    };
```

- Pass the data to the `icm.model.InbasketDynamicFilter.fromJSON(data)`; function to run the dynamic filter.
 - Push model data that the function returns to a model array.
 - Model array that is returned is sent out in the outgoing event payload.

```
    var model = icm.model.InbasketDynamicFilter.fromJSON(data);  
    console.debug(this.name, "model:", model);  
    modelArray.push(model);  
    console.debug(this.name, "modelArray:", modelArray);  
}  
console.debug(this.name, "Returning the filter");  
return {  
    "dynamicFilters" : modelArray  
};
```

Filter Search SA - Script Adapter

Introduction

- This widget filters only the cases that have the current user as the case owner.
 - It also filters only the active cases.
- The Case Type that you use has a case property that identifies the Case Owner.

Filter Search SA wiring and how it works

- Filter Search SA is wired to the Page Container.
- When the user opens the custom Solution page (WorkBench), the initial payload is sent to the Filter Search SA widget.
 - The Filter Search SA widget processes the initial payload, builds the search criteria (<Case Owner> property value = Current user and Case State = Active), and searches for the cases.
- The event is broadcast and the Case List widget receives the event.
 - This filter (with the search criteria) ensures that only the active cases for which the current user is the Case Owner are displayed in the CaseList widget.
 - The Filter Search SA also filters for a particular Case Type that is specified in the SA script. The configuration block must be configured for each Case Type.
- Optionally, you can also wire your widget to the default Search widget to further filter the cases.

How does the script work?

You create and run a custom search query that is based on the case owner in your script.

- Identify the specific case type and the case property that specifies the case owner.

```
/* Identify the specific case type without the prefix. */
var myCaseType = "AutoClaimsGeneral";

/*
 * Specify the case property (without the prefix)
 * that is matched against the logged in user ID.
 */
var myCaseProperty = "CaseOwner";

/* The display name of the case property
var myCasePropertyName = "CaseOwner";
```

- Retrieve the logged in user and the solution.

```
/* ID of the logged in user that will be inserted into the search criteria. */
var userId = ecm.model.desktop.userId;

/* The object for the solution we are working. */
var solution = this.solution;

/* The solution prefix. */
var prefix = solution.getPrefix();
```

- Build the search criterion for active cases.

```
/* Criterion for the search we are building. ( display only working cases.) */
var criterion = new ecm.model.SearchCriterion({
    "id" : "cmAcmCaseState",
    "name" : "CaseState",
    "selectedOperator" : "EQUAL",
    "dataType" : "xs:integer",
    "defaultValue" : 2,
    "value" : 2
});
```

- Build the search criterion for user ID.

```
var criterion1 = new ecm.model.SearchCriterion({
    "id" : myCaseProperty,
    "name" : myCasePropertyName,
    "selectedOperator" : "EQUAL",
    "dataType" : "xs:string",
    "defaultValue" : userId,
    "value" : userId
});
```

- Set up the search parameters.

- Add the object store, solution name, case type, and search criteria to a variable.

```
/* Set object store Id on params. */

params.ObjectStore = solution.getTargetOS().id;

/* Set up the search parameters on params. */

params.criterions = [ criterion, criterion1 ];
params.CaseType = caseType;
params.solution = solution;
```

- Create an instance of a search payload object and pass the parameters.
 - Call the `getSearchPayload` function.
 - Within the function, call the `onBroadcastEvent` method.

```
var searchPayload = new icm.util.SearchPayload();
searchPayload.setModel(params);

searchPayload.getSearchPayload(dojo.hitch(self, function(payload) {

    self.onBroadcastEvent("icm.SearchCases", payload);
})
```

Lab overview

To create a Case workbench page, do the following steps:

1. Configure a filter for the `F_CaseFolder` system field of the in-basket.
 2. Create a Case property that indicates the case ownership, and assign it to a Case Type.
 3. Copy the default Cases Solution page to create the workbench page.
 4. Edit the new workbench page.
 - a. Add In-baskets widget to the new page.
 - b. Disable Search event broadcasting.
 - c. Create Filter In-baskets SA Script Adapter.
 - d. Create Filter Search SA Script Adapter.
 - e. Wire the Script Adapters.
 - f. Wire Case List widget to In-baskets to trigger refresh.
 5. Assign the page to the roles where it is used.
 6. Save and Deploy the solution.
 7. Test the workbench page.
-

Exercise 2.5.1: Configure your system for the workbench page

Introduction

In this lab exercise, you configure your Case Manager system for the workbench page. You create filters for the In-baskets that you use in your custom page. You add a case property for the Case Type that you use to create the cases. This property value indicates the case owner.

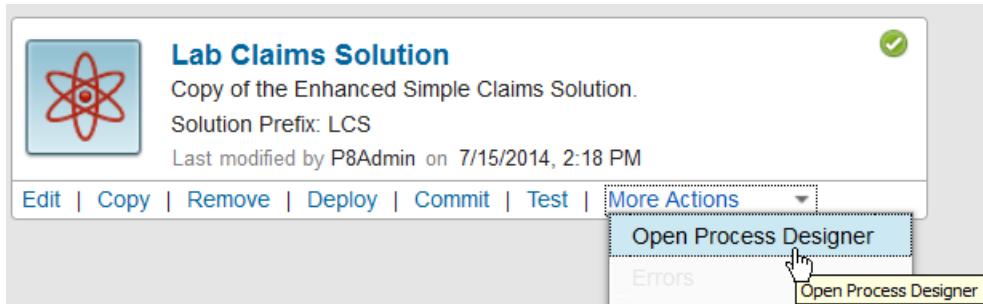
Procedures

Procedure 1: Configure queues to add a filter, page. 2-62

Procedure 2: Add a case property to assign a case owner, page. 2-65

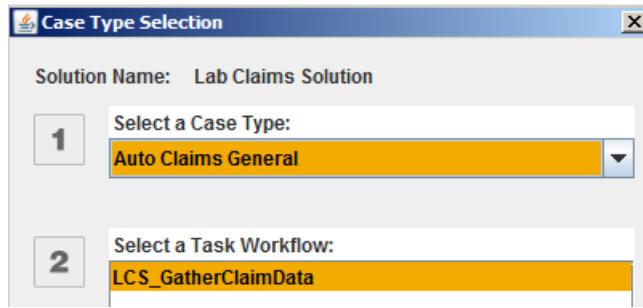
Procedure 1: Configure queues to add a filter

1. In Firefox, log on to IBM Case Manager Builder as an administrative user.
 - URL: <http://ecmedu01:9080/CaseBuilder>
 - User name: P8admin
 - Password: IBMFileNetP8
2. Start the Process Designer from the Solution.
 - a. Hover over your solution, click More Actions, and then click “Open Process Designer”.

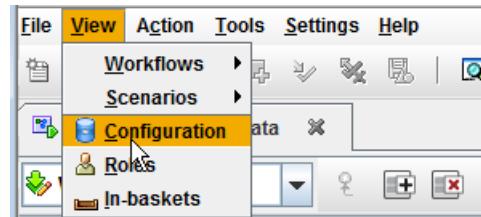


3. If the “Java Update Needed” page comes up, select “Do not ask again until the next update is available” and click Later.
4. In Case Type Selection page, select the values that is provided in the following table:

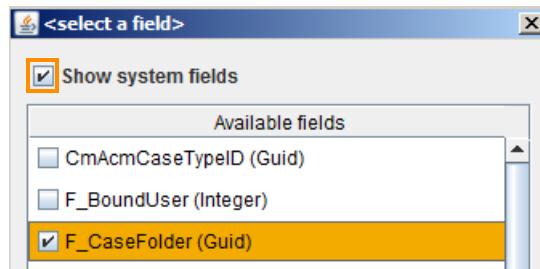
Field	Value
Select a Case Type	Auto Claims General
Select a Task Workflow	LCS_GatherClaimData



5. Click OK. The Workflow is opened in the Process Designer.
6. Click View > Configuration. The Configuration tab is opened.

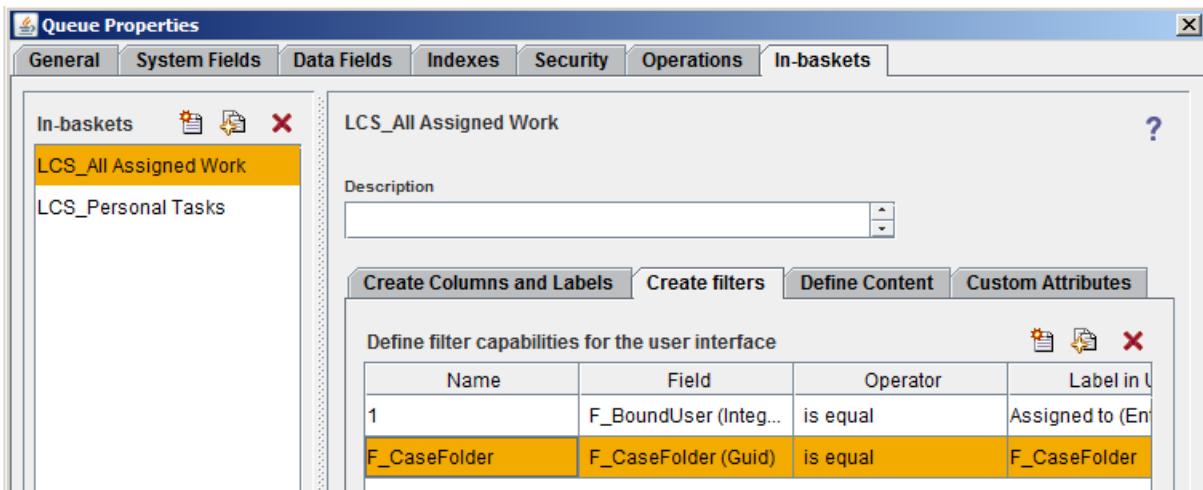


7. Open the In-baskets configuration tab for User Queues:
 - a. Expand User Queues > Inbox
 - b. Right-click Inbox and select properties.
 - c. In the Queue Properties window, select the In-baskets tab.
 - d. Select the "LCS_All Assigned Work" in-basket in the left pane.
8. Configure a filter for the in-basket:
 - a. Select the "Create filters" tab in the right pane and click the Add icon. 
 - b. A new row is added.
 - c. In the NewFilter2 row > Field column, click and then select the ellipsis (...) button to display the "Available Fields" menu.
 - d. Select "Show system fields". A list of system fields are added to the menu.
 - e. Select F_CaseFolder (Guid) from the list and then click OK.



9. Rename the filter:

- Double-click in the Name column. Clear the text, and type F_CaseFolder.
- The “Label in UI” column updates automatically to match the Name column.
- If the UI column is not updated, then double-click in that column and type F_CaseFolder.
- The completed configuration must look like the following screen capture.

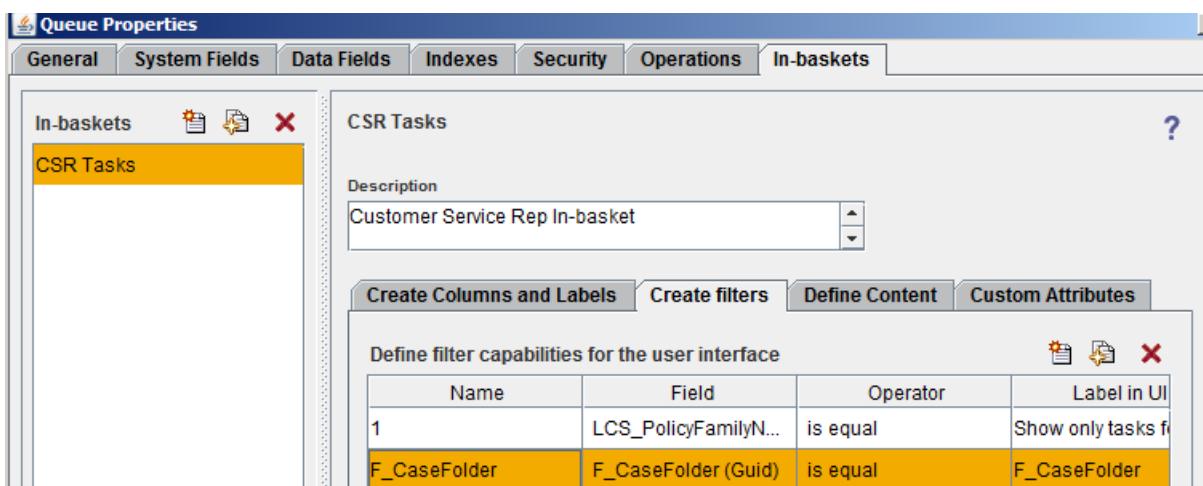


10. Save the changes to the solution:

- Click OK to close the “Queue Properties” page.
- Back in the Process Designer, click File > Solution > Save.

11. Open the In-baskets configuration tab for Work Queues:

- Expand Work Queues. Right-click LCS_CustomerServiceRep and select properties.
- In the Queue Properties window, select the In-baskets tab.
- Select the “CSR Tasks” in-basket in the left pane.



- d. Repeat Steps 8-10 to configure a filter for this In-basket.
12. Click File > Solution > “Save and Close” to save the changes and close the tool.



Troubleshooting

What to do when you are configuring the queues, if the page stops responding? It might be due to a long period of inactivity, and the session times out. To continue, go to the Case Manager Builder window and log in again.



Note

You must edit all the queues that you plan to show on your Workbench page. When you edit the queue configuration, it applies to all case types for the solution because the role and queue configuration is shared across the entire solution.

13. In Case Manager Builder, select Lab Claims Solution and commit the changes.
14. Leave the Case Manager Builder opened for the next procedure.

Procedure 2: Add a case property to assign a case owner

1. In Case Manager Builder, open Lab Claims Solution by clicking the Edit link.
2. Add a case property.
 - a. In the Properties tab, click Add Property and select New from the list.
 - b. Enter CaseOwner for the Name field, leave the default (String) for the Type field.
 - c. Optionally, enter a description.
 - d. Click OK.
3. Add the CaseOwner property to the Auto Claims General case type.
 - a. Select the Case Types tab.
 - b. Click the Auto Claims General link.
 - c. In the Case Type page, select Properties from the left pane.
 - d. Click Add Property > Existing, select CaseOwner and click OK.
 - e. Click OK All.
 - f. Verify that CaseOwner is added to the list.
4. Click “Save and Close”.
5. In the “Manage Solutions” page, commit the changes to your solution.
6. Leave the Case Manager Builder opened for the next procedure.

Exercise 2.5.2: Create a case custom workbench page

Introduction

In this lab, you create a custom workbench page to display cases that are assigned for a specific user and to display the associated work items. To achieve this configuration, you create a custom page that contains the default widgets and Script Adapters.

Procedures

Procedure 1: Create a page, page. 2-66

Procedure 2: Customize your page, page. 2-66

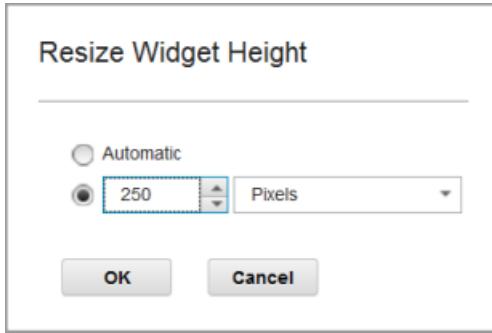
Procedure 1: Create a page

The workbench page is a customized version of the default Cases page. In this procedure, you create a copy the default Cases page layout for your workbench page.

1. In Case Manager Builder, open Lab Claims Solution by clicking the Edit link.
2. Create a custom page:
 - a. Open the Pages tab and expand the Solution Pages.
 - b. Hover the mouse over the Cases page name.
 - c. Select the copy icon on the right side of the page.
 - d. In the resulting page, edit the name to WorkBench for your new page and click OK to create the copy.
 - e. Save your work by clicking Save at the top of the page.
3. Leave the Pages tab opened for the next procedure.

Procedure 2: Customize your page

1. In the Pages tab, click your custom page (WorkBench) to edit it in Page Designer.
2. Reduce the Height of the Case List widget to make space for the In-baskets widget.
 - a. Click the “100%” link for the Height field.
 - b. In the Resize dialog window, edit the value to 250 pixels.
 - c. Click OK.



Case List
Height: 250Pixels
0 Incoming Events
2 Outgoing Events

**Note**

You can also use percentages to resize the widget height. The sizes are accepted in the order that the widgets appear on the column from top to bottom.

**Note**

Use of "Automatic" option might make the widget that is placed at the lower part of the page not get shown properly.

3. In the "Case Widgets" pallet on the left pane, scroll down and drag the In-baskets widget to the middle pane on your page below the Case List widget.
 - a. Click the "Edit Settings" icon for the In-baskets widget.
 - b. Select the "Do not populate the in-basket until the dynamic filter is received" option.



- c. Click OK.

- d. Click Save, and then Close to close the Page Designer.

**Note**

Enabling this option delays the in-baskets display until the dynamic filter is received. You configure the filter in a later procedure in this lesson.

4. Save your work by clicking "Save and Close".
 - a. In the Manage Solutions page, commit the changes to your solution.
5. Leave the Case Manager Builder opened for the next lab exercise.

Exercise 2.5.3: Add a Script Adapter to filter In-baskets

Introduction

In this lab, you add a Script Adapter to your custom workbench page to filter the information that is presented in user In-baskets. When you select a case, you see only the work items that are associated with the selected case.

For this lab, to display the Activity work items for a selected case, you filter the in-baskets that are based on the case folder GUID. You already configured this filter in Process Designer in a previous procedure.

Procedures

Procedure 1: Create Filter In-Basket SA Widget, page. 2-68

Procedure 2: Wire the Filter In-Basket SA widget, page. 2-69

Procedure 3: Assign the custom page to a role, page. 2-71

Procedure 4: Redeploy the solution, page. 2-71

Procedure 5: Test the Script Adapter, page. 2-72



Hint

For screen captures of the steps that are detailed in the following procedures, refer to Procedure 3: Edit the page to add the Script Adapter, page. 2-16 in the previous lesson.

Procedure 1: Create Filter In-Basket SA Widget

1. In Case Manager Builder, open Lab Claims Solution by clicking the Edit link.
2. In the Pages tab, expand the Solution Pages.
 - a. Click your custom page (WorkBench) to edit it in Page Designer.
3. Click the “Show or Hide Hidden Widgets” button.
 - a. Notice that a gray section appears on the bottom of the main layout area.
 - b. The section already contains two Script Adapters.
4. Drag and drop the Script Adapter widget (under the “Utility Widgets”) from widget palette on the left column to the bottom of the page on the right.
 - a. If your Script Adapter is not visible, scroll down the bottom section (it might be below the existing Script Adapters).
 - b. You can also expand the Script Adapter area.

5. Rename the widget:
 - a. Click the down-arrow, and select “Rename Widget”.
 - b. In the “Rename Widget” dialog page, edit the widget name as Filter In-Basket SA.
 - c. Click OK.
6. Click the “Edit Settings” icon of the Filter In-Basket SA widget.
 - a. In the Script Adapter dialog window that opens, clear the text in the JavaScript text box.
 - b. Copy the code from the C:\ICM\Script Adapter Code\In-Basket SA.txt file and paste it into the JavaScript text box.
 - c. Click OK to close the Script Adapter dialog page.



Troubleshooting

If you have any errors in the JavaScript, the OK button is disabled. Also, it shows an error message. Verify the code that you entered. If needed, clear the code, copy, and paste the code again.

7. Save the changes to the solution by clicking Save at the top of the page.
8. Leave the Page Designer open for the next procedure.

Procedure 2: Wire the Filter In-Basket SA widget

1. The Page Designer, click the Edit Wiring icon for your Script Adapter widget.
 - a. The “Wire Events” page opens.
2. In the “Incoming Events for Filter In-Basket SA” section, complete the following steps:
 - a. Select the values in the following table and click Add Wire.

Field	Value
Source widget	Case List
Outgoing event	Select Case
Incoming event	Receive event payload

- b. Repeat the wiring with the values in the following table:

Field	Value
Source widget	In-baskets
Outgoing event	In-baskets selected
Incoming event	Receive event payload

- c. Validate that the completed wiring looks like the following screen capture.

Incoming Events for the Filter In-Basket SA

Source widget: Outgoing event: Incoming event:
 In-baskets In-basket selected Receive event payload Add Wire

Source	Event	Target	Event
Case List	Select case	Filter In-Basket SA	Receive event payload
In-baskets	In-basket selected	Filter In-Basket SA	Receive event payload

3. In the “Outgoing Events for Script Adapter” section, complete the following steps:
 - a. Select the values in the following table and click Add Wire.

Field	Value
Outgoing event	Send event payload
Target widget	In-baskets
Incoming event	Apply filter

- b. Repeat the wiring with the values in the following table:

Field	Value
Outgoing event	Send event payload
Target widget	In-baskets
Incoming event	Refresh

- c. Validate that the completed wiring looks like the following screen capture.

Outgoing Events for the Script Adapter

Outgoing event: Target widget: Incoming event:
 Send event payload In-baskets Refresh Add Wire

Source	Event	Target	Event
Filter In-Basket SA	Send event payload	In-baskets	Apply filter
Filter In-Basket SA	Send event payload	In-baskets	Refresh

4. Click OK at the bottom of the page to close the “Wire Events” page.
5. Save your work.
 - a. Click Save and then Close to close Page Designer.
6. Leave the solution opened for the next procedure.

Procedure 3: Assign the custom page to a role

1. Your solution is already opened in the Case Manager Builder.
 - a. Open the Roles tab.
2. Click the Customer Service Rep role link.
 - a. Open the Pages subtab.
3. Optional: Remove the “Custom Scripts” page that you created in the previous lesson.
 - a. Select the page, hover over, and click the Remove (trash can) icon.



Note

Step 3, removes only the association of the page with a role. The actual page is still available if you need to use it. The removal step helps to minimize the number of tabs in the Case Manager Client.

4. Assign the new page that you created.
 - a. Click Assign Page.
 - b. Select Workbench.
 - c. Click OK to close the dialog window.
5. Verify that your page is listed in the Pages tab.
 - a. Click OK.
 - b. Click OK All to accept the changes to the role.
 - c. Click “Save and Close” at the top of the page to exit the solution editor.
 - d. Leave the Case Manager Builder open for the next procedure.

Procedure 4: Redeploy the solution

1. In the Manage Solutions page, select Lab Claims Solution and hover the mouse over.
 - a. Click Deploy.
 - b. In the Confirmation page, make sure that “Commit my changes and make them available for deployment” is selected, and click Deploy.
 - c. Wait for the green check mark to appear next to the solution.
2. Select Lab Claims Solution and hover the mouse over again.
 - a. Click Test to open Case Manager Client.



Troubleshooting

If browser caching is an issue, and your changes are not reflected in the Case Manager Client, you must log out of the Case Manager Builder and close all the browsers. Start the client in a new browser session.

Procedure 5: Test the Script Adapter

1. In the Case Manager Client, verify that Customer Service Rep role is selected on the upper right of the page.
 - a. If it is not selected already, select the role from the list.
2. Click the WorkBench tab to open your custom page.
3. Search for cases.
 - a. In the Search section, select Policy Family name from the list.
 - b. Enter % in the next text box of the Search widget and click the Search button.
 - c. Available cases are listed in the Case List widget.
4. Validate the Script Adapter widget.
 - a. Select the case with the "Policy Family Name": Smith in the Case List widget.



Hint

If you click the Title link of a case, the case opens in the Case Details page in a separate tab. Select a case by clicking other columns.

- b. Verify that the In-basket widget is shown below the Case List widget and it has a work item in the list.
- c. Maximize the browser to see the field values for the work item.
- d. Collapse the Case Information widget on the far right of the page with the arrow control.
- e. Verify that the work item has Smith as the "Policy Family Name".

The screenshot shows the IBM Case Manager interface with the 'WorkBench' tab selected. On the left, there's a sidebar with 'Add Case' and search filters for 'Policy Family Name' (set to '%') and 'Search'. The main area displays a table of cases:

Title	Policy Family Name	Case Type	Case State
LCS_AutoClaimsGeneral_000000100001	Smith	Auto Claims General	Working
LCS_AutoClaimsGeneral_000000110001	Beckner	Auto Claims General	Working

Below the table, there are tabs for 'CSR Tasks (1)' and 'Personal Tasks'. At the bottom, there's a 'Filter' section with a dropdown set to 'Policy Fam' and the value 'Smith' highlighted with a red box.

Use Scripts to Customize Case Manager Client

2-72

- f. To compare, open the Work tab and verify that the In-basket widget on the Work page shows all the active work items.

**Note** _____

Your custom code filtered the work items that are based on the Case Folder ID, and shows only the work item for that case in the In-basket widget.

- g. Log out of the Case Manager Builder and the client and close the browsers.
-

Exercise 2.5.4: Add a Script Adapter to filter cases

Introduction

In this lab, you add a Script Adapter to your custom workbench page to filter the cases that are presented in the Case List widget.

For this lab, you filter the cases that have the current user as the case owner. You already created a case property that is called CaseOwner for your case type in a previous procedure. You also filter only the active cases. To achieve the result, you create a custom search for the cases.

Procedures

Procedure 1: Create Filter Search SA Widget, page. 2-74

Procedure 2: Wire the Filter Search SA widget, page. 2-75

Procedure 3: Edit the Case List widget wiring, page. 2-76

Procedure 4: Disable Broadcasting from the Search widget, page. 2-77

Procedure 5: Test the Script Adapter, page. 2-78

Procedure 1: Create Filter Search SA Widget

1. In Firefox, log on to IBM Case Manager Builder as an administrative user.
 - URL: <http://ecmedu01:9080/CaseBuilder>
 - User name: P8admin
 - Password: IBMFileNetP8
2. In Case Manager Builder, open Lab Claims Solution by clicking the Edit link.
Hover the mouse over the solution to see the links.
3. Open the workbench page:
 - a. Open the Pages tab and expand the Solution Pages.
 - b. Click your custom page (workBench) to edit it in Page Designer.
4. Click the “Show or Hide Hidden Widgets” button.
5. Drag the Script Adapter widget from widget palette to the bottom of the page next to Filter In-basket SA.
 - a. If your Script Adapter is not visible, scroll down the bottom section.
 - b. Expand the Script Adapter area.

**Hint**

For screen captures of the steps that are detailed in the following procedures, refer to Procedure 3: Edit the page to add the Script Adapter, page. 2-16 in the previous lesson.

6. Rename the widget:
 - a. Click the down-arrow, and select “Rename Widget”.
 - b. In the “Rename Widget” dialog page, edit the widget name as `Filter Search SA`.
 - c. Click OK.
7. Click the “Edit Settings” icon of the Filter Search SA widget.
 - a. In the Script Adapter dialog window, clear the text in the JavaScript text box.
 - b. Copy the code from the `C:\ICM\Script Adapter Code\Search SA.txt` file and paste it into the JavaScript text box.
 - c. Click OK to close the Script Adapter dialog page.

**Troubleshooting**

If you have any errors in the JavaScript, the OK button is disabled. Also, it shows an error message. Verify the code that you entered. If needed, clear the code, copy, and paste the code again.

8. Save the changes to the solution by clicking Save at the top of the page.
9. Leave the Page Designer open for the next procedure.

Procedure 2: Wire the Filter Search SA widget

1. In the Page Designer, click the Edit Wiring icon for your Script Adapter widget.
 - a. The “Wire Events” page opens.
2. In the “Incoming Events for Filter Search SA” section, complete the following steps:
 - a. Select the values in the following table and click Add Wire.

Field	Value
Source widget	Page Container
Outgoing event	Page opened
Incoming event	Receive event payload

- b. Repeat the wiring with the values in the following table:

Field	Value
Source widget	Search
Outgoing event	Search Cases
Incoming event	Receive event payload

- c. Validate that the completed wiring looks like the following screen capture.

Source	Event	Target	Event
Page Container	Page opened	Filter Search SA	Receive event payload
Search	Search cases	Filter Search SA	Receive event payload

3. Click OK at the bottom of the page to close the “Wire Events” page.
4. Leave the Page Designer opened for the next procedure.

Procedure 3: Edit the Case List widget wiring

1. In the Page Designer, click the Edit Wiring icon for Case List widget.
 - a. The “Wire Events” page opens.
2. In the “Outgoing Events for the Case List” section, select the values in the following table:

Field	Value
Target widget	In-baskets
Outgoing event	Select case
Incoming event	Refresh

- a. Click Add Wire.
- b. This wiring refreshes the In-basket when a case is selected.
- c. Validate that the completed wiring looks like the following screen capture.

Outgoing Events for the Case List

Outgoing event:	Target widget:	Incoming event:	Add Wire
Source	Event	Target	Event
Open case	Page Container	Add case	
Case List	Select case	Filter In-Basket SA	Receive event payload
Case List	Select case	In-baskets	Refresh

3. Click OK at the bottom of the page to close the “Wire Events” page.
4. Leave the Page Designer opened for the next procedure.

Procedure 4: Disable Broadcasting from the Search widget

For the Case List widget to receive the search payload from your custom widget, the broadcast from the default Search widget must be disabled.

1. In the Page Designer, click the Edit Wiring icon for “Search” widget.
 - a. The “Wire Events” page opens.
 - b. Select the Event Broadcasting tab.
2. Clear the Enabled option.

Wire Events

Widget

If you disable a broadcast event, you must provide an alternative way for the widget to communicate with other

Enabled	Event Title	Event ID
<input checked="" type="checkbox"/>	Search cases	icm.SearchCases

- a. Scroll down and click OK.
3. Click Save and then Close to close Page Designer.
4. Click “Save and Close” at the top of the page to exit the solution editor.
5. Redeploy the solution:
 - a. In the Manage Solutions page, select Lab Claims Solution and hover the mouse over.
 - b. Click Deploy.
 - c. In the Confirmation page, make sure that “Commit my changes and make them available for deployment” is selected, and click Deploy.

- d. Wait for the green check mark to appear next to the solution.
6. Select Lab Claims Solution and hover the mouse over again.
 - a. Click Test to open Case Manager Client.

Procedure 5: Test the Script Adapter

1. In the Case Manager Client, verify that Lab Claim Solution > Customer Service Rep role is selected on the upper right of the page.
 - a. If it is not selected already, select the role from the list.
2. Select the Cases tab to add several cases for Auto Claims General case type:
 - a. Some cases with the CaseOwner property value as P8Admin and others with clara.
 - b. Sample data for adding cases is provided in the following table. Leave the other fields that are not given in the table blank.



Important

The search criteria values are case-sensitive. Make sure that you enter P8Admin ("P" and "A" are upper cases) correctly for the custom search to work. The value "clara" has all lowercases.

Auto Claims General case type	Case property	Value
Case - 1	CaseOwner	P8Admin
	Policy Family name	Steven Heights
Case - 2	CaseOwner	P8Admin
	Policy Family name	Barbara Seth
Case - 3	CaseOwner	clara
	Policy Family name	Emma Watts
Case - 4	CaseOwner	clara
	Policy Family name	Sara Cummings

3. Refresh the browser for the pages reload since the custom widget received the initial payload when the page loads.
 - a. Select the WorkBench tab to open your custom page.
4. Verify that only the cases with "P8Admin" as the case owner is listed in the Case List widget automatically without running a search. Your widget does the custom search.



Troubleshooting

If the cases are not listed, reload the page by logging out and logging back in to the Case Manager Client. The existing two cases (Policy Family Name: Smith and Beckner) are not listed because these cases were created before you added the CaseOwner property. These two cases do not have a value for the CaseOwner property.

5. You implemented the “Filter In-basket” code in the previous lab exercise.
 - a. Verify that when you select a case, only that work item for that case is listed (in stead of all the active work items) in the In-basket widget.
6. Validate the other case owners.
 - a. You already created a few cases with “clara” as the case owner.
7. Add the case owner “clara” as a member to the Customer Service Rep role.
 - a. Open the Work tab and click Manage Roles.
 - b. Select the Customer Service Rep role and click “Add Users and Groups”.
 - c. Search for clara user and move clara from the Available to Selected pane.

The screenshot shows the 'Add Users and Groups' dialog box. At the top, there is a 'Directory' dropdown set to 'sample'. Below it is a 'Search for' input field with 'Users' selected and 'clara' typed into the search bar. To the right of the search bar is a magnifying glass icon. The main area has two sections: 'Available' on the left and 'Selected' on the right. In the 'Available' section, 'clara' is listed with a small user icon and a blue dashed border around the name. Between the 'Available' and 'Selected' sections is a right-pointing arrow icon, which is also highlighted with an orange box. The 'Selected' section is currently empty.

- d. Click Add and then Save.
8. Verify the Workbench page with clara as the case owner.
 - a. Log out of the Case Manager Client and log in back as clara (password: filenet).
 - b. Select the Workbench tab to open your custom page.
 - c. Verify that only the cases with “clara” as the case owner is listed in the Case List widget.
 - d. Select a case, only that work item for that case is listed in the In-basket widget.
 - e. Log out of the Case Manager Builder and the client and close the browser.

3

Develop Custom Widgets

This unit provides guidance for developing IBM Case Manager custom widgets.

© Copyright IBM Corp. 2014

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

LESSON 3.1: Custom widget development overview

What this lesson is about

This lesson gives an overview of custom widget development environment. You create an IBM Content Navigator plug-in project in Eclipse for your custom widget.

What you should be able to do

After completing this lesson, you should be able to:

- Describe custom widget development environment.
- Create an IBM Content Navigator plug-in project in Eclipse.

How you will check your progress?

- Hands on labs.

Lab Solution files

- The solution files for this lesson are included in the following folder:
C:\ICM\Widgets\SearchWidget\Solution Files

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Developing case management applications

Introduction

IBM Case Manager and the IBM FileNet P8 software provide tools for building custom web applications to manage cases that you can use in the following ways:

- Extension points and application programming interfaces (APIs) to extend Case Manager Client.
 - Add custom pages, widgets, actions, events, or services.
- The APIs to build custom applications that incorporate Case Manager features.
 - It enables caseworker to process cases without using Case Manager Client.



Note

You can use Case Manager Builder to create a solution with a template, and then modify that solution to meet your requirements. You cannot use a custom application to create a solution. You must use Case Manager Builder to create or modify solutions.

IBM Case Manager JavaScript API

IBM Case Manager provides a JavaScript API that you can use to customize your case management client application.

- The IBM Case Manager JavaScript API uses the Dojo toolkit, which is an open source JavaScript library for web development.
- In addition, you use the IBM Content Navigator JavaScript API to customize your client application.
 - It includes more modeling and widget classes that you can use in your application.



Note

The “Use Scripts to Customize Case Manager Client” unit in this course has a detailed introduction to the JavaScript APIs.

Prerequisites

To develop a page widget, you must know the following technologies, and the programming models.

- Programming models:
 - IBM Case Manager Client
 - IBM Content Navigator
 - Technologies:
 - Dojo
 - Extensible Markup Language (XML)
 - HyperText Markup Language (HTML)
 - Java 2 Enterprise Edition
 - JavaScript
 - Eclipse IDE
-

Creating a custom page widget and actions package

High-level steps

1. **Create an IBM Content Navigator plug-in** for the widget package.
 - The plug-in classes and files are needed to use your custom page widgets in Case Manager Client, which runs in IBM Content Navigator.
 - These files also make any custom actions available in Case Manager Client, and IBM Content Navigator.

 Note _____

Refer to Lesson: 3.1, Custom widget development overview, on page 3-2 in this unit for more details about the IBM Content Navigator plug-in.
2. **Create the registry files** for the widget package.
 - The files are required to register the page widgets and make them available for use in Case Manager Builder.

 Note _____

Refer to Lesson: 3.2, Create catalog and widget definition files, on page 3-17 in this unit for more details about the registry files.
3. **Implement custom widgets** and any custom actions that are used by the widgets.
 - In this step, you create the JavaScript files that implement the functions for your custom page widgets and actions.

 Note _____

Refer to Lesson: 3.3, Implement a custom widget, on page 3-28 in this unit for more details about the implementation of custom widgets.
4. **Create the web project** for the custom page widget and actions (Optional).
 - The web project defines the folder structure that is required for building, packaging, and registering your custom widget package. Widgets are contained in an EAR file.

 Note _____

Refer to Lesson: 3.5, Create a widget with a toolbar and a menu, on page 3-57 in this unit for more details about the web project.

5. **Package the page widgets** and actions files for deployment.
 - You create the package as a compressed file (ZIP).
6. **Deploy the custom widget package.**
 - Use the IBM Case Manager administration web client for a small widget package.
 - Use the IBM Case Manager configuration tool for a larger widget package that contains widgets in an EAR file.
7. **Add the custom page widgets to a page** in Page Designer.
8. **Save and Deploy the solution** in Case Manager Builder.
9. **Test your custom widgets** in Case Manager Client.

**Note**

Refer to Lesson: 3.4, Build and register a widget package, on page 3-42 and Lesson: 3.6, Build and deploy a widget as an EAR file, on page 3-76 in this unit for more details about packaging and deploying a custom widget package. Refer to Lesson: 3.7, Update an existing widget package, on page 3-89 in this unit for more details about updating a widget package.

IBM Content Navigator plug-in for the custom widget package

Introduction

- In IBM Case Manager V5.2, the Case Manager Client application is deployed as a plug-in on IBM Content Navigator.
- When you create Case Manager custom widgets, you use the Content Navigator Model API.
- You must create the Content Navigator plug-in that makes your custom page widgets and actions available within IBM Content Navigator for Case Manager Client.

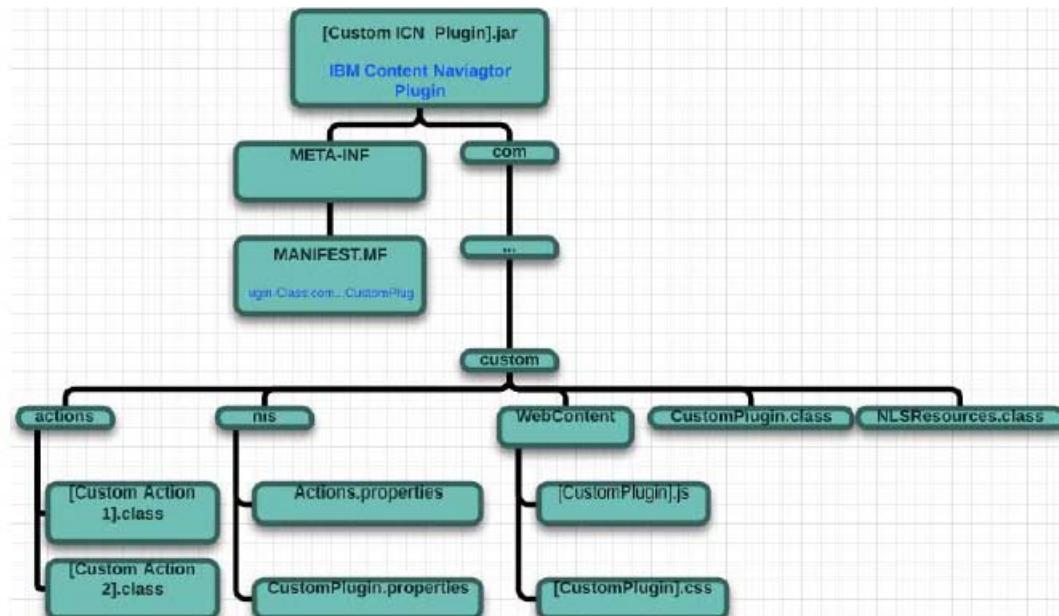
What is an IBM Content Navigator plug-in?

- A plug-in is a Java class that a developer must extend to implement a plug-in.
- It also refers to a Java Archive (JAR) file that contains Java classes to implement defined mid-tier extension points and web resources for extending the browser user interface.

Create an IBM Content Navigator plug-in

To create the plug-in package, you combine the classes and JavaScript files into a JAR file.

The following graphic shows a sample JAR file that contains the directories and files that are needed for the custom page widget and actions:



- The CustomPlugin.class describes the custom plug-in.

- The class provides the initial JS file name for the plug-in, and the action list in the package.
 - To declare the plug-in class to IBM Content Navigator, add the plug-in class to the META-INF\MANIFEST.MF.
 - The WebContent\[Custom Plug-in].js file registers the Dojo module path for the custom runtime code.
 - Depending on whether debug mode is on or off, the [Custom Plug-in].js file can load in the source code for debugging or in the combined and compressed code for better performance.
 - Each custom action class in the actions folder also provides an action definition.
 - The Java class com.ibm.icm.extension.custom.ILCMCustomPlugin is the single entry point of the sample IBM Content Navigator plug-in.
-

Setting up the development environment for plug-ins

Development environment

You can use any Eclipse-based development environment to create IBM Content Navigator plug-ins and custom widget packages.

- IBM Rational Application Developer
- Eclipse Software Development Kit (has different packages)



Note

For the student image, you use “Eclipse IDE for Java EE Developers” package. The lab exercises in this course use the Eclipse Kepler package that is based on Eclipse Version 4.3.1.

Eclipse plug-in for IBM Content Navigator

The Eclipse plug-in for IBM Content Navigator makes the creation of new Content Navigator plug-in projects easy. It contains the extensions for Eclipse and can be integrated in your development environment.

- The plug-in contains the following Java Archive (JAR) file:
 - com.ibm.ecm.plugin.202.jar

Installation of the Eclipse plug-in

1. Copy the JAR files in the <Eclipse installation path> /dropins directory.
 - Example for base Eclipse: C:/eclipse/dropins
 - Example for Eclipse for WebLogic:
C:/Oracle/Middleware/Oracle_Home/oepe/eclipse/plugins
 - Example for Rational Application Developer: C:/Program Files/IBM/SDP/plugins
2. Restart Eclipse.
Sometimes it is necessary to call Eclipse with the -clean parameter from the Command Prompt to make the plug-in active: C:/eclipse/eclipse.exe -clean



Note

The Eclipse plug-in can be downloaded from the IBM Redbooks publication page. For the student image, the plug-ins are already included in Eclipse.

Exercise 3.1.1: Create a plug-in project in Eclipse

Introduction

In this exercise, you create an IBM Content Navigator plug-in project in Eclipse.

User accounts

Type	User ID	Password
Operating system	administrator	passw0rd
IBM Content Navigator Administrator	P8Admin	IBMFFileNetP8

Passwords are always case-sensitive.

Procedures

Procedure 1: Start WebSphere Application Server, page 10

Procedure 2: Create a plug-in project in Eclipse, page 11

Procedure 3: Check the packages and files, page 12

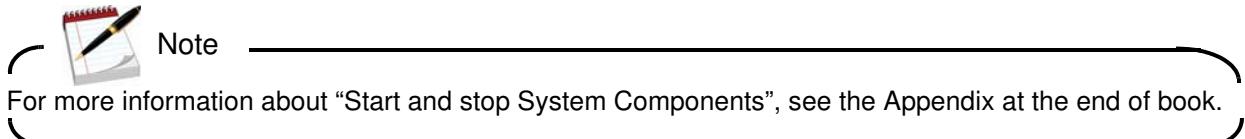
Procedure 4: Enable line numbers in Eclipse, page 14

Procedure 5: Configure Java Compiler, page 14

Procedure 6: Check the generated code in EDUPlugin.java file, page 15

Procedure 1: Start WebSphere Application Server

1. If it is not already started, start the WebSphere Application Server.
 - a. Click Start > All Programs > IBM WebSphere > IBM WebSphere Application Server V8.5 > Profiles > AppSrv01 > Start the server.
 - You can also use the `Start Server1.bat` file in the WebSphere Admin folder on the desktop.
2. Wait for the Start the server page to close.



Procedure 2: Create a plug-in project in Eclipse

Eclipse IDE for Java EE Developers and the Eclipse plug-in for IBM Content Navigator are already installed on your student image.

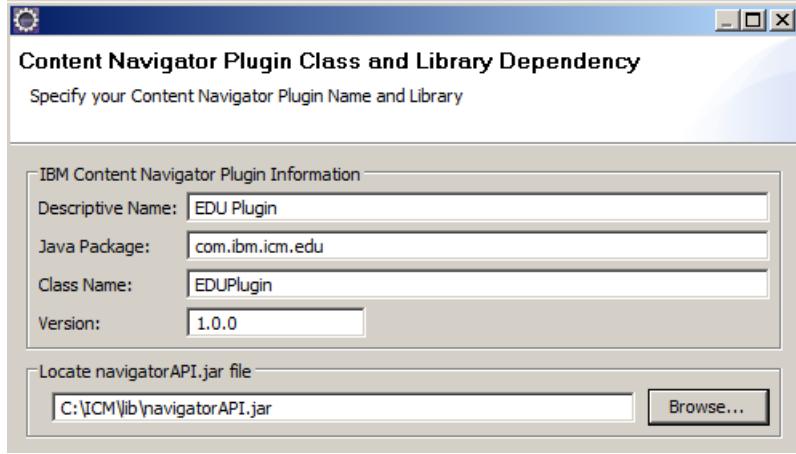
1. Open Eclipse by double-clicking the Eclipse icon in your desktop.
 - a. In the Workspace Launcher page, leave the default workspace directory (C:\ICM\workspace_Eclipse) and click OK.
2. Open the project creation wizard.
 - a. In Eclipse, click File > New > Other.
 - b. In the New page, scroll down and select IBM Content Navigator > Content Navigator Plug-in.
 - c. Click Next.
3. In the Content Navigator Plugin Project page, enter a Project Name (Example: EDUPlugin).
4. Leave other default settings and click Next.
5. In the “Content Navigator Plugin Class and Library Dependency” page, enter the values from the following table.

Item	Value	Notes
Descriptive Name	EDU Plugin	A string that identifies your plug-in in the IBM Content Navigator administration plug-in interface
Java Package	com.ibm.icm.edu	Namespace for your plug-in source code
Class Name	EDUPlugin	Name of your primary plug-in class. The class provides instructions to the IBM Content Navigator server about your plug-in extensions and the classes to load at run time.
Version	1.0.0	Version of the plug-in. The Version number is set in the class that is provided under Class Name.
Locate navigatorAPI.jar	C:\ICM\lib\navigatorAPI.jar	The JAR file contains the plug-in interfaces that the IBM Content Navigator provides you to build custom extensions.



Note

The navigatorAPI JAR file is installed with IBM Content Navigator under the lib directory (Example: C:\Program Files (x86)\IBM\ECMClient\lib). In your student system, this file is copied into the C:\ICN\lib folder.

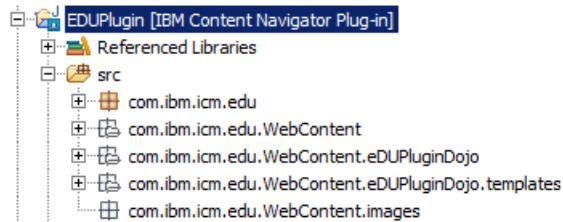


6. Click Finish. If the “Remember my decision” page prompts, click No.
7. The wizard generates a working Content Navigator plug-in project.

Procedure 3: Check the packages and files

In this procedure, you check the packages and files that the wizard generates for your project.

1. In the Package Explorer pane on the left, expand the EDUPugin [IBM Content Navigator Plug-in] project > src folder.

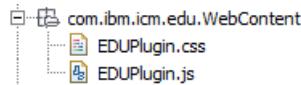


2. The following table lists the directories and their content in the src folder of your project.

Directories under the src folder of your project	Content
<PackageName> com.ibm.icm.edu	All the Java Classes extending the IBM Content Navigator plug-in classes
<PackageName>/WebContent	CSS files, JavaScript plug-in, and images folder
<PackageName>/WebContent/<Dojo>	Dojo classes (where you add your page widget files)
<PackageName>/WebContent/<Dojo>/templates	HTML templates, if you choose to extend the dijit._Templated dojo class

3. Expand each directory and check the files.

- WebContent directory (`com.ibm.icm.edu.WebContent`)



- Root directory for your plug-in extensions to the client-side.
- The wizard created shells that you can use to build your customizations.
- Example: You can add a style change to the CSS file that is generated in this directory.

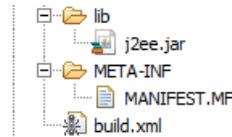
- <Dojo> package (`com.ibm.icm.edu.WebContent.eDUPuginDojo`)



- The wizard generates a Dojo package name for your custom Dojo widgets.
- IBM Content Navigator registers this namespace for your Dojo widgets.
- For example, if you create a custom dialog (Example: EDUDialog), you must add it to the `eDUPuginDojo` package and reference it as `eDUPuginDojo.EDUDialog`.

4. Expand the lib and META-INF folders in your project.

- lib folder
 - It contains the `j2ee.jar` by default.
 - You can add the required JARs for your plug-in to the lib folder.
- META-INF folder
 - It contains the `MANIFEST.MF` file that defines the main plug-in class for the project.
 - The Content Navigator plug-in registration requires this value to define the plug-in.
 - In the `MANIFEST.MF` file, the parameter “`Plugin-Class`” points to `com.ibm.icm.edu.EDUPugin` and it is automatically set through the wizard.



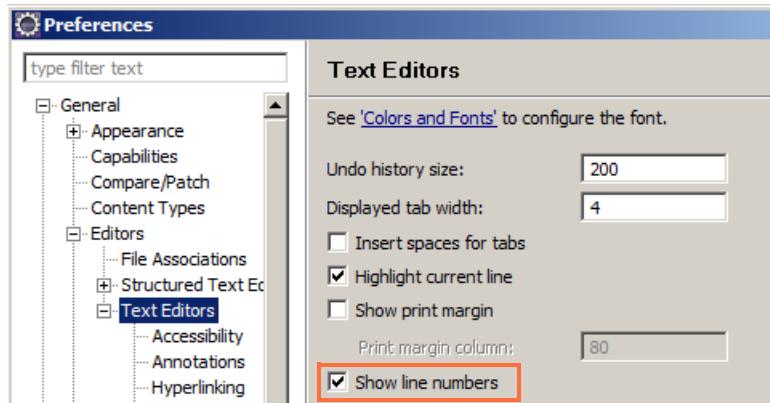
5. build.xml

- It is an ANT script that generates a JAR file out of the existing project code.
- This file contains the name of the output plug-in JAR file in the JAR section.

Procedure 4: Enable line numbers in Eclipse

If the line numbers are not visible in the text editor of the Eclipse, do the following steps or skip to the next procedure.

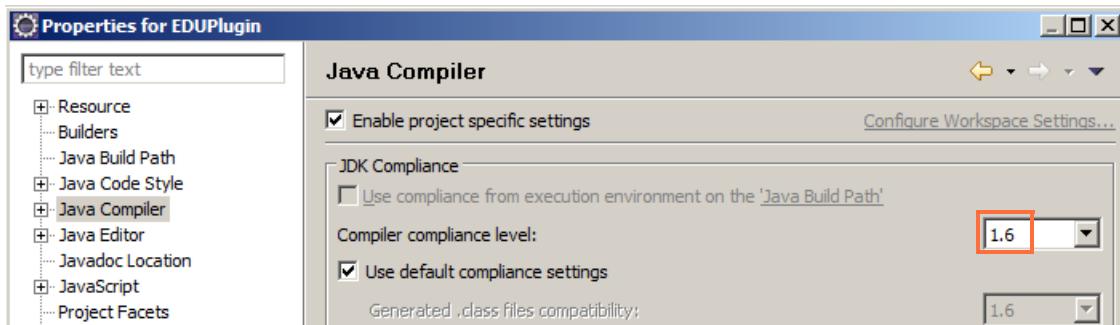
1. Select Window > Preferences from the top menu.
2. In the Preferences window, expand General > Editors, and select Text Editors.
3. Verify that the “Show line numbers” option is selected and click OK.



Procedure 5: Configure Java Compiler

 **Important**
You must ensure that you are using Java Compiler Level 1.6 in your plug-in project to avoid Java Version errors during plug-in registration.

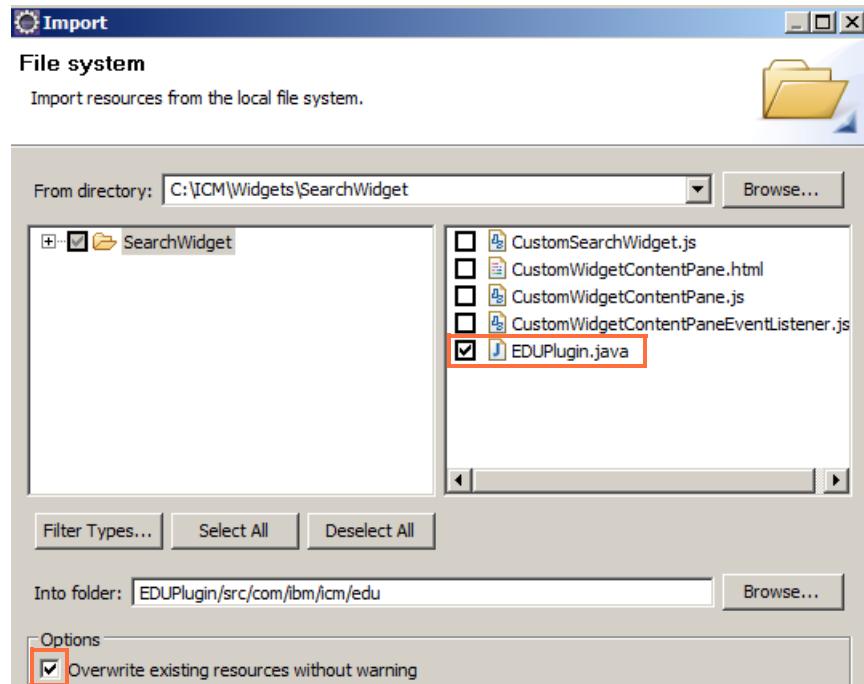
1. In Package Explorer, right-click the EDUPlugin project and select Properties from the list.
2. In the Properties for EDUPlugin page, select Java Compiler from the left pane.
3. In the right pane, select the “Enable project-specific settings” option.
4. Select the 1.6 from the list.



5. Select the Use “Default compliance settings” option and click OK.
6. When prompted, click Yes in the “Compiler Settings Changed” page to rebuild the project.

Procedure 6: Check the generated code in EDUPlugin.java file

1. If the file is not already opened:
 - a. In Package Explorer, expand your project > src > com.ibm.icm.edu.
 - b. Double-click the EDUPlugin.java file.
 - c. Observe that the wizard generated many methods in the Java file. Close the file.
2. Your student image has a copy of the EDUPlugin.java file with only the methods that are required for this lab exercise.
 - a. Replace the wizard-generated file with the lab file in the following steps.
3. Right-click the com.ibm.icm.edu package and click Import from the list.
 - a. In the Import page, expand General, select “File System”, and click Next.
 - b. In the Import > File system page, click Browse.
 - c. In the “Import from directory” page, go to C:\ICM\Widgets folder and select the SearchWidget folder. Click OK.
 - d. Back in the Import > File system page, select EDUPlugin.java from the list.
4. Make sure that the “Into folder” field has the following value:
EDUPlugin/src/com/ibm/icm/edu
 - a. Select the “Overwrite existing resources without warning” option.



- b. Click Finish.
5. Open the EDUPlugin.java and observe the methods.
 - a. The id value of the plug-in is required for the Content Navigator.
 - b. The Name of your plug-in is displayed in the admin tool.

```
public String getId() {  
    return "EDUPlugin";  
}  
  
public String getName(Locale locale) {  
    return "EDU ICN Plugin";  
}
```

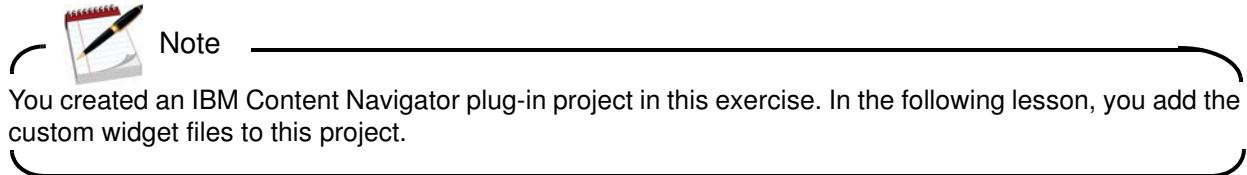
6. The base JavaScript file is loaded when IBM Content Navigator loads the plug-in.
 - a. You can use this JavaScript file to apply any global changes (such as a style override) or load any JavaScript classes that must be available throughout the session.

```
public String getScript() {  
    return "EDUplugin.js";  
}
```

7. The Dojo package where you are going to add your custom widget files.
 - a. IBM Content Navigator registers this namespace for your Dojo widgets.
 - b. Your custom page widgets become available in Case Manager Client, which runs in IBM Content Navigator.

```
public String getDojoModule() {  
    return "eDUPuginDojo";  
}
```

8. Leave Eclipse open through out this unit.



LESSON 3.2: Create catalog and widget definition files

What this lesson is about

This lesson provides guidance for creating the catalog and widget definition files. The files are used to register a custom widget package with Case Manager.

What you should be able to do

After completing this lesson, you should be able to:

- Create a catalog JSON file.
- Create a widget definition JSON file.

How you will check your progress?

- Hands on labs.

Lab Solution files

- The solution files for this lesson are included in the following folder:
C:\ICM\Widgets\SearchWidget\Solution Files

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Create registry files for the custom widget package

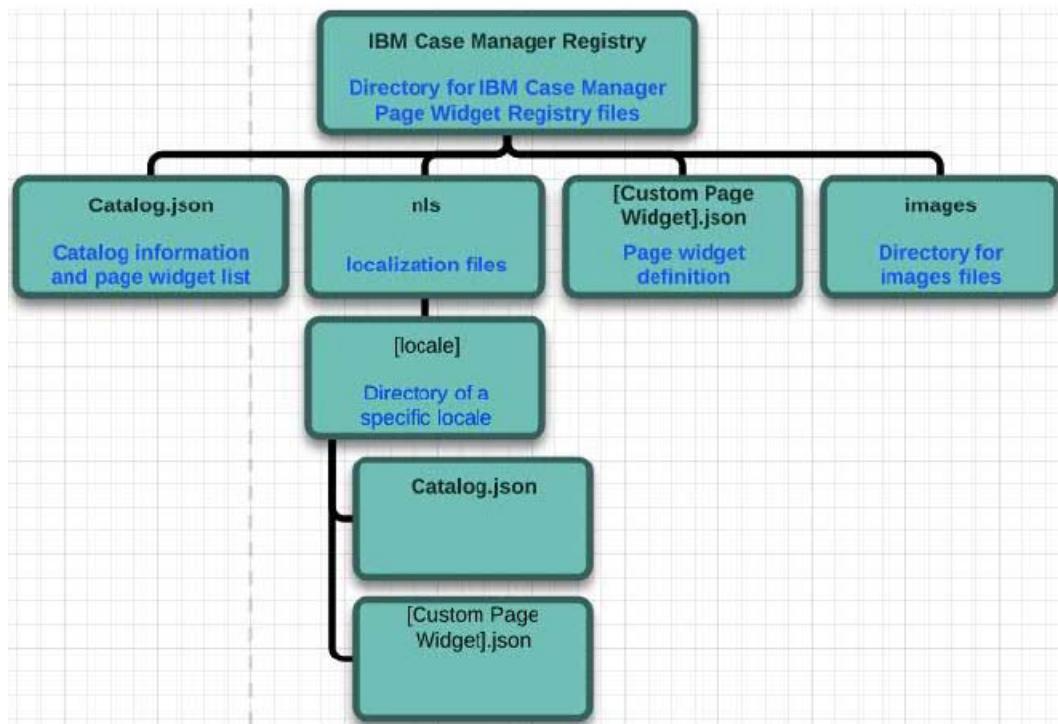
Introduction

As part of the widget package, you create files that are used to register your custom page widgets in Page Designer. After registration of the page widgets, they are available in the Page Designer palette.

- Following are the two files that are required for the registration.
 - Catalog.json
 - [Custom page widget].json
- You can also include the auxiliary files.
 - Example: Localized resource files and the image files that are used for the page widget.

Folder structure for registry files

The diagram shows a sample of the folder structure for registry files.



Catalog JSON file

Introduction

The Catalog.json file provides the following information:

- A description of your custom page widget package
- A list of page widgets in the package.

Description of the custom widget package

Description of the custom widget package contains the following properties.

Property	Required?	Type	Notes
Name	Required	String	- Name for your custom page widget package. - Must specify a unique name
Description	Required	String	
Locale	Required	String	The code that identifies the locale for the current catalog. This code must correspond to the subfolder name in the ICMRegistry/nls folder where the Catalog.json and widget definition file for the locale are located.
Version	Optional	String	
Categories	Optional	String	Creates a category in Case Manager Builder in which the custom page widgets in this package are listed. You can also choose to list your widgets in one of the following IBM Case Manager default categories: - Case Widgets - Utility Widgets For each category, you must provide an identifier and a title.

List of widgets in the custom widget package

Information about each widget in the package contains the following properties.



Important

Except for the definition property, these properties are identical to the properties in the widget definition file. For consistency, you can copy the values from that file into the Catalog.json file. If a value does not match, IBM Case Manager uses the value from the Catalog.json file.

Property	Required?	Type	Notes
id	Required	String	A unique identifier for the page widget.
category	Required	String	The identifier of the category in which the page widget is listed in Case Manager Builder.
title	Required	String	The name to be displayed for the page widget in Case Manager Builder.
description	Required	String	
definition	Required	String	The full path and name of the definition file for the page widget.
preview	Required	String	The relative path and name of the preview image file (.png or .gif) for the page widget.
icon	Required	String	The full path and name of the icon image file (.png or .gif) for the page widget. This image is used in the widget palette in Page Designer.
runtimeClassName	Required		The class name for the page widget as specified in the runtime plug-in for the widget package.
previewThumbnail	Required		The full path and name of the thumbnail image file (.png or .gif) for the page widget.
properties	Array		An array to define the properties that can be set for the page widget in Case Manager Builder.
events	Array	String	An array that identifies the events that the page widget publishes and subscribes to.

Page widget definition JSON file

Introduction

You create a JSON definition file to define the following information for the custom page widget:

- Properties that define the widget
- Properties that user can set for the page widget
- Events

Page widget properties

- The properties are used to define the custom widget.



Note

The widget property values are also used for the page widget in the Catalog.json file. Refer to the table in Procedure : List of widgets in the custom widget package, page 20 for a list of properties.

- Define the properties that can be set for the widget when the user adds the widget to a page.
 - Example: Toolbars, menus, and actions that can be added to the widget.

Page widget events

Define the events that the page widget subscribes to and publishes. For each event, provide the following properties:

Property	Description
id	A unique identifier for the event.
title	The title that is displayed in the Wiring dialog box for the event.
functionName	For an event that the page widget subscribes to, the name of the function that handles the event.
type	For an event that the page widget publishes, the type of event. Set the type to Broadcast if the widget broadcasts the event to other widgets on the page. Set the type to Wiring if the event must be wired to another widget on the page.
direction	Set to either Subscribed or Published.
description	The description of the event is displayed in the hover help for the event in the Wiring dialog box.

Exercise 3.2.1: Create catalog and widget definition JSON files

Introduction

In this exercise, you develop the catalog and widget definition JSON files. You need these files to define your custom widget properties, and define any event broadcasting or event handling within your custom widget. These files are used to register the widget with the IBM Case Manager.

Procedures

Procedure 1: Create a folder for the registry files, page 22

Procedure 2: Edit the catalog JSON file, page 23

Procedure 3: Edit the widget definition JSON file, page 25

Procedure 1: Create a folder for the registry files

Sample widget definition and the catalog JSON files are included in the student system. In this procedure, you copy these files into your project and use them as a starting point.

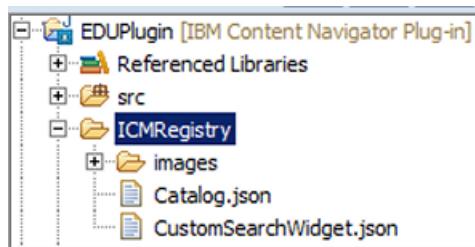
1. Copy the folder that contains the registry files into your project.
 - a. In Windows Explorer, go to the C:\ICM\Widgets\SearchWidget folder.
 - b. Right-click the ICMRegistry folder and select Copy.

The ICMRegistry folder contains the widget definition and the catalog JSON files. It also contains a folder with image files.

 - c. In Eclipse > Package Explorer, right-click your project and select Paste to add the ICMRegistry folder.



- d. Validate that your project folder looks as in the following screen capture.



Procedure 2: Edit the catalog JSON file



Hint

You can refer to the solution file at C:\ICM\Widgets\SearchWidget\Solution Files\Catalog_Solution.json. Optionally, you can use the JSON Viewer application shortcut on the Windows Desktop to view and edit the JSON files.

1. In Package Explorer, expand your project > ICMRegistry folder and double-click the file Catalog.json to open it.
Observe that the data fields of the custom widget package are not completed.
2. Enter the values for the fields with the data in the following table.

Section	Field	Value
General		
	Name	“EDU Custom Search Widget Package”
	Description	“IBM Case Manager custom search widget package”
	Locale	“”
	Version	“1.0”
Categories		
	id	“EDUWidgets”
	title	“EDU Widgets”
Widgets		
	id	“CustomSearchWidget”
	title	“Custom Search Widget”
	category	“EDUWidgets”
	description	“This widget does a custom search for cases”
	definition	“CustomSearchWidget.json”
	preview	“images/customSearch_preview.png”
	icon	“images/customSearch_icon.png”
	runtimeClassName	“eDUPuginDojo.CustomSearchWidget”
	help	“acmwrh126.htm”
	previewThumbnail	“images/customSearch_thumb.png”

3. Save your changes to the JSON file and leave the file open.

Information about the fields in the Catalog.json file:

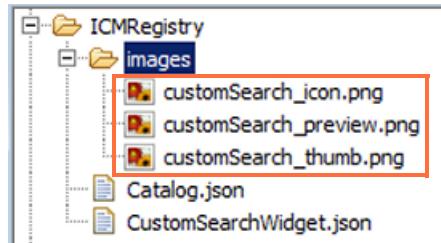
- The value of the Categories Title for the custom widget package is shown in the widget palette bar on the left pane in Page Designer.

```
"Categories": [
    {
        "id":"EDUWidgets",
        "title":"EDU Widgets"
    }
],
```

- ICMRegistry folder already contains the widget definition file. You specify the file name of the widget definition file (CustomSearchWidget.json) in the catalog.json.

```
"Widgets": [
    {
        "id":"CustomSearchWidget",
        "title":"Custom Search Widget",
        "category":"EDUWidgets",
        "description":"This widget does a custom search for cases",
        "definition":"CustomSearchWidget.json",
    }
]
```

- The image files for the preview, icon, and previewThumbnail for the custom widget are already copied into the ICMRegistry folder.



- You are going to create a custom widget file with the name: CustomSearchWidget.js, and add to the eDUPuginDojo folder in the next lesson.
 - You specify the custom widget file name for the runtimeClassName field.

```
"icon": "images/caseinfo icon.png",
"runtimeClassName": "eDUPuginDojo.CustomSearchWidget",
"help": "acmwrh126.htm",
```

Procedure 3: Edit the widget definition JSON file

1. In the ICMRegistry folder, double-click the file CustomSearchWidget.json to open it.
2. Notice that the first part of the file is similar to the widgets section of the Catalog.json file.
 - a. Use the same values that you entered for the Catalog.json file to complete the first 10 fields of the CustomSearchWidget.json file. You can also copy and paste them.
 - b. The completed file looks like the following screen capture.

```
1 {
2     "id": "CustomSearchWidget",
3     "title": "Custom Search Widget",
4     "category": "EDUWidgets",
5     "description": "This widget does a custom search for cases",
6     "definition": "CustomSearchWidget.json",
7     "preview": "images/customSearch_preview.png",
8     "icon": "images/customSearch_icon.png",
9     "runtimeClassName": "eDUPuginDojo.CustomSearchWidget",
10    "help": "acmwrh126.htm",
11    "previewThumbnail": "images/customSearch_thumb.png"
```

Procedure 4: Edit the Properties section of the CustomSearchWidget.json file

You can enter one or more properties to be shown in the settings window of your custom widget in Page Designer.

1. Set a PreferredHeight property for sizing of the custom widget.
 - a. Set the visibility of this property to false.
This step makes the property invisible to the user.
 - b. Set the id, disabled, required, visibility, and title fields as shown in the following screen capture.

```
"properties": [
  {
    "propertyType": "property",
    "type": "string",
    "id": "PreferredHeight",
    "defaultValue": "100%",
    "disabled": true,
    "required": false,
    "visibility": false,
    "title": "Preferred Height"
  },
]
```

2. Define the next property as a “Custom Property 1” and set the values with the data in the following table.

Field	Value
propertyType	“property”
type	“string”
id	“customProperty1”
defaultValue	“http://”
required	false
visibility	true
style	“width:95%;”
title	“Custom Property 1”

Procedure 5: Edit the Events section of the CustomSearchWidget.json file

The events section determines whether to handle or broadcast an event. It sets the function name to handle the event, if applicable.

1. For this exercise, set the id to `icm.SearchCases` as the event id to broadcast.
2. Enter a title and description. These fields do not require specific values.
3. Because the widget is broadcasting an event to another widget, set the direction field to broadcast.



Note

If you want to handle an event that is published from another widget, you set the direction field to `subscribed`. You specify the same function name in this JSON file as the one that is used in your widget file to handle this event. The Subscribed event is discussed in a later lesson.

4. Your completed events section must look like the following screen capture:

```
"events": [  
    {  
        "id": "icm.SearchCases",  
        "title": "EDU Custom Search Event",  
        "functionName": "",  
        "direction": "broadcast",  
        "description": "This is a custom search event"  
    }  
]
```

5. Save the changes to the JSON file and close both the files.



Hint

You can refer to the solution file at C:\ICM\Widgets\SearchWidget\Solution Files\CustomSearchWidget-Solution.json.

LESSON 3.3: Implement a custom widget

What this lesson is about

This lesson describes how to implement a custom widget. For this lab exercise, you use custom search as an example. You add the widget files in your IBM Content Navigator plug-in project that you created in a previous lesson.

What you should be able to do

After completing this lesson, you should be able to:

- Implement a custom widget by creating the JavaScript files.

How you will check your progress?

- Hands on labs.

Lab Solution files

- The solution files for this lesson are included in the following folder:
C:\ICM\Widgets\SearchWidget\Solution Files

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

IBM Case Manager custom page widget development

Introduction

- An IBM Case Manager page widget or an IBM Case Manager custom widget is a Dojo dijit.
- In addition, the Case Manager widgets also contain the following two more items:
 - Page container events
 - Widget settings
- These two facilities enable the page widget construction and allow business analyst to dynamically adjust page behavior without any (or a little) programming skills.
- You develop the widgets in two separate steps to reuse them more easily.
 - Dojo dijit - UI component
 - Page widget wrapper

Dojo dijit

- A Dojo dijit is a widget that is a part of Dojo User Interface (UI) library.
- When creating a page widget, first create a normal Dojo Dijit that is responsible for rendering the user interface and handling user interaction to collect user input.
- Design the dijit as a reusable component that takes in a context and renders itself.
- The UI dijit must not access any IBM Case Manager Page Container related APIs.
 - Accessing these APIs limits its usage to only the IBM Case Manager Case Manager Client.
 - Separation of the UI dijit from the Page widget layer allows you to reuse this dijit in a custom application or plug-in that is built on top of IBM Content Navigator.
- The dijit can be reused in a non-ECM environment if it doesn't use any Enterprise Content Manager related API.
- The dijit must contain necessary Dojo methods and events:
 - The methods and events allow the page widget wrapper or other custom widgets to exchange data or get notified by user interaction.

Page widget wrapper

After completing the UI part of the page widget, you must create a Page widget wrapper that extends this UI dijit and converts it into an IBM Case Manager page widget.

- The page widget wrapper contains the page container-related code.
 - It accepts widget settings from the page container and passes them to the underlying dijit.
 - It must also connect to Dojo events in the dijit to be notified of user actions.

- Optionally, you can separate the event handling logic from the page widget when you have a more complex event handling.
 - If you have complex logic to pre-process or post-process the event data, you can create a separate event listener object that holds the event handler implementations.
 - Then, the page widget might delegate the event handling logic to the event listener object.

Dojo Asynchronous Module Definition (AMD) loading

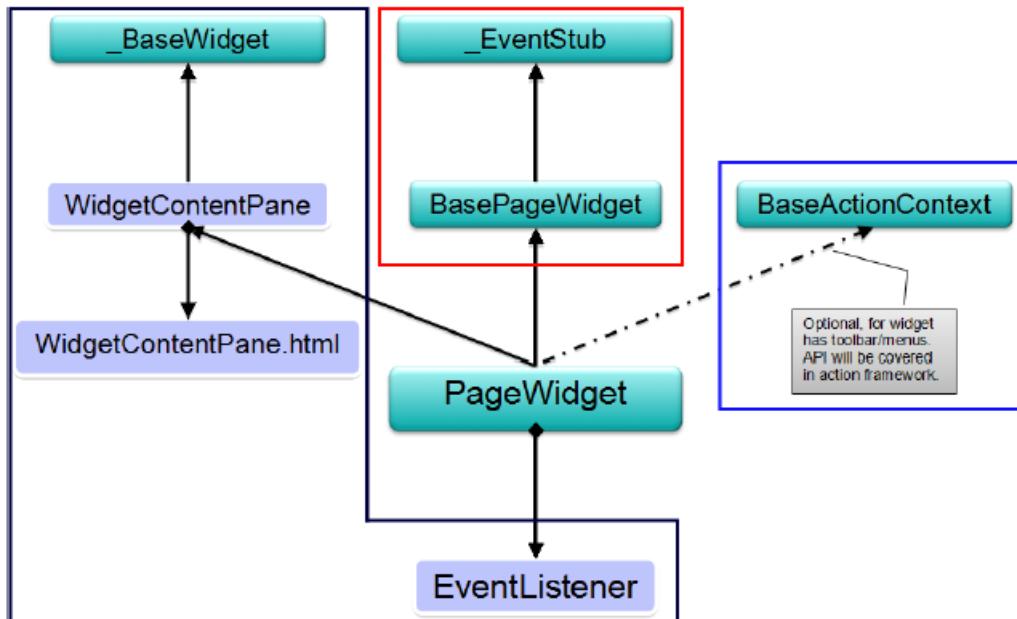
Dojo AMD loading allows the application developer to load different classes to use in the custom widget.

- The Dojo loader includes the AMD APIs.
 - Use case: To define and load the IBM Case Manager or the IBM Content Navigator JavaScript API classes.
-

Implement a page widget

Structure for a page widget

The following graphic shows the structure for a page widget:



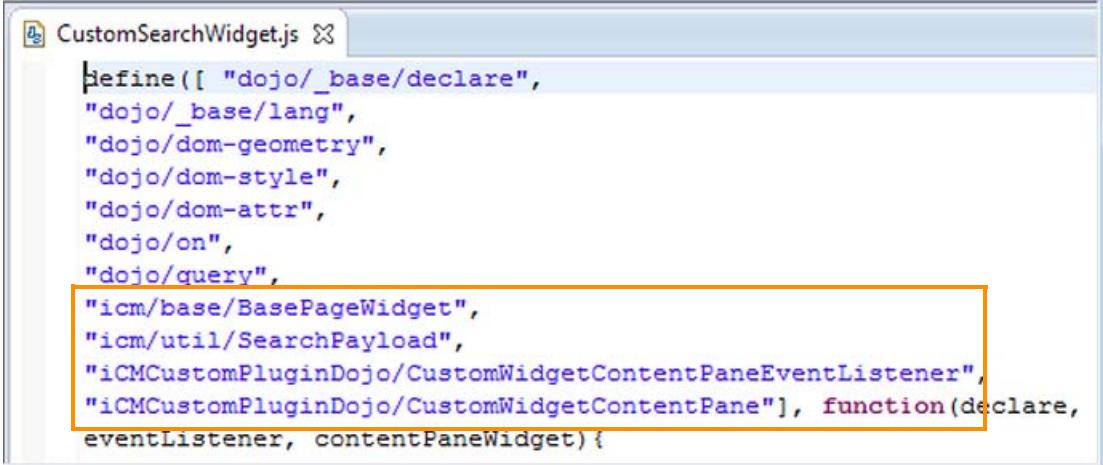
- In the graphic, the classes in the **black** box implement the user interface for the page widget.
 - It handles the interaction of a user with the interface.
- The classes in the **red** box are base IBM Case Manager classes from which the custom page widget inherits functions.
 - The icm.base._BaseWidget class, which provides functions that display the widget description and show or hide the content pane.
 - The icm.base._EventStub class provides functions for publishing and broadcasting methods.
- The class in the **blue** box is the base class for any page widget that hosts a toolbar or a menu.

Implement a custom page widget

Following are the high-level steps to implement a custom widget.

1. Create a JavaScript file that implements the WidgetContentPane class.
 - The file is used to create the user interface for the CustomPageWidget widget.
 - The WidgetContentPane class Inherits from the icm.base._BaseWidget class.

- The class implements a destroy function that cleans up the user interface for the page widget when the page is closed.
2. Create a JavaScript file that implements the custom page widget.
- The file inherits from the following classes:
 - CustomWidgetContentPane
 - BasePageWidget
 - BaseActionBarContext (optional, it is required for implementing toolbars and menus)



```
define([
    "dojo/_base/declare",
    "dojo/_base/lang",
    "dojo/dom-geometry",
    "dojo/dom-style",
    "dojo/dom-attr",
    "dojo/on",
    "dojo/query",
    "icm/base/BasePageWidget",
    "icm/util/SearchPayload",
    "iCMCustomPluginDojo/CustomWidgetContentPaneEventListener",
    "iCMCustomPluginDojo/CustomWidgetContentPane"], function(declare,
    eventListener, contentPaneWidget) {
```

3. Write code to add a custom search function for the custom page widget.
4. If the widget subscribes an event from other widgets, add an event handler for the custom page widget.

Files that are used in this lesson

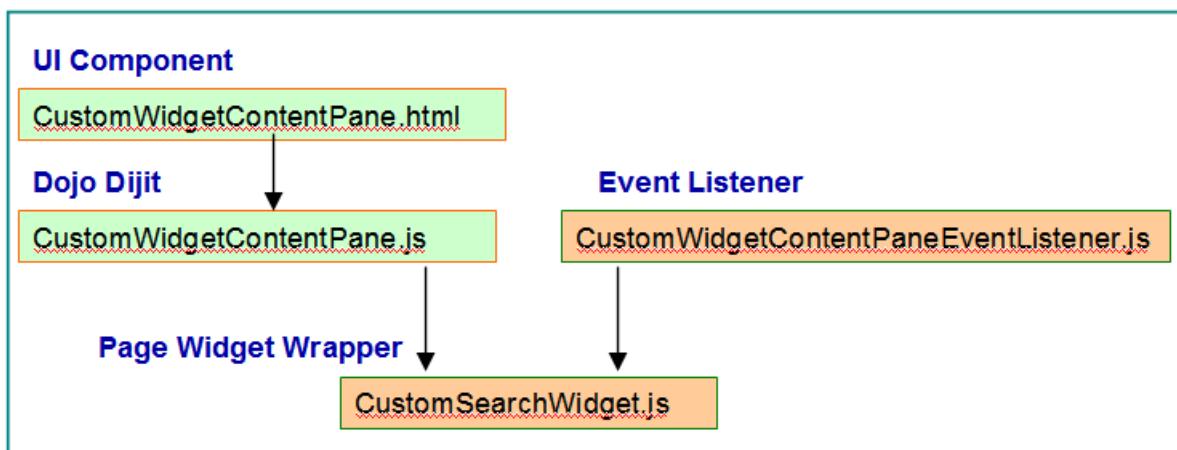
Introduction

To create a page widget, you use the following files:

- **CustomWidgetContentPane.js**
 - A Dojo dijit that implements the User Interface (UI) for your custom page widget.
 - Responsible for rendering the UI, and handling user interaction to collect user input.
 - Calls an event stub method when users click the submit button.
 - The CustomSearchWidget connects to event stub method, constructs the page widget event, and sends it out.
- **CustomWidgetContentPane.html**
 - The Dijit HTML content pane for the custom widget.
 - Add "div" tags or setup HTML to display when your widget renders.
- **CustomSearchWidget.js**
 - A Page widget wrapper
 - Extends the UI dijit and converts it into an IBM Case Manager page widget.
- **CustomWidgetContentPaneEventListener.js**
 - Contains the logic to pre-process or post-process the event data.
 - Separates the complex logic from the page widget wrapper class
 - Builds a custom search for cases for this lesson.

Relationship diagram

The following diagram shows how the four files are connected.



Exercise 3.3.1: Implement a custom widget

Introduction

The files for this lab exercise are already created on the student image. You copy the files into your project, use them as a starting point and edit the code to complete the custom widget project. You use custom search function as an example.

Procedures

Procedure 1: Copy the files into your project, page 34

Procedure 2: Set up the Dijit HTML content pane, page 35

Procedure 3: Edit the CustomWidgetContentPane.js file, page 36

Procedure 4: Edit CustomWidgetContentPaneEventListener.js, page 38

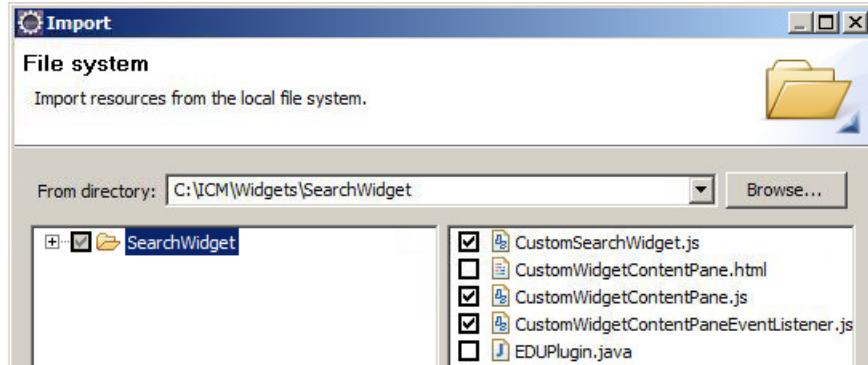
Procedure 5: Create the model object and broadcasting the payload, page 39

Procedure 6: Add Dojo AMD style loading, page 40

Procedure 7: Edit the base JavaScript file, page 41

Procedure 1: Copy the files into your project

1. In Eclipse > Package Explorer, expand your project > `src` node and right-click `com.ibm.icm.edu.WebContent.eDUPuginDojo`
 - a. Click Import from the list.
2. In the Import page, select General > File System and click Next.
 - a. In the Import > File system page, click Browse.
 - b. In the “Import from directory” page, go to `C:\ICM\Widgets\SearchWidget` folder and click OK.
 - c. Back in the Import > File system page, select the following files:
 - `CustomSearchWidget.js`
 - `CustomWidgetContentPane.js`
 - `CustomWidgetContentPaneEventListener.js`
3. Make sure that the “Into folder” field has the following value:
`EDUPugin/src/com/ibm/icm/edu/WebContent/eDUPuginDojo`
 - a. Select the “Overwrite existing resources without warning” option.



- b. Click Finish.
4. Repeat Steps 1-3 to copy the CustomWidgetContentPane.html file into the EDUPlugin/src/com/ibm/icm/edu/WebContent/eDUPuginDojo/templates package.

**Hint**

Notice that your project shows errors; It is because the files that you copied are not completed files. In the following procedures, you complete the code for these files.

Procedure 2: Set up the Dijit HTML content pane

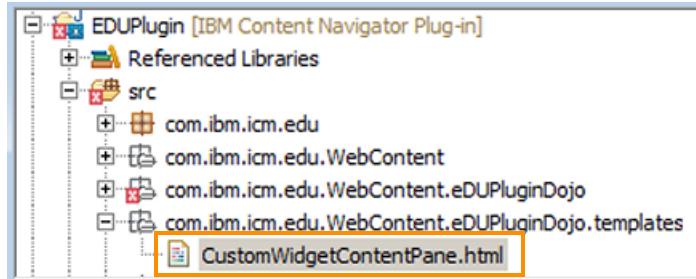
In this procedure, you set up the Dijit HTML content pane for your custom widget.

**Note**

The content pane is where you add "div" tags or setup HTML for display when your widget renders. This HTML content pane is connected to the Dijit content pane class (CustomWidgetContentPane.js) that retrieves the solution attributes and displays them on the content pane.

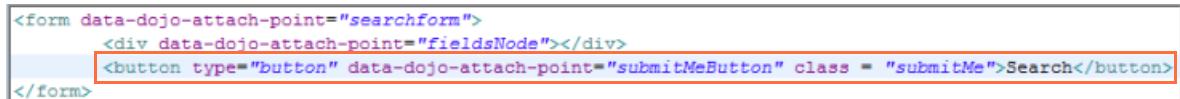
1. In Package Explorer, expand your project > src > com.ibm.icm.edu.WebContent.eDUPuginDojo.templates
2. Delete the wizard generated file: ConfigurationPane.html. This file is not used for this lab.
 - a. Right-click the file and click Delete from the list. Click OK to confirm the delete.
3. Double-click the CustomWidgetContentPane.html file to open it.

You dynamically populate the data in this HTML file to render your custom search widget.



4. Add a submit button to detect that the user wants to search on the user-defined values for the variables.
 - a. You add this button underneath the "div" tag labeled fieldsNode.

```
<button type="button" data-dojo-attach-point="submitMeButton" class = "submitMe">Search</button>
```



5. Save the CustomWidgetContentPane.html file and close it. Leave Eclipse open.



Hint

You can refer to the completed file at C:\ICM\Widgets\SearchWidget\Solution Files\CustomWidgetContentPane-Solution.html.

Procedure 3: Edit the CustomWidgetContentPane.js file

You load the solution attributes dynamically into the custom search widget to display the case properties for users to enter the criteria for search. In this procedure, you make a Model API call in the CustomWidgetContentPane.js file to complete the task.

- See “IBM Case Manager JavaScript API Reference” for details about the icm.model.Solution class.
- The Solution class contains the retrieveAttributeDefinitions(callback)function.
 - This method provides the information about the case properties of the solution.
 - The case properties are shown in the interface for the users to search upon them.

[icm.model.ResultSet](#)
[icm.model.Solution](#) **icm.model.Task**
[icm.model.TaskEditable](#)
[icm.model.TaskType](#)
[icm.model.Timeline](#)

retrieveAttributeDefinitions(callback)

Retrieves attribute definitions that correspond to all attributes of any case types of the solution. This information is cached so that the callback function can be called right away.

Parameters:**callback**

a function called with an array of ecm.model.AttributeDefinition objects

1. In Package Explorer, expand your project > src > com.ibm.icm.edu.WebContent.eDUPuginDojo
2. Delete the wizard generated file: ConfigurationPane.js. This file is not used for this lab.
 - a. Right-click the file and click Delete from the list. Click OK to confirm the delete.
3. Double-click the CustomWidgetContentPane.js file to open it.
 - a. Add the following line of code inside the createSearchFields function (in line 20).
solution.retrieveAttributeDefinitions(lang.hitch(this, this._buildSearchFields));
4. After the Model API call with retrieveAttributeDefinitions(callback), observe that these attribute definitions are pushed into the next function in CustomWidgetContentPane.js named _buildSearchFields(attrs).
 - In this lab exercise, the id and name of each attribute are used for the output.
 - The output is directly written to the content pane inner HTML.

```
56     });
57     fields.push("</table>");
58     this.fieldsNode.innerHTML = fields.join("");
59
60 },
```

5. Notice that the CustomWidgetContentPane.html file that you edited in the previous procedure is referenced at the top of the file.
6. Observe the code in the postCreate function in CustomWidgetContentPane.js.
 - The user-inputted values for each attribute are sent to the CustomWidgetContentPaneEventListener to construct the searchPayload and the Model object.
 - You are going to edit the Listener in the next procedure.
7. Save and close the file.

Procedure 4: Edit CustomWidgetContentPaneEventListener.js

To run a case search, you call the `setModel(model)` function and set the schema and data structures. In this procedure, you construct the search parameters to pass them to the `setModel(model)` function.

- See “IBM Case Manager JavaScript API Reference” for details about the `icm.util.SearchPayload` class.
 - The `SearchPayload` class contains the `setModel(model)` function.

`setModel(model)`

Set the schema and data structures associated with the current search. This is called when a case search action is performed and before the query is executed.

Parameters:

`model`

The search parameters used to construct the payload.

- `criterions`: search criterions that will be used in the `SearchTemplate`
- `objectStore`: object store to search against
- `caseType`: case type for single case type search
- `solution`: solution that the search is run against
- `ceQuery`: {optional} A user provided Content Engine query that is used to search on cases. When `ceQuery` is set, the `criterions` parameter will be ignored.

Deprecated Unless necessary, `criterions` should be used.

1. In Package Explorer, expand your project > src > `com.ibm.icm.edu.WebContent.eDUPuginDojo`
 - a. Double-click the `CustomWidgetContentPaneEventListener.js` file to open it.
 - b. Edit the file to add code in the following steps.
 - c. You define the `objectStore` and `caseType` parameters for the `Model` object for the search payload.
2. Define the object store:
 - a. Add the following code in the `buildPayload(values)` function at line 44.

```
params.ObjectStore = solution.getTargetOS().id;
```
3. Define the case type for the search.
 - a. Get the first case type defined in the solution.
 - b. Add the following code at line 63 inside the `this.widget.solution.retrieveCaseTypes(function(types))` method.

```
params.caseType = types && types.length > 0 && types[0].name;
```

This code defaults to the first case type.
 - c. The solution is already defined after the case type at line 65.

```
params.solution = solution;
```
4. Save and leave the file opened for the next procedure.

Procedure 5: Create the model object and broadcasting the payload

The following steps are required to accomplish search function in the custom widget:

- Set variable params with the Target Object Store, solution, case type, and ceQuery.
- Pass the variable params to `setModel` function that sets the Model object.
- Call the `getSearchPayload` function (of the `icm/util/SearchPayload` class), which conducts the search and retrieves the payload to send.
- Load a variable with the payload from `getSearchPayload`.
- Broadcast the payload that is received to Case List widget.



Note

You set the variable params in the previous procedure.

1. Pass the variable params to the `setModel` function.
 - a. In the `CustomWidgetContentPaneEventListner.js` file, add the following code at line 68 inside the `this.widget.solution.retrieveCaseTypes(function(types))` method.

```
searchPayload.setModel(params);
```
2. Observe the rest of the code in the file that does the following steps:
 - Calls the `getSearchPayload()` function and pass the payload that is received into a variable.
 - Broadcasts the payload out with the event `icm.SearchCases`.
 - The Case List widget can handle the `icm.SearchCases` incoming event and the payload to display the results.

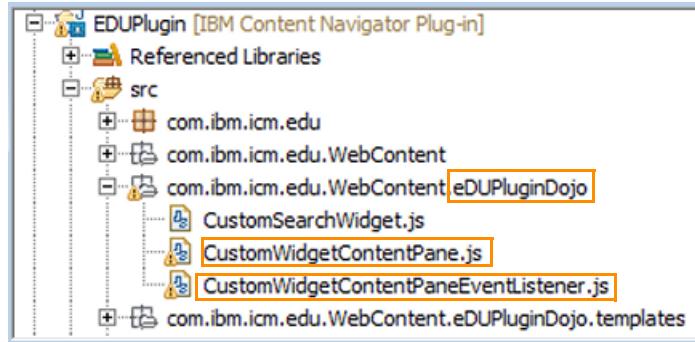
```
//getSearchPayload function and the onBroadcastEvent
var w = that.widget;
searchPayload.getSearchPayload(function(payload) {
    w.onBroadcastEvent("icm.SearchCases", payload);
    console.log(payload);
    that.displayPayload(payload);
});
```

3. Save and close the `CustomWidgetContentPaneEventListner.js` file.

Procedure 6: Add Dojo AMD style loading

In this procedure, you define the JavaScript files that you edited in the previous procedures with Dojo AMD loading in CustomSearchWidget.js.

1. In Package Explorer, expand your project > src > com.ibm.icm.edu.WebContent.eDUPuginDojo
 - a. Double-click the CustomSearchWidget.js file to open it.
 - b. Edit the file to add code in the following steps.
2. Observe that the Dojo libraries (dojo/dom-attr and dojo/query) are already loaded.
 - a. The IBM Case Manager JS files (icm/base/BasePageWidget and icm/util/SearchPayload)are referenced.
3. In Package Explorer, note the paths for CustomWidgetContentPaneEventListener.js and CustomWidgetContentPane.js.



- a. Define the files in CustomSearchWidget.js. The functions in these classes become available in CustomSearchWidget.js.
- b. Complete the last two empty quotations in the define header:
"eDUPuginDojo/CustomWidgetContentPaneEventListener",
"eDUPuginDojo/CustomWidgetContentPane"

```
1 define([
2   "dojo/_base/declare",
3   "dojo/_base/lang",
4   "dojo/dom-geometry",
5   "dojo/dom-style",
6   "dojo/dom-attr",
7   "dojo/on",
8   "dojo/query",
9   "icm/base/BasePageWidget",
10  "icm/util/SearchPayload",
11  "eDUPuginDojo/CustomWidgetContentPaneEventListener",
12  "eDUPuginDojo/CustomWidgetContentPane"],
```

4. Add the two JS file definitions into the `function` method at line 12.

 Note

For every entry in the “define” section, you must have a matching entry in the “function”. The “define” defines modules, and the name in the “function” is a mapping to the same field in the define. Both are required.

```
function(declare, lang, domGeom, domStyle, domAttr, on, query, BasePageWidget, SearchPayload,  
eventListener, contentPaneWidget){
```

5. Save and close the CustomSearchWidget.js file.

Procedure 7: Edit the base JavaScript file

The base JavaScript file is loaded when IBM Content Navigator loads the plug-in. You use this JavaScript file to load any JavaScript classes that must be available throughout the session.

1. In Package Explorer, expand your project > src > com.ibm.icm.edu.WebContent.
 - a. Double-click the `EDUPlugin.js` file to open it.
 - b. Edit the file to add code in the following steps.
2. Observe that the wizard generated this file with sample code.
 - a. Edit the file to load `CustomSearchWidget.js` as shown in the following code:

```
require([ "eDUPluginDojo/CustomSearchWidget" ],  
function(CustomSearchWidget) {  
});
```
3. Save and close the `EDUPlugin.js` file.

 Note

You package, register, and test your widget package in the next lesson.

LESSON 3.4: Build and register a widget package

What this lesson is about

This lesson provides guidance for building and registering the widget package. You create a compressed file that contains the Registry folder and the plug-in JAR file, register with IBM Content Manager and test your widget.

What you should be able to do

After completing this lesson, you should be able to:

- Build and register the widget package

How you will check your progress?

- Hands on labs.

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Register the custom widget package

Contents in the custom page widget package

A Case Manager custom widget compressed file (ZIP) must include the following items:

- IBM Content Navigator plug-in JAR file
 - If you have a small widget customization, you can add the page widget code as part of the plug-in JAR file. No need for a separate EAR file.
- Optional EAR file
 - If you have a large custom application, then you package the custom widget code as an EAR file separately.
- Custom widget definition JSON file (in a folder)
- Custom widget catalog JSON file (in a folder)

Options for widget package deployment and registration

Use one of the following two options for deploying and registering your custom widget package. The option depends on how you packaged the widgets:

- Option 1 - Widget package with a JAR file:
 - If you have a JAR and not the EAR file in the widget package, you must register it in the IBM Case Manager administration web client.



Note

Lessons 1-4 in this unit provides details for the JAR file approach.

- Option 2 - Widget package with an EAR file:
 - If you include an EAR file in the widget package, you must deploy and register it in IBM Case Manager configuration tool.



Note

Lessons 5 and 6 in this unit provides details for the EAR file approach.

Exercise 3.4.1: Build and register the widget package

Introduction

- The wizard-generated `build.xml` file packages the Content Navigator plug-in files into a JAR file.
- To register a Case Manager widget, you require a compressed file (ZIP) that includes the plug-in JAR file and the ICMRegistry folder that contains the catalog and widget definition files.
- In this exercise, you create the ZIP file.

Procedures

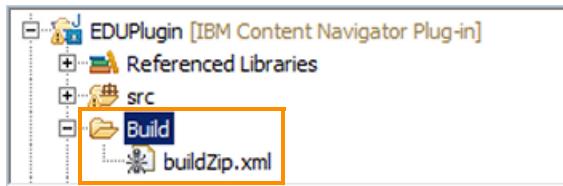
Procedure 1: Package the widget files into a ZIP file, page 44

Procedure 2: Register the custom widget, page 46

Procedure 1: Package the widget files into a ZIP file

The `buildZIP.xml` file that packages your project into a ZIP file is included in the student system. In this procedure, you copy the file into your project and generate the package.

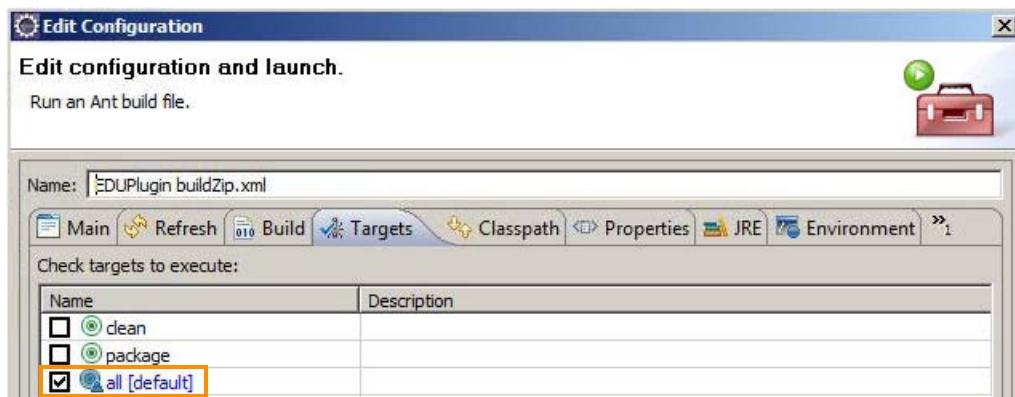
1. In Package Explorer, expand your project.
 - a. Optionally, open the `build.xml` file to observe the contents of the file.
2. Copy the folder that contains the `buildZIP.xml` file into your project.
 - a. In Windows Explorer, go to the `C:\ICM\Widgets\SearchWidget` folder.
 - b. Right-click the `Build` folder and select Copy.
 - c. In Eclipse > Package Explorer, right-click your project and select Paste.
 - d. Validate that your project folder looks as in the following screen capture.



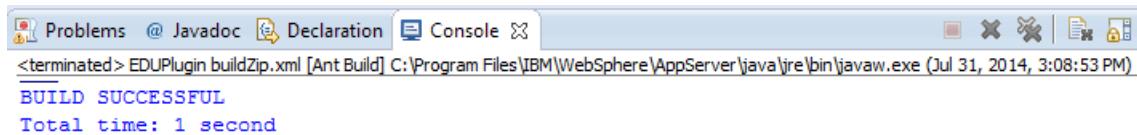
3. In Package Explorer, expand your project > Build.
 - a. Open the `buildZIP.xml` file and observe the contents of the file.
This file calls the `build.xml` that creates the JAR file; then, it packages the JAR file and the ICMRegistry folder into a ZIP file.
 - b. Right-click the `buildZIP.xml` and select Run As > "2.Ant Build" from the list.



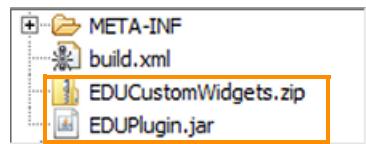
4. In the “Edit Configuration” window, verify that the “all [default]” option under the Name column is selected.
 - a. Click Run at the end of the page.



5. Validate that the console tab at the bottom pane displays that the Build was successful.
 - a. Verify that the message indicates that the JAR, and ZIP files are created.
 - b. Ignore the “includeantruntime” warning.



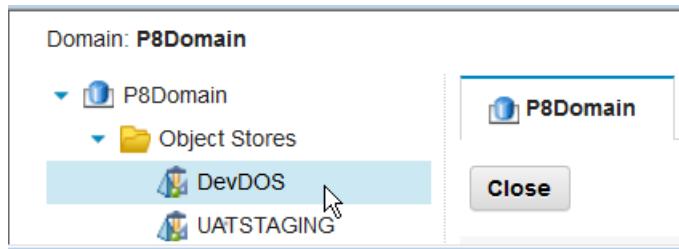
6. Right-click your project (EDUPlugin) and select Refresh from the list.
 - a. Verify that the JAR and ZIP files are listed in your project.



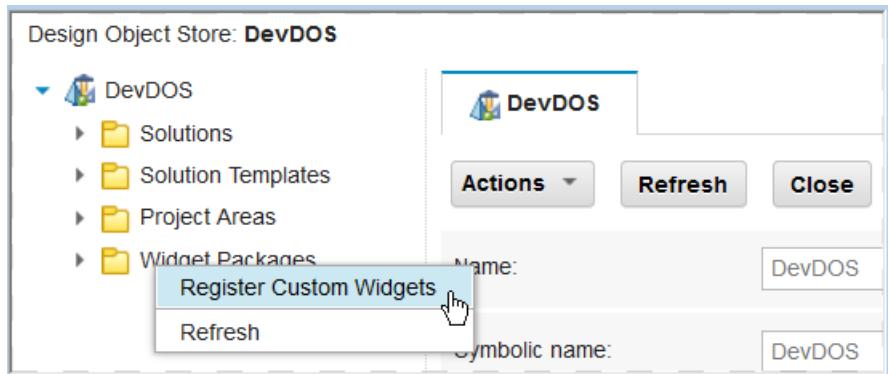
Procedure 2: Register the custom widget

In this procedure, you register your custom widget package in the IBM Case Manager admin tool.

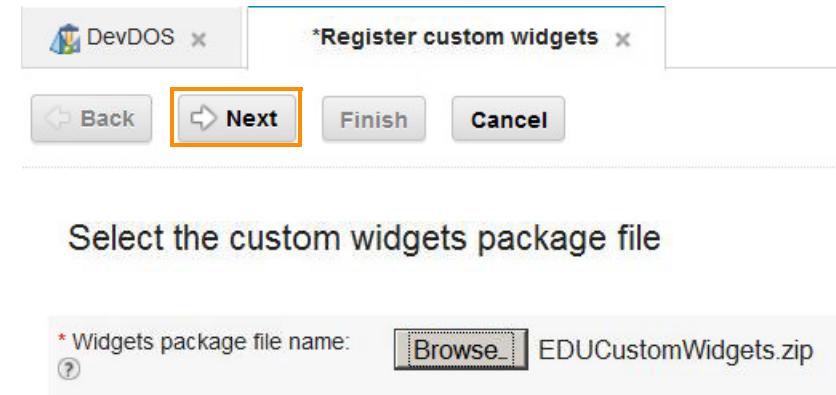
1. Start the IBM Case Manager administration client.
 - URL: <http://ecmedu01:9080/navigator/?desktop=icmadmin>
 - User name: P8admin
 - Password: IBMFileNetP8
2. Open the Design Object Store.
 - a. Click P8Domain > Object Stores > DevDOS in the left pane.
The DevDOS tab opens.



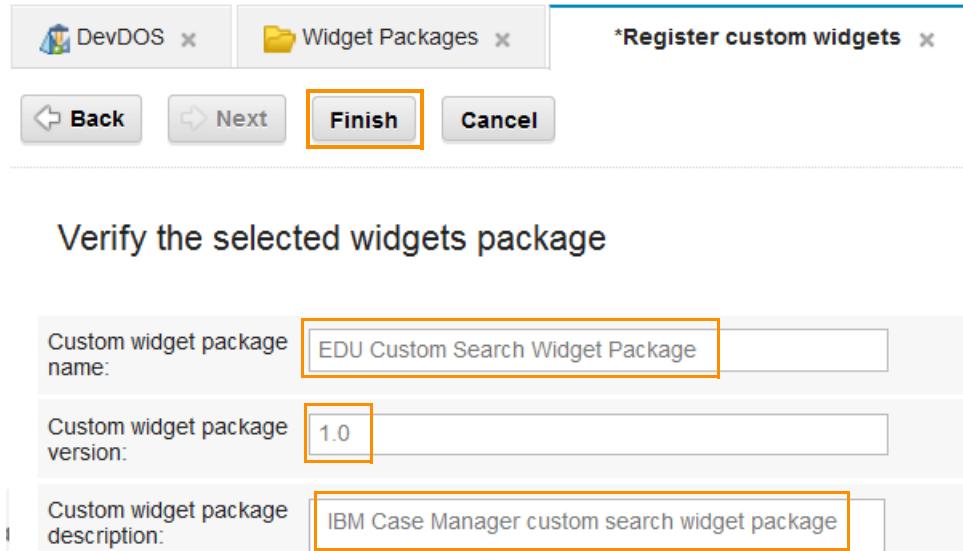
3. In the DevDOS tab > left pane, expand the DevDOS.
 - a. Right-click "Widget Packages" and select "Register Custom Widgets" from the list.
The "Register custom widgets" tab opens.



4. In the "Register custom widgets" tab, click Browse.
 - a. Go to the folder where the ZIP file for your widget package is created:
C:\ICM\workspace_Eclipse\EDUPlugin
 - b. Select `EDUCustomWidgets.ZIP` and click Open.
 - c. Verify that the Zip file is added to the "Widget package file name" field and click Next.



5. In the “Verify the selected widgets package” page, check the widget package name, version, and the description.



- a. Recall that the same values that you added to your Catalog.json file as shown in the following screen capture (Lesson 2 in this unit).

```
1 {
2     "Name": "EDU Custom Search Widget Package",
3     "Description": "IBM Case Manager custom search widget package",
4     "Locale": "",
5     "Version": "1.0",
```

- b. Click Finish.
6. When you get the “The widget package was successfully registered” (at the end of the page), click Close.

7. Click “Widget Packages” node from the left pane.
 - a. Check that your widget package is listed in the “Widget Packages” tab.

The screenshot shows the DevDOS interface with the "Widget Packages" tab selected. The top navigation bar includes "Actions", "Refresh", and "Close" buttons. Below is a table with columns: Name, Version, and Description. Two rows are visible: one for "EDU Custom Search Widget Package" (Version 1.0) and another for "IBM Case Manager Widget package" (Version 5.2). The first row is highlighted with a yellow border.

	Name	Version	Description
	EDU Custom Search Widget Package	1.0	IBM Case Manager custom search widget package
	IBM Case Manager Widget package	5.2	IBM Case Manager Widget package

8. Log out of the IBM Case Manager Client.

Procedure 3: Verify the Content Navigator plug-in registration

1. Start the IBM Content Navigator Administration Desktop.
 - URL: <http://ecmedu01:9080/navigator/?desktop=admin>
 - User name: P8admin
 - Password: IBMfileNetP8
2. In the IBM Content Navigator Admin desktop, select Plug-ins in the left pane.
 - a. In the Plug-ins tab, verify that your plug-in is listed.

The screenshot shows the IBM Content Navigator Administration Desktop. The left sidebar has nodes: Desktops, Repositories, Settings, **Plug-ins**, Viewer Maps, Menus, and Labels. The right panel has tabs: Desktops and **Plug-ins**. A message says: "You must use the administration tool to register plug-ins for the web client. If a plug-in's parameters are displayed after you register the plug-in." An important note states: "Important: If you edit a plug-in that is referenced in another area of the administration, behavior of the plug-in." Below is a table with columns: Name and Version. Two entries are listed: "EDU ICN Plugin" (Version 1.0) and "IBM Case Manager API plug-in" (Version 5.2.0.1 (icmapi5.2.0.001.047)). The "EDU ICN Plugin" entry is highlighted with a yellow border.

Name	Version
	1.0
	5.2.0.1 (icmapi5.2.0.001.047)

3. Recall that the same values that you added to your EDUPlugin.java file as shown in the following screen capture (Lesson 1 in this unit).

```
EDUPlugin.java
1 package com.ibm.icm.edu;
2
3 import java.util.Locale;
4
5 public class EDUPlugin extends Plugin {
6
7     public String getId() {
8         return "EDUPlugin";
9     }
10
11
12     public String getName(Locale locale) {
13         return "EDU ICN Plugin";
14     }
15
16
17     public String getVersion() {
18         return "1.0";
19     }
}
```

4. Verify that the custom plug-in configuration:
 - a. In the Plug-ins tab, select your plug-in (EDU ICN Plugin) and click Edit.
 - b. In the EDU ICN Plugin tab, click Load for the JAR file path.
 - c. Verify that the details about the plug-in are displayed as shown in the screen capture.

Plug-in: EDU ICN Plugin

A plug-in can be either a JAR file or a compiled class file.

Important: The IBM Content Navigator web application server must be able to access the plug-in file on the local file system or through a URL.

<input checked="" type="radio"/> JAR file path <small>?</small>	C:\Program Files (x86)\IBM\CaseManagement\configure	<input type="button" value="Load"/>
<input type="radio"/> Class file path: <small>?</small>	<input type="text"/>	<input type="button" value="Load"/>
Class name: <small>?</small>	<input type="text"/>	
Name:	EDU ICN Plugin	
Version:	1.0	
Actions:	None	

5. Click Save and Close.
6. Log out of the admin tools and close the browser.

Exercise 3.4.2: Test the custom widget

Introduction

In this exercise, you create a custom Solution Page in Case Manager Builder, add your custom widget to the page, and test it.

Procedures

Procedure 1: Create a custom page, page 50

Procedure 2: Edit the page to add the custom widget, page 51

Procedure 3: Assign the custom page to a role, page 52

Procedure 4: Redeploy the solution, page 52

Procedure 5: Test the custom search widget, page 53

Procedure 1: Create a custom page

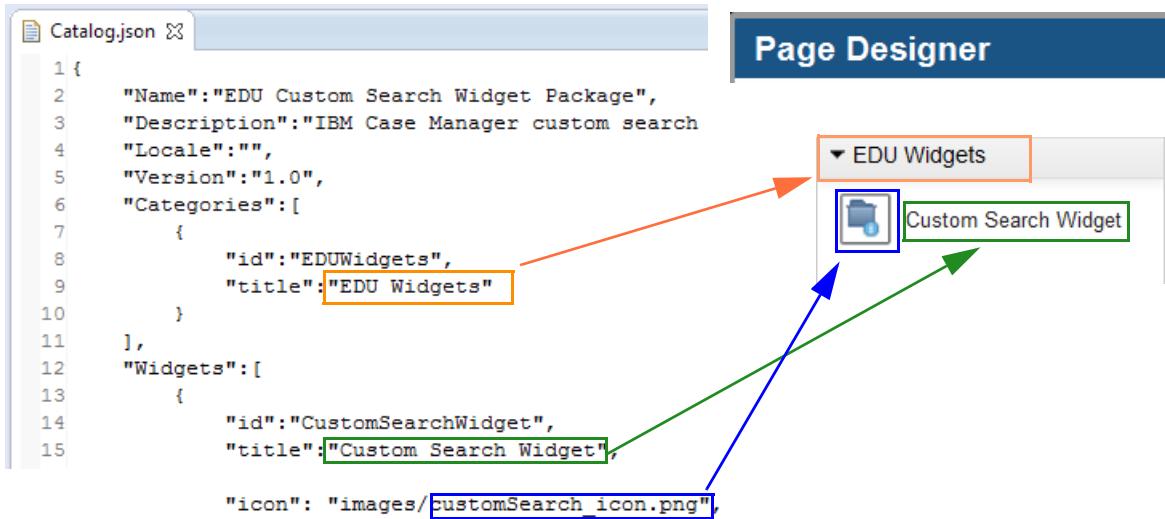
A solution for this lab is already created. In this procedure, you copy the default Cases page and modify it.

1. In Firefox, log on to IBM Case Manager Builder as an administrative user.
 - URL: <http://ecmedu01:9080/CaseBuilder>
 - User name: P8admin
 - Password: IBMFileNetP8
2. In Case Manager Builder, open Lab Claims Solution by clicking the Edit link.
Hover the mouse over the solution to see the links.
3. Create a custom page:
 - a. Open the Pages tab and expand the Solution Pages.
 - b. Hover the mouse over the Cases page name.
 - c. Select the Copy icon on the right side of the page.
 - d. In the resulting page, edit the name to Custom Search for your new page and click OK to create the copy.
 - e. Save your work by clicking Save at the top of the page.
4. Leave the Pages tab opened for the next procedure.

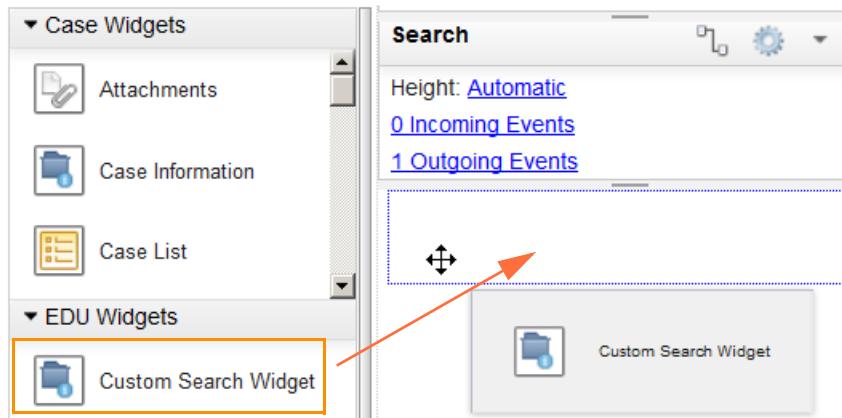
Procedure 2: Edit the page to add the custom widget

In this procedure, you edit the page to add your custom search widget.

1. In the Pages tab, double- click your page (Custom Search) to edit it in Page Designer.
2. Observe that the title of the widget category (“EDU Widgets”) is displayed in the left pane of the Page Designer.
 - a. The title of the widget that you added is displayed under the widget category in which it is defined. The widget is available to add to the pages.
 - b. Recall the values that entered in the catalog.json file in your widget package.



3. Drag the “Custom Search Widget” (under the “Custom Widgets” section) from widget palette on the left column to the page on the right.
 - a. Place the widget below the existing default Search widget.



4. Click Save to save your work.
 - a. Click Save and then Close to close Page Designer.
 - b. Leave the solution open for the next procedure.

Procedure 3: Assign the custom page to a role

1. Your solution is already opened in the Case Manager Builder. Open the Roles tab.
2. Click the Customer Service Rep role link.
 - a. Open the Pages subtab.
3. Optional: Remove the “Work Bench” page that you created in the previous unit.
 - a. Select the page, hover over, and click the Remove (trash can) icon.



Note

Step 3, removes only the association of the page with a role. The actual page (Work Bench) that you created is still available if you need to use it. The removal step helps to minimize the number of tabs in the Case Manager Client.

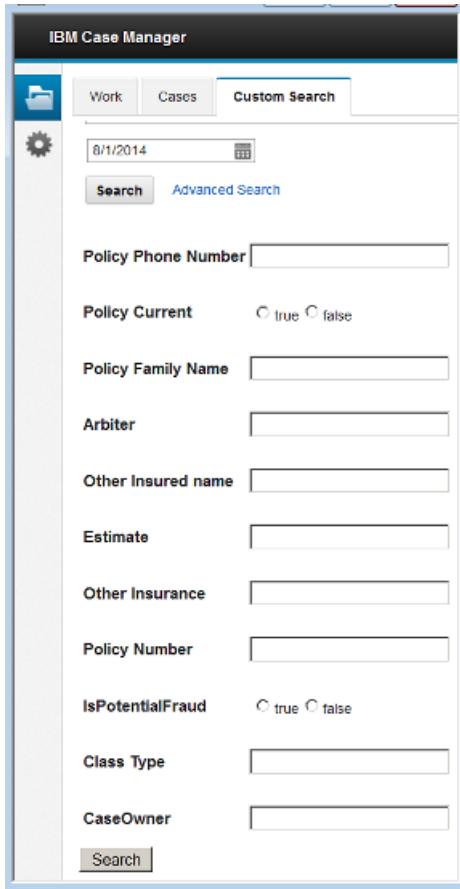
4. Assign the new page that you created.
 - a. Click Assign Page.
 - b. Select the “Custom Search” Page.
 - c. Click OK to close the dialog window.
5. Verify that your page is listed in the Pages tab.
 - a. Click OK All to accept the changes to the role.
 - b. Click “Save and Close” at the top of the page to exit the solution editor.
 - c. Leave the Case Manager Builder open for the next procedure.

Procedure 4: Redeploy the solution

1. In the Manage Solutions page, select Lab Claims Solution and hover the mouse over.
 - a. Click Deploy.
 - b. In the Confirmation page, make sure that “Commit my changes and make them available for deployment” is selected, and click Deploy.
 - c. Wait for the green check mark to appear next to the solution.
2. Select Lab Claims Solution and hover the mouse over again.
 - a. Click Test to open Case Manager Client.

Procedure 5: Test the custom search widget

1. In the Case Manager Client, select the “Custom Search” tab to open your custom page.
2. Verify that the custom Search fields are shown.

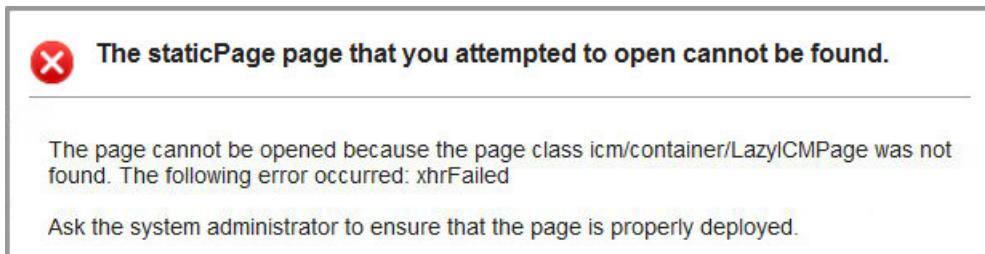


3. Search for cases.
 - a. In the Custom Search widget section, enter % in the text box for the Policy Family Name field and click the Search button.
 - b. Verify that available cases are listed in the Case List widget.
4. Optionally, enter a specific Policy Family Name (Example: Smith) and verify that it returns only one case that case the Value Smith.
5. Log out of applications and close the browser.

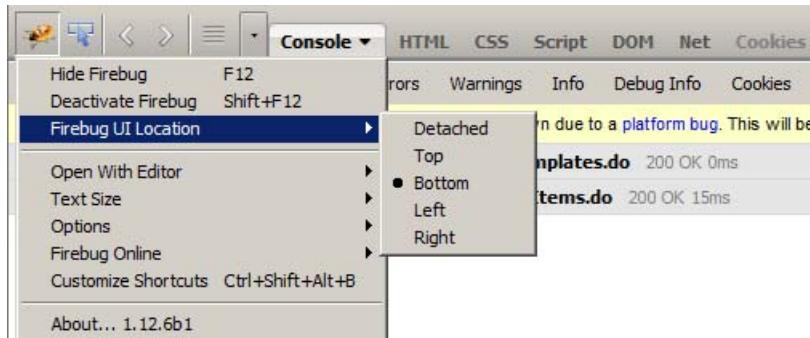
Exercise 3.4.3: Troubleshooting (if the widget does not work)

Page not found error

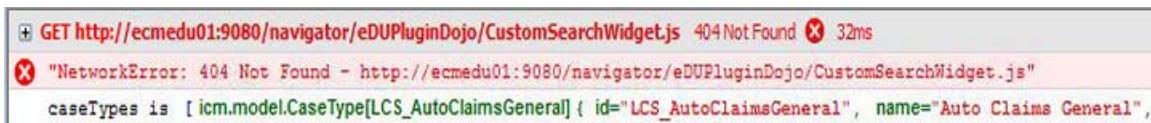
After you implemented the widget, and when you test it in Case Manager Client, one of the errors might look like the following screen capture.



1. Press F12 to launch the Firebug in the browser.
 - a. In the Firebug options, you can set it to any location of the browser or separately in an individual window.



- b. Refresh the browser and open your page again.
2. Verify the error in Firebug. One of the errors might look like the following screen capture:



3. Do the Procedure 3: Verify the Content Navigator plug-in registration, page 48 to make sure if your widget registration is successful.
4. Log out of the applications and close the browser to clear the cache.
 - a. Test the widget again.

General troubleshooting steps



Troubleshooting

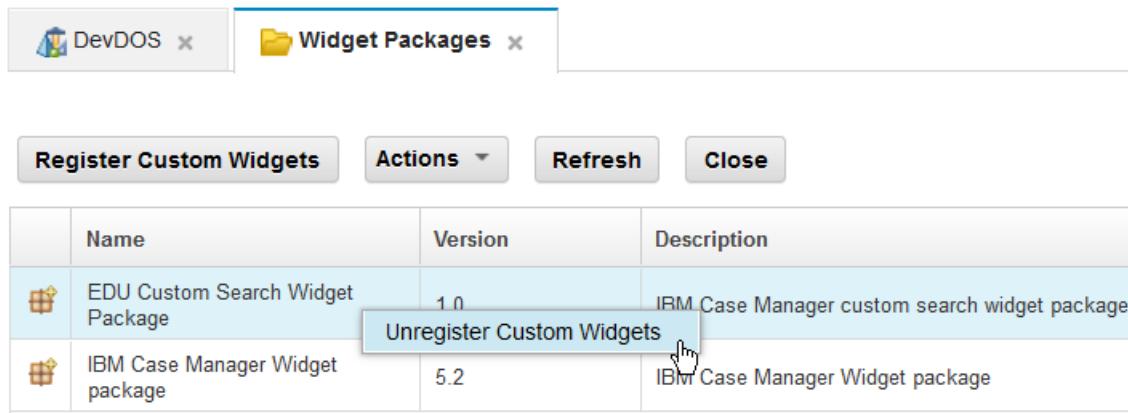
If the previous step did not resolve your issue, do the following steps.

1. Check your code and if needed, replace your code with the solution files.
2. Package the code again into a ZIP file.
 - a. Refer to Procedure 1: Package the widget files into a ZIP file, page 44 for more details.
3. Unregister the existing widget package.
 - a. Refer to Procedure 1: Unregister the custom widget, page 55 for more details.
4. Repeat the registration with the new package.
 - a. Refer to Procedure 2: Register the custom widget, page 46 for more details.
5. Verify the Content Navigator plug-in.
 - a. Refer to Procedure 3: Verify the Content Navigator plug-in registration, page 48 for more details.
6. Clear browser cache.
 - a. The student image is configured to clear the cache when you close the browser.
7. Test your widget.
 - a. Refer to Procedure 5: Test the custom search widget, page 53 for more details.

Procedure 1: Unregister the custom widget

In this procedure, you unregister your custom widget package in the IBM Case Manager admin tool.

1. Start the IBM Case Manager administration client.
 - URL: <http://ecmedu01:9080/navigator/?desktop=icmadmin>
 - User name: P8admin
 - Password: IBMFileNetP8
2. Open the Design Object Store.
 - a. Click P8Domain > Object Stores > DevDOS in the left pane.
The DevDOS tab opens.
3. In the DevDOS tab, expand the DevDOS and select “Widget Packages” in the left pane.
 - a. In the Widget Packages tab, right-click your widget package and select “Unregister Custom Widgets”.
The “Unregister Custom Widgets” tab opens.



4. Click Finish.
 - a. When you get the message that the package is removed, click Close.
 - b. Verify that your widget package is removed from the list in the Widgets Packages tab.
5. Log out of the IBM Case Manager admin tool and close the browser.

LESSON 3.5: Create a widget with a toolbar and a menu

What this lesson is about

This lesson describes how to create a widget with toolbar and menu actions. You also define widget properties, and add event handling for your widget. In the lessons 1-4, you created an IBM Content Navigator plug-in project, added widgets as part of the plug-in, and registered the widget in IBM Case Manager administration client.

In this lesson, you create a web module for your widgets along with the plug-in, and deploy the widget in IBM Case Manager configuration tool.

What you should be able to do

After completing this lesson, you should be able to:

- Create a widget with a toolbar and a menu.
- Define widget properties and add event handling to your widget.
- Create a web module for the widgets.

How you will check your progress?

- Hands on labs.

Lab Solution files

- The solution files for this lesson are included in the following folder:
C:\ICM\Widgets\CustomPageWidget\Solution Files

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>
-

Folder structure for the custom widget project in this lesson

Sub-projects and folders for the primary project

In this lesson, you use the following sub-projects and folders for the primary project that is required for packaging your custom widgets:

- **ICMCustomPlugin**
 - This project contains the files that are required to create a plug-in for IBM Content Navigator.
 - You can name this project as wanted.
 - You create a JAR file from this project.
 - **ICMCustomWidget**
 - This project is used for the web module.
 - It contains the package structure and the files for the custom widget.
 - You can name this project as wanted.
 - You create an EAR file from this project.
 - **ICMRegistry**
 - This folder provides the widget definition and catalog JSON files for your widget.
 - This folder name is required.
 - **Build**
 - This folder contains the build XML files to package your project into a ZIP file.
-

Implement toolbar and menu for your widget

Menu implementation

IBM Case Manager JavaScript API provides the following class to include menu items in your widget: `icm.widget.menu.ContextualMenu`

This class

- Extends `icm.widget.menu.Menu`.
- Represents a contextual menu.
- Defined in: <`icm/widget/menu/ContextMenu.js`>.

Toolbar implementation

IBM Case Manager JavaScript API provides the following class to include toolbar items in your widget: `icm.widget.menu.Toolbar`

This class

- Extends `icm.widget.menu.Menu`.
 - Represents a toolbar.
 - Defined in: <`icm/widget/menu/Toolbar.js`>.
-

Exercise 3.5.1: Create a Java project in Eclipse

Introduction

In this exercise, you create a Java project, copy the folders with files for the project, and review the files.

Procedures

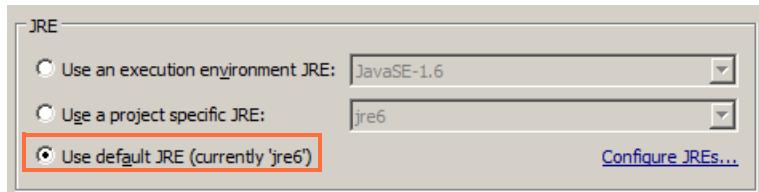
Procedure 2: Create a plug-in project in Eclipse, page 11

Procedure 2: Copy the folders for your project, page 61

Procedure 3: Review the IBM Content Navigator custom plug-in, page 61

Procedure 1: Create a Java project in Eclipse

1. Open Eclipse by double-clicking the Eclipse icon in your desktop.
 - a. In the Workspace Launcher page, leave the default workspace directory (C:\ICM\workspace_Eclipse) and click OK.
2. Open the project creation wizard.
 - a. In Eclipse, click File > New > Java Project.
The New Java Project page opens.
3. Create a Project.
 - a. In the New Java Project page, enter a Project Name (Example: CPageW - for “Custom Page Widget”).
 - b. In the JRE section, select the “Use default JRE” option. Make sure jre6 is selected.



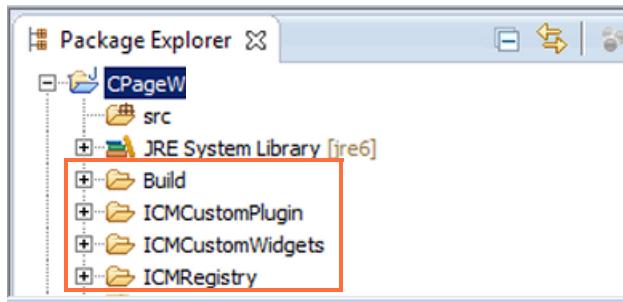
- c. Leave other default settings and click Next.
- d. Click Finish.
- e. Click Yes, if you are prompted to open the Java Perspective.

Procedure 2: Copy the folders for your project

The files that are required for this project are included in the student system. In this procedure, you copy these files into your project and use them as a starting point.

For this lab exercise, you use the following four folders:

1. In Windows Explorer, go to the C:\ICM\Widgets\CustomPageWidget folder.
 - a. Right-click the ICMCustomPlugin folder and select Copy.
 - b. In Eclipse > Package Explorer, right-click your project and select Paste.
2. Repeat Step 1 to copy the following folders into your project.
 - ICMCustomWidgets
 - ICMRegistry
 - Build
 - a. The directory must look like the following screen capture:



Procedure 3: Review the IBM Content Navigator custom plug-in

The IBM Content Navigator custom plug-in for this project is similar to the one that you did in Lesson 1. In this procedure, you check the code that is provided for adding actions.

1. In Eclipse > Package Explorer, expand your project > ICMCustomPlugin > src > com > ibm > icm > extension > custom.
 - a. Open the ICMCustomPlugin.java file.
2. The method public String getId returns "ICMCUSTOMPlugin" in line 18. This value provides an identifier for the plug-in.

```
16     public String getId() {  
17         // TODO Auto-generated method stub  
18         return "ICMCUSTOMPlugin";  
19     }
```

3. Observe the method public PluginAction[] getActions() near line 35. It provides a list of actions that this plug-in adds to the main toolbar of the web client.

```
public PluginAction[] getActions() {  
    return new PluginAction[] { new CustomAddCaseAction(), new  
    CustomAddToAttachmentAction(), new CustomAddTaskAction(), new  
    CustomAddCasePerRoleAction() };  
}
```

4. Close the file without any changes.



Note

The actions are defined under the ICMCustomPlugin > src > com > ibm > icm > extension > custom > actions folder. For more information, see the “F120: IBM Content Navigator 2.0.2: Customize and Extend the Features” course, and the ‘Customizing and Extending IBM Content Navigator’ RedBook.

5. In Eclipse > Package Explorer, expand your project > ICMCustomPlugin > src > com > ibm > icm > extension > custom > WebContent.

- a. Open the ICMCustomPlugin.js file that is the base JavaScript file.

It is loaded when IBM Content Navigator loads the plug-in. You can use this JavaScript file to apply any global changes (such as a style override) or load any JavaScript classes that must be available throughout the session.

- b. Observe that the context root for the web module of the custom widget package is specified.
 - c. The code that is shown in the screen capture registers the Dojo module path icm/custom for the custom runtime code.

```
3 var icmContextRoot = "/ICMCustomWidgets";  
  
45 //setup ICM runtime  
46 dojo.setObject("ecmwdgt.contextRoot", icmContextRoot);  
47  
48@var paths = {  
49     "icm/custom":"/ICMCustomWidgets/icm/custom",  
50     "icm":"/ICMClient/icm"  
51 };  
52 require({paths:paths});
```

- d. In the following lines of this file, CSS and other scripts are loaded.
6. Close the file without any changes.

Exercise 3.5.2: Create the widget definition JSON file

Introduction

You already copied a folder that contains the widget definition and the catalog JSON files into your project. In this procedure, you review the contents in the files.

Procedures

Procedure 1: Define widget properties (attributes), page 63

Procedure 2: Define the menu action for the custom widget, page 64

Procedure 3: Define the toolbar action for the custom widget, page 65

Procedure 4: Define the events for the custom widget, page 66

Procedure 1: Define widget properties (attributes)

The widget definition and the catalog JSON files that are used in this project are similar to the one that you created in Lesson 2.

1. In Package Explorer, expand your project > ICMRegistry folder and double-click the file Catalog.json to open it.
 - a. Review the values for the fields.
2. In the ICMRegistry folder, double-click the widget definition JSON file, CustomPageWidget.json to open it.
3. Observe the “properties” section > “propertyType”: “property” blocks.
 - You can enter one or more widget properties.
 - For this exercise, String, Integer, and Boolean properties are used as an example.
 - The properties are shown in the Settings tab of your custom widget configuration page.
 - In Case Manager Builder, the users can access the configuration page from the Page Designer, and assign values during the design time.

Custom Page Widget

Settings	menu	toolbar
String property:		
<input type="text" value="http://www.ibm.com"/>		
Integer Property:		
<input type="text" value="50"/>		
<input checked="" type="checkbox"/> Boolean property		

Procedure 2: Define the menu action for the custom widget

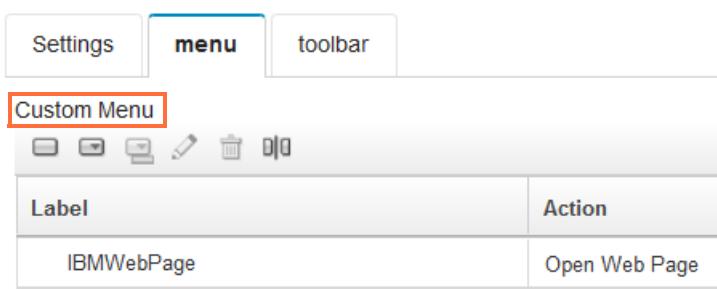
The menus allow the user to right-click the custom widget to get a list of options to invoke.

1. Observe the "properties" section > "propertyType": "group" > "id": "Menu" block.
 - a. Verify that the "type" field has "contextualMenu" as the value.
2. The "id" value ("customContextualMenu") and the "context" value ("CustomContext") that are specified in this file must be used in the widget JavaScript file.
3. The menu has an actionList that might be an array of values. For this lab, "Open Web page" is used as an example.

```
"propertyType": "group",
"type": "tab",
"id": "Menu",
"title": "menu",
"propertyMembers": [
    {
        "propertyType": "property",
        "type": "contextualMenu",
        "id": "customContextualMenu",
        "context": ["CustomContext"],
        "defaultValue": {
            "actionList": [
                {
                    "actionDefinitionId": "icm.action.utility.OpenWebPage",
                    "propertiesValue": {
                        "label": "IBMWebPage",
                        "targetURL": "http://www.ibm.com"
                    }
                }
            ]
        },
        "required": false,
        "visibility": true,
        "title": "Custom Menu"
    }
]
```

4. The Custom Menu that is defined in this section is displayed in the menu tab of the custom widget configuration page as shown in the following screen capture.
 - a. The value for the "title" ("Custom Menu"), the "label" ("IBMWebpage") and the action name are also displayed on the page.
 - b. In Case Manager Builder, the users can access the page from the Page Designer during the design time to assign values.

Custom Page Widget



Procedure 3: Define the toolbar action for the custom widget

A toolbar with a button allows the user to click it to start an action.

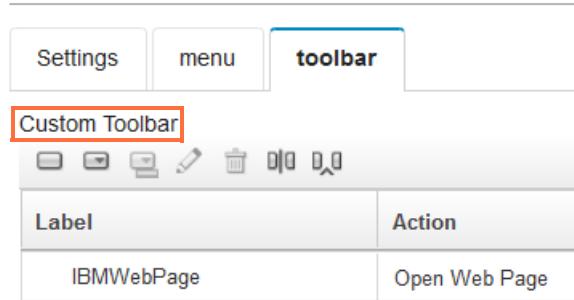
1. Observe the "properties" section > "propertyType": "group" > "id": "Toolbars" block.

```
"propertyType": "group",
"type": "tab",
"id": "Toolbars",
"title": "toolbar",
"propertyMembers": [
    {
        "propertyType": "property",
        "type": "toolbar",
        "id": "customToolbar",
        "context": ["CustomContext"],
        "defaultValue": {
            "actionList": [
                {
                    "actionDefinitionId": "icm.action.utility.OpenWebPage",
                    "propertiesValue": {
                        "label": "IBMWebPage",
                        "targetURL": "http://www.ibm.com"
                    }
                }
            ]
        },
        "required": false,
        "visibility": true,
        "title": "Custom Toolbar"
    }
]
```

2. Verify that the "type" field has "toolbar" as the value.

3. The "id" value ("customToolbar") and the "context" value ("CustomContext") that are specified in this file must be used in the widget JavaScript file.
4. The toolbar has an actionList that might be an array of values. For this lab, "Open Web page" is used as an example.
5. The Custom toolbar that is defined in this section is displayed in the toolbar tab of the custom widget configuration page as shown in the following screen capture.

Custom Page Widget



- a. The value for the "title" ("Custom Toolbar"), the "label" ("IBMWebpage") and the action name are also displayed on the page.
- b. In Case Manager Builder, the users can access the page from the Page Designer during the design time to assign values.

Procedure 4: Define the events for the custom widget

This section determines whether to handle or broadcast an event and sets the function name to handle the event if applicable.

1. Verify that the id is set to `icm.CustomEvent`.

This event id allows the custom widget to handle any event that is wired to the widget.

```
"events": [
  {
    "id": "icm.CustomEvent",
    "title": "Custom Event 1",
    "functionName": "handleICM_CustomEvent",
    "direction": "subscribed",
    "description": "Custom Event 1"
  }
]
```

2. The values for title and description of the event are not specific like icm.CustomEvent. You can provide any string for the title and description fields.
3. Set the `functionName` and `direction` fields for the event.
 - a. Because the widget is handling an event from another widget, verify that the direction is set to "subscribed".

**Note** _____

Recall that in lesson 2, you set the direction to broadcast, because the custom search widget sends an event to another widget.

- b. The `functionName` is `handleICM_CustomEvent`.
You use this function name in the widget JavaScript file. The names in both files must match.
 4. Close the widget definition file without any changes.
-

Exercise 3.5.3: Implement your widget

Introduction

In the previous exercise, you copied the folder that contains the widget implementation files to your project. In this exercise, you review the code and edit the files.

Procedures

Procedure 1: Set up Dojo AMD loading, page 68

Procedure 2: Expose widget properties (attributes) in the content pane, page 70

Procedure 3: Handle and expose an event, page 71

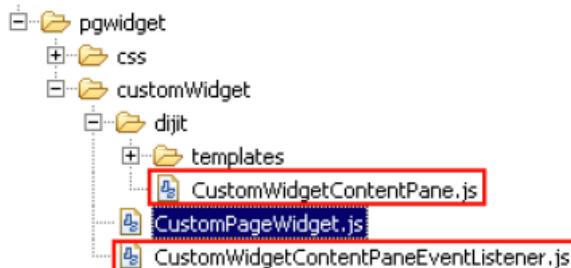
Procedure 4: Incorporate a toolbar button into the custom widget, page 72

Procedure 5: Incorporate a menu action into the custom widget, page 74

Procedure 1: Set up Dojo AMD loading

In this procedure, you add Dojo AMD loading into CustomPageWidget.js.

1. In Eclipse > Package Explorer > your project, expand ICMCustomWidgets > WebContent > icm > custom > pgwidget > customWidget
 2. Open the CustomPageWidget.js file.
 - Notice that the Dojo libraries are already loaded (Example: dojo/dom-style).
 - Some of the IBM Case Manager Java Script files are also referenced (Example: icm/base/BasePageWidget).
 3. Define your Java Script files in the custom widget package with Dojo AMD.
 - a. In your Eclipse project navigation pane on the left, note the path for CustomWidgetContentPaneEventListener . js and CustomWidgetContentPane . js.



- b. You define these files in the CustomPageWidget.js file so that you can call the functions in those classes.

- c. Complete the last two empty quotation marks in CustomPageWidget.js file in the define header as shown in the screen capture.

```
define(["dojo/_base/declare",
        "dojo/_base/lang",
        "dojo/dom-geometry",
        "dojo/dom-style",
        "icm/base/BasePageWidget",
        "icm/custom/pgwidget/customWidget/CustomWidgetContentPaneEventListener",
        "icm/custom/pgwidget/customWidget/dijit/CustomWidgetContentPane"],
```

- d. Add the two JavaScript file definitions into the function method after the define header as shown in the screen capture.

```
"icm/base/BaseActionContext"], function(declare, lang, domGeom, domStyle, I
eventListener, contentPaneWidget, MenuManager, toolBar, ContextualMenu,
```

4. Define the following IBM Case Manager JavaScript API classes to make the menu and toolbar features available in the custom widget.

"icm/widget/menu/ContextMenu"

"icm/widget/menu/Toolbar"

- a. Add the paths for these two classes into the define section and into the function section.
b. The completed code looks like the one in the screen capture:

```
"icm/widget/menu/MenuManager",
"icm/widget/menu/Toolbar",
"icm/widget/menu/ContextMenu",
"icm/base/BaseActionContext"], function(declare, lang, domGeom, domStyle, I
eventListener, contentPaneWidget, MenuManager, toolBar, ContextualMenu,
```

5. Save the changes and close the file.



Hint

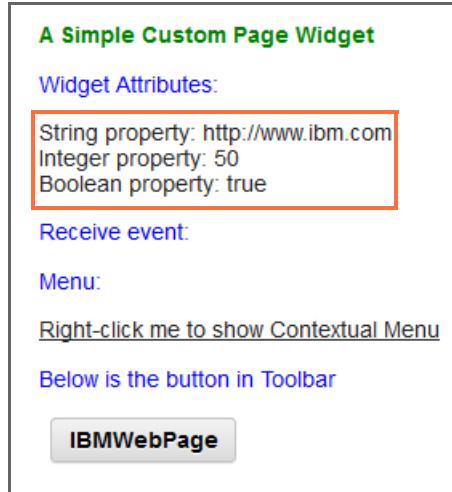
You can refer to the completed file at C:\ICM\Widgets\CustomPageWidget\Solution
Files\CustomPageWidget-Solution.js.

Procedure 2: Expose widget properties (attributes) in the content pane

In the previous lab exercise, you defined custom properties for your widget in the widget definition file.

- The properties are displayed in the settings tab of your custom widget configuration page.
- In Case Manager Client, the users can access the configuration page from the Page Designer, and assign values during the design time.
- To see the values for these properties in the custom widget at the runtime, you must expose the properties on the widget's content pane.

In this procedure, you edit the HTML file for the custom widget content pane, and then set up in the CustomWidgetContentPaneEventListerner.js file to show the properties on the content pane.



1. In Eclipse > Package Explorer > your project, expand ICMCustomWidgets > WebContent > icm > custom > pgwidget > customWidget > dijit > templates
2. Define a dojoAttachPoint to display the widget attributes on the content pane page.
 - a. Open the CustomWidgetContentPane.html file.
 - b. On line 6, add a dojoAttachPoint with a label "displayAttributes" as shown in the following code:

```
<div dojoAttachPoint="displayAttributes" style="width:100%;"></div>
```
 - c. Save the changes and close the file.
3. Open the CustomWidgetContentPaneEventListerner.js file in the ICMCustomWidgets > WebContent > icm > custom > pgwidget > customWidget folder.
You use the dojoAttachPoint (displayAttributes) that you added to display the string, integer, and boolean widget attribute values.
 - a. Observe that in line 25, a variable with a name: props is defined and contains code to display each property as shown in the following screen capture.

```
var props = 'String property: ' +
this.contentPane.widgetProperties['customProperty1']
+'<br> Integer property: ' +
this.contentPane.widgetProperties['customProperty2']
+'<br> Boolean property: ' +
this.contentPane.widgetProperties['customProperty3'];
```

4. In the next line (26), assign the `props` variable to the `dojoAttachPoint` (`displayAttributes`) as shown in the following line:

```
this.contentPane.displayAttributes.innerHTML = props;
```

5. Save the changes and close the file.

**Hint**

You can refer to the completed files (`CustomWidgetContentPane-Solution.html` and `CustomWidgetContentPaneListner-Solution.js`) in the “`C:\ICM\Widgets\CustomPageWidget\Solution Files`” folder.

Procedure 3: Handle and expose an event

In this procedure, you handle an event that is wired to the custom widget, and expose the event name to the content pane of the custom widget. The event name is shown in your custom widget in Case Manager Client.

For this project, you wire the Search widget to the custom widget and the `icm.SearchCases` event is displayed as shown in the following screen capture.

A Simple Custom Page Widget

Widget Attributes:

String property: <http://www.ibm.com>
Integer property: 50
Boolean property: true

Receive event:

`icm.SearchCases`

Menu:

[Right-click me to show Contextual Menu](#)

Below is the button in Toolbar

[IBMWebPage](#)

1. If it is not already expanded, in Eclipse > Package Explorer > your project, expand ICMCustomWidgets > WebContent > icm > custom > pgwidget > customWidget > dijit > templates
2. Define a dojoAttachPoint to display the event name in the content pane page.
 - a. Open the CustomWidgetContentPane.html file.
 - b. On line 8, add a dojoAttachPoint with a label "displayEvent" as shown in the following code:

```
<div dojoAttachPoint="displayEvent" style="width: 100%; "></div>
```
 - c. Save the changes and close the file.
3. Open the CustomPageWidget.js file in the ICMCustomWidgets > WebContent > icm > custom > pgwidget > customWidget folder.
 - a. Starting in line 74, verify that the name of the handler function that was defined earlier in the widget definition JSON file. The name in both files must match.
 - b. Call the displayPayload function of the CustomWidgetContentPaneEventListener.js file. You defined this Listener with Dojo AMD loading. Add the following code in line 79.

```
this.contentPaneEventListener.displayPayload(payload);
```
 - c. Save the changes and close the file.
4. Open the CustomWidgetContentPaneEventListener.js file in the ICMCustomWidgets > WebContent > icm > custom > pgwidget > customWidget folder.
 - a. Locate the displayPayload function block on line 19.
 - b. Assign the eventName field of the payload to the dojoAttachPoint (displayEvent) that you defined earlier. Add the following code in line 20.

```
this.contentPane.displayEvent.innerHTML = payload.eventName;
```
 - c. Save the changes and close the file.

**Hint**

You can refer to the completed files (CustomWidgetContentPane-Solution.html, CustomWidgetContentPaneListner-Solution.js, and CustomPageWidget-Solution.js) in the "C:\ICM\Widgets\CustomPageWidget\Solution Files" folder.

Procedure 4: Incorporate a toolbar button into the custom widget

In this procedure, you incorporate a toolbar action into the custom widget that can be displayed as a button within a toolbar area on the custom widget.

- You defined "open a web page" for the toolbar action in your widget definition file.
- In Case Manager Client, you can access the configuration page from the Page Designer, and add more toolbar action items during the design time.

For this project, the custom widget shows the toolbar button in Case Manager Client as shown in the following screen capture:

A Simple Custom Page Widget

Widget Attributes:

String property: http://www.ibm.com
Integer property: 50
Boolean property: true

Receive event:

icm.SearchCases

Menu:

Right-click me to show Contextual Menu

Below is the button in Toolbar

IBMWebPage

1. Define a dojoAttachPoint for the toolbar action for the content pane page.
 - a. Open the `CustomWidgetContentPane.html` file.
 - b. In line 13, add a `dojoAttachPoint` with a label "wrapTopToolbar".
 - c. Set the `data-dojo-type` to "dijit.layout.ContentPane".
 - d. Set the `data-dojo-props` to be "region: 'bottom', style: 'border:none; '".
 - e. The completed code looks like the following lines:

```
<div data-dojo-type="dijit.layout.ContentPane"
      data-dojo-attach-point="wrapTopToolbar"
      data-dojo-props="region: 'bottom', style: 'border:none; '>
</div>
```
 - f. Save the changes and close the file.
2. Open the `CustomPageWidget.js` file.
 - a. Observe the code in line 36. Verify that the toolbar with the `dojoAttachPoint` (that you defined in the previous step) is instantiated.

```
dojoAttachPoint: "customToolbar"
```



Note

Recollect that "customToolbar" is the value for the "id" field in the widget definition JSON file that you added in a previous exercise.

- b. Set the toolbar as a content of the page widget to get the action configuration from the page widget in line 39.

```
this.wrapTopToolbar.set("content", this.topToolbar.domNode);
```
- c. Verify that the toolbar is activated in line 42 (`this.topToolbar.startup()`).

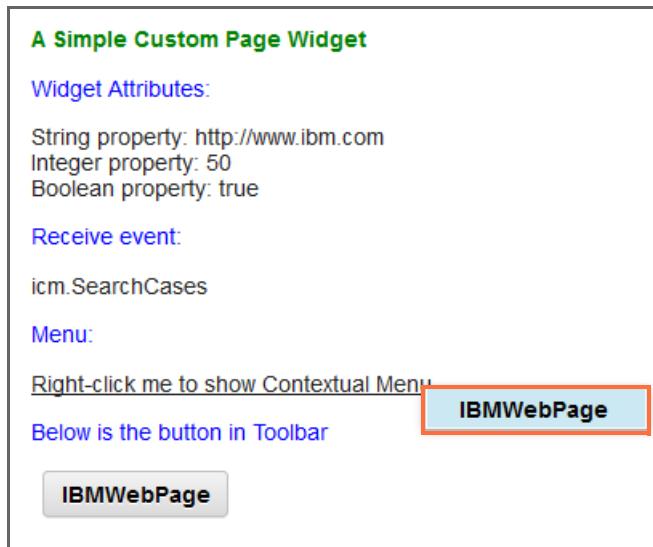
3. Save the changes and close the file.

Procedure 5: Incorporate a menu action into the custom widget

In this procedure, you incorporate a menu action into the custom widget that can be started when the user right-clicks on an area on the custom widget.

- You defined “open a web page” for the menu action in your widget definition file.
- In Case Manager Client, you can access the configuration page from the Page Designer, and add more menu action items during the design time.

For this project, the custom widget shows the menu item in Case Manager Client as shown in the following screen capture:



1. Define a dojoAttachPoint to display for the menu action for the content pane page.
 - a. Open the CustomWidgetContentPane.html file.
 - b. In line 10, check a dojoAttachPoint with a label “contextualMenuStore” as shown in the following code:

```
<div data-dojo-attach-point="contextualMenuStore"></div>
```
 - c. Close the file.

2. Open the `CustomPageWidget.js` file in the `customWidget` folder.
 - a. Observe the code in line 48. Verify that the `contextualMenu` with the `dojoAttachPoint` (that you defined in the previous step) is instantiated.
`dojoAttachPoint: "customContextualMenu"`

**Note**

Recollect that “`customContextualMenu`” is the value for the “`id`” field in the widget definition JSON file that you added in a previous exercise.

- b. Append the menu in the custom widget to start the menu action configuration from the page widget in line 52.
`this.contextualMenuStore.appendChild(this.menu.domNode);`
- c. Notice that the target reference of the `contextualMenu` is set in line 55 so that it can bind to the target point.
- d. Verify that the menu is activated in line 61 (`this.menu.startup();`).
3. Save the changes and close the file.
4. Optionally, open the files in the `ICMCustomWidgets > WebContent > icm > custom > action` folder and check the contents. These files implement custom actions.

LESSON 3.6: Build and deploy a widget as an EAR file

What this lesson is about

This lesson describes how to create a widget package that contains an EAR file and register the package.

What you should be able to do

After completing this lesson, you should be able to:

- Create a widget package that contains an EAR file
- Deploy and register a widget package.

How you will check your progress?

- Hands on labs.

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

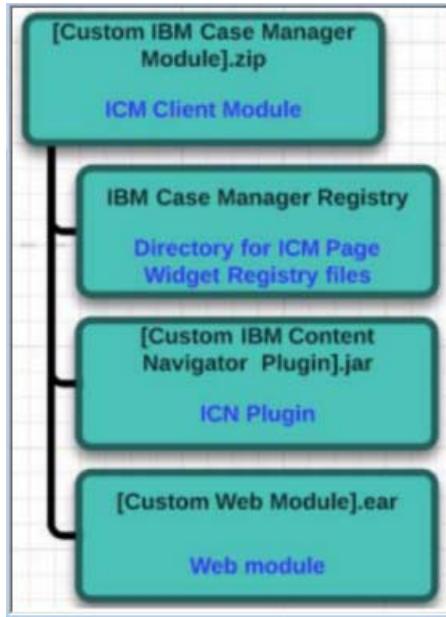
IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Widget package structure

Contents in a custom widget package

The following diagram shows the files that are included in the ZIP file for the widget package.



Build XML files that are used for this lesson to create the package

This project requires three build xml files to complete the following tasks:

- Create a Jar file for the IBM Content Navigator plug-in (ICMCustomPlugin))
- Create an EAR file for the widget (ICMCustomWidgets) package.
- Creates a final ZIP file that is needed for deployment. The ZIP file contains the following items:
 - JAR
 - EAR
 - Registry folder (ICMRegistry) that contains widget definition and catalog files.

Exercise 3.6.1: Build and deploy the widget package

Procedures

Procedure 1: Create your custom widget package, page 78

Procedure 2: Deploy and register the widget, page 79

Procedure 3: Verify the custom widget package deployment and registration, page 81

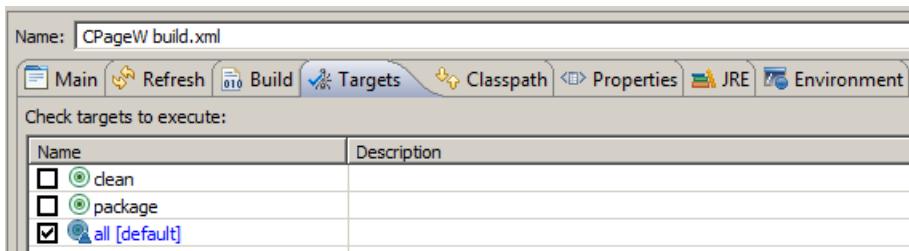
Procedure 4: Create a custom page to add the widget, page 84

Procedure 5: Test the custom widget, page 87

Procedure 1: Create your custom widget package

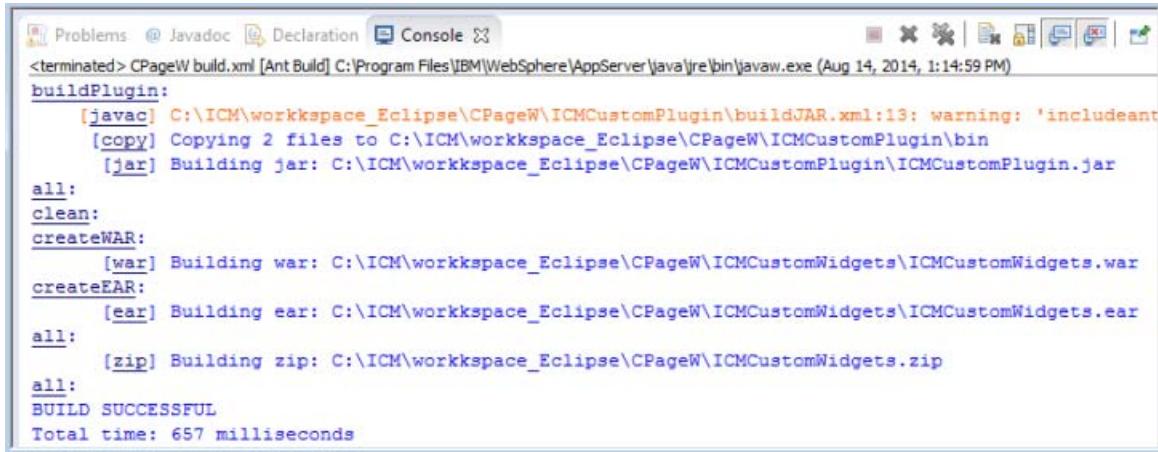
The build xml files that packages your project into a ZIP file is included in the student system. In this procedure, you review the files and generate the package.

1. In Package Explorer, expand your project.
 - a. Open the `build.xml` file in the Build folder of your project.
 - b. Notice that this file calls the other two build files to create the JAR and EAR files in the `<target name="package">` block.
 - c. It creates a final ZIP file that contains the JAR and EAR files and the ICMRegistry folder.
2. Optionally, open the `buildJAR.xml` file in the ICMCustomPlugin folder of your project.
 - a. The contents are similar to the file in Lesson 4.
3. Open the `buildEAR.xml` file in the ICMCustomWidgets folder of your project.
 - a. Notice that this file creates a WAR file first in the `<target name="CreateWAR">` block.
 - b. Then, it creates the EAR file in the `<target name="CreateEAR">` block.
 - c. Close all the files.
4. Create a package:
 - a. Right-click the `build.xml` in the Build folder.
 - b. select Run As > “2.Ant Build” from the list.
 - c. Make sure that “all [default]”, is selected, then click Run.



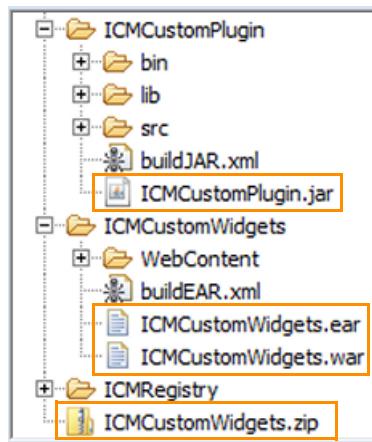
5. Validate that the console tab at the bottom pane displays that the Build was successful.

6. Verify that the message indicates that the JAR, WAR, and ZIP files are created.
7. Ignore the “includeantruntime” warning.



```
<terminated> CPageW build.xml [Ant Build] C:\Program Files\IBM\WebSphere\AppServer\java\jre\bin\javaw.exe (Aug 14, 2014, 1:14:59 PM)
buildPlugin:
[javac] C:\ICM\workspace_Eclipse\CPageW\ICMCustomPlugin\buildJAR.xml:13: warning: 'includeant
[copy] Copying 2 files to C:\ICM\workspace_Eclipse\CPageW\ICMCustomPlugin\bin
[jar] Building jar: C:\ICM\workspace_Eclipse\CPageW\ICMCustomPlugin\ICMCustomPlugin.jar
all:
clean:
createWAR:
[war] Building war: C:\ICM\workspace_Eclipse\CPageW\ICMCustomWidgets\ICMCustomWidgets.war
createEAR:
[ear] Building ear: C:\ICM\workspace_Eclipse\CPageW\ICMCustomWidgets\ICMCustomWidgets.ear
all:
[zip] Building zip: C:\ICM\workspace_Eclipse\CPageW\ICMCustomWidgets.zip
all:
BUILD SUCCESSFUL
Total time: 657 milliseconds
```

8. Right-click your project and select Refresh from the list.
 - a. Verify that the JAR, EAR, WAR, and ZIP files are listed in your project.

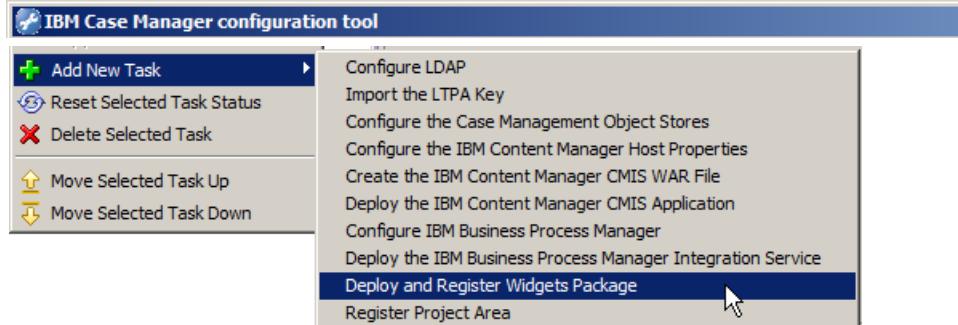


Procedure 2: Deploy and register the widget

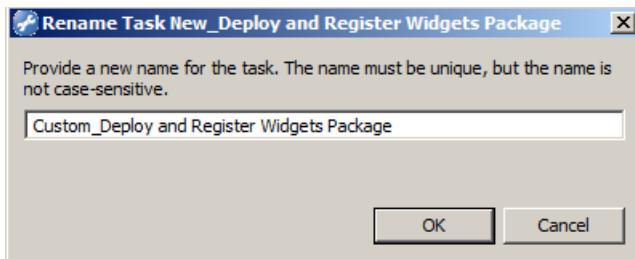
In this procedure, you deploy and register the widget in IBM Case Manager configuration tool.

1. Double-click the Case Manager Configuration Tool icon on the desktop.
 - a. You can also start it from Windows Start > All Programs > IBM Case Manager > Case Manager Configuration Tool.
2. Select File > Open Profile to open the configuration profile.
3. In the open window, go to the C:\Program Files (x86)\IBM\CaseManagement\configure\profiles\ICM Lab folder.

4. Select ICM_Lab.cfgp and click Open.
5. Add the “Deploy and Register Widgets Package” task.
 - a. Expand the ICM Lab node.
 - b. Right-click on any task, and select Add New Task > Deploy and Register Widgets Package. This makes a copy of the task.



6. Right-click the “New_Deploy and Register Widgets Package” and select “Rename Task”.
 - a. Edit the name to “Custom_Deploy and Register Widgets Package” and click OK.

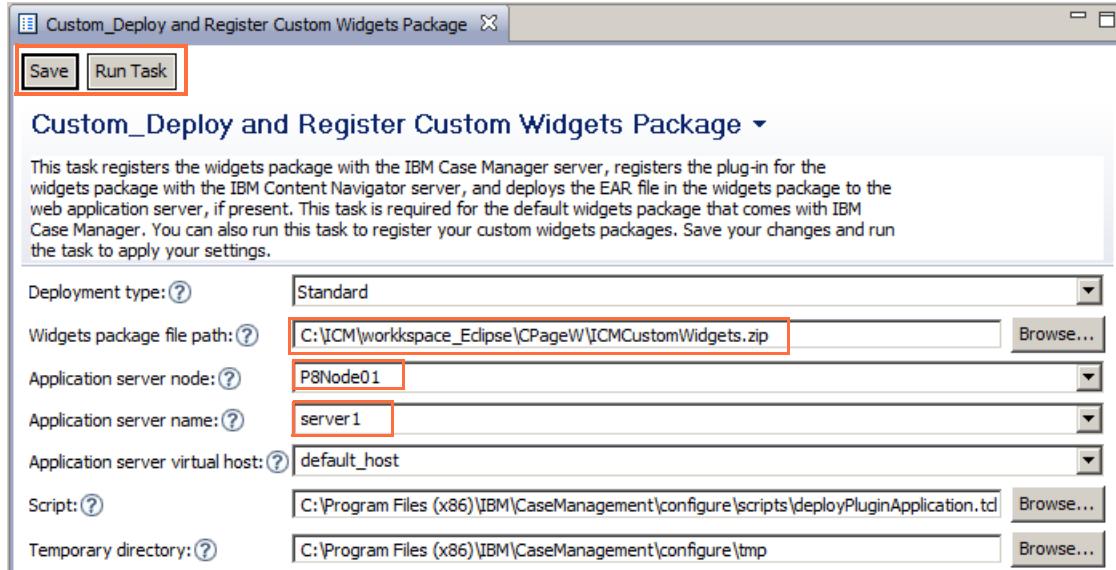
**Hint**

You can name this task to indicate your widget and save it. So that you can reuse this task every time you redeploy your widget package. To reuse the task, you must save the zip file in the same directory.

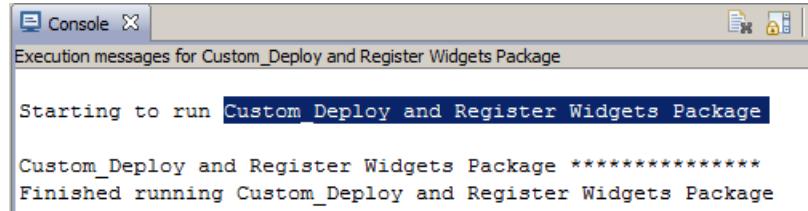
7. Edit the “Custom_Deploy and Register Widgets Package” task.
 - a. Right-click your task and select “Edit Selected Task” from the list.
 - b. Select the values for the fields with the data in the table.

Item	Value
Widgets package file path	C:\ICM\workspace_Eclipse\CPageW\ICMCustomerWidget.ZIP
Application server node	P8Node01
Application server name	server1

- c. Leave the default values for the other fields.
- d. For “Widgets package file path”, click Browse and select your widget package.



8. Click Save to save your changes. Run the task. It takes a few moments.
- a. Validate that the task ran successfully in the console below the task.

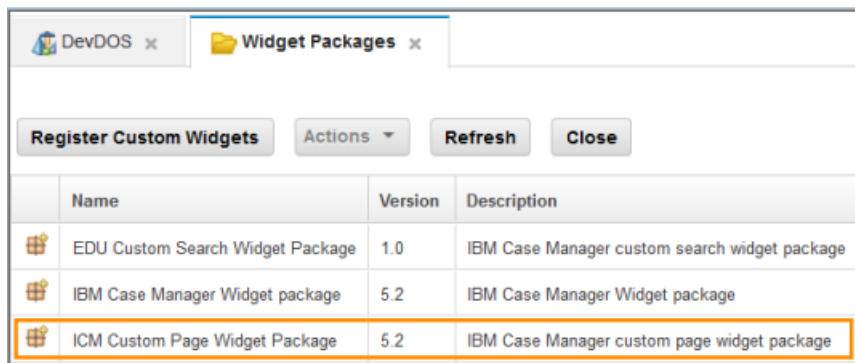


9. Click File > Exit to close the IBM Case Manager configuration tool.

Procedure 3: Verify the custom widget package deployment and registration

1. Verify that the custom widget package is registered:
 - a. Start the IBM Case Manager administration client.
 - URL: <http://ecmedu01:9080/navigator/?desktop=icmadmin>
 - User name: P8Admin
 - Password: IBMFileNetP8
 - b. Click P8Domain > Object Stores > DevDOS in the left pane to open the Design Object Store.
 - c. In the DevDOS tab > left pane, expand the DevDOS and click Widget Packages.

- d. Verify that your widget package is listed in the “Widget Packages” tab.



Name	Version	Description
EDU Custom Search Widget Package	1.0	IBM Case Manager custom search widget package
IBM Case Manager Widget package	5.2	IBM Case Manager Widget package
ICM Custom Page Widget Package	5.2	IBM Case Manager custom page widget package

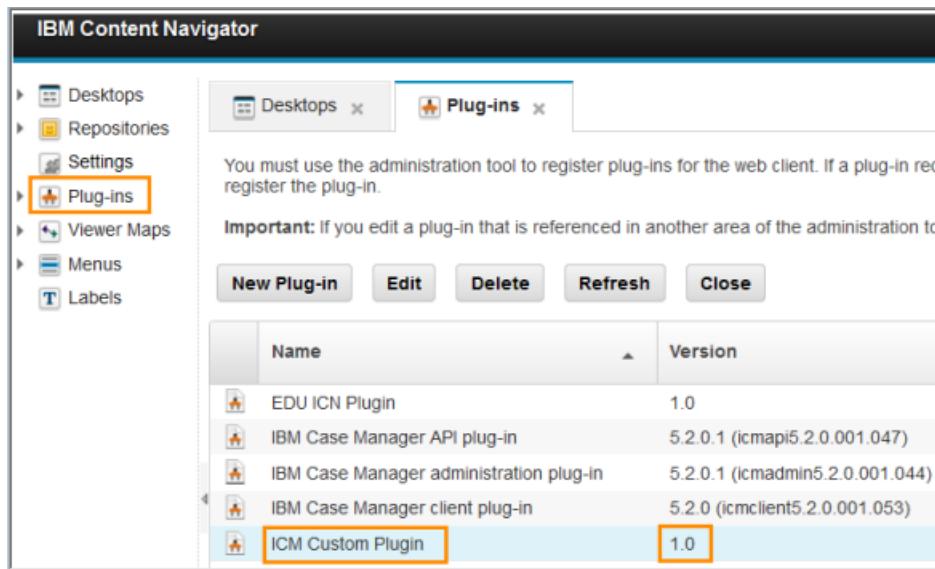


Note

The catalog and widget definition JSON files are deployed in the C:\Program Files (x86)\IBM\CaseManagement\configure\properties\widgetsPackage\DevDOS\ICM Custom Page Widget Package directory.

2. Verify that the custom plug-in is registered:

- Start the IBM Content Navigator Admin desktop.
 - URL: <http://ecmedu01:9080/navigator/?desktop=admin>
 - User name: P8Admin
 - Password: IBMFileNetP8
- In the IBM Content Navigator Admin desktop, select Plug-ins in the left pane.
- In the Plug-ins tab, verify that your plug-in is listed.



Name	Version
EDU ICN Plugin	1.0
IBM Case Manager API plug-in	5.2.0.1 (icmapi5.2.0.001.047)
IBM Case Manager administration plug-in	5.2.0.1 (icmadmin5.2.0.001.044)
IBM Case Manager client plug-in	5.2.0 (icmclient5.2.0.001.053)
ICM Custom Plugin	1.0

**Note**

The Content Navigator plug-in for your widget is deployed in the C:\Program Files (x86)\IBM\CaseManagement\configure\properties\plugins\ICMCustomePlugin.jar directory.

3. Verify that the widget EAR file is deployed:
 - a. Start the WebSphere Integrated Solutions Console.
 - URL: http://ecmedu01:9043/ibm/console/logon.jsp
 - User name: P8Admin
 - Password: IBMFileNetP8
 - b. Expand Applications > Application Types and select WebSphere enterprise applications.



- c. Verify that the custom widget EAR file is deployed and ICMCustomWidgets is listed.

The screenshot shows the 'Enterprise Applications' page. At the top, there's a toolbar with buttons for Start, Stop, Install, Uninstall, Update, Rollout Update, Remove File, and Export. Below the toolbar is a section titled 'You can administer the following resources:' which lists two items: 'CaseBuilder' and 'ICMCustomeWidgets'. The 'ICMCustomeWidgets' entry is highlighted with an orange border.

**Note**

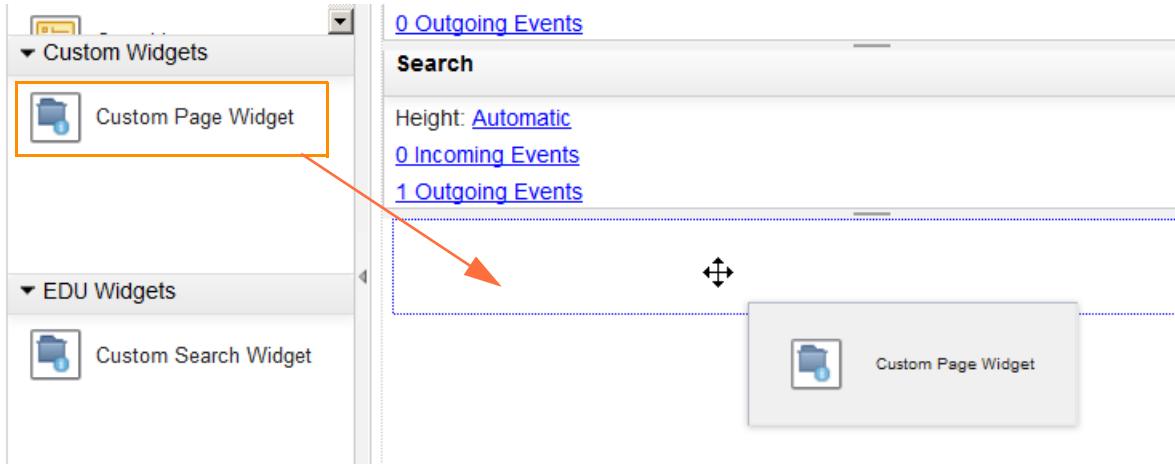
The EAR file for your widget is deployed in the C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\InstalledApps\P8Node01Cell\ICMCustomeWidgets.ear directory.

- d. Log out of all applications and close the browser.

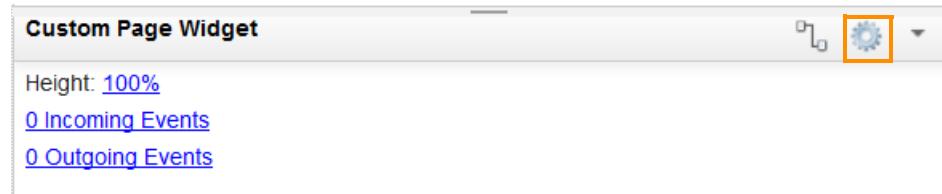
Procedure 4: Create a custom page to add the widget

The steps to test the custom widget are similar to the steps in Lesson 4.

1. In Firefox, log on to IBM Case Manager Builder as an administrative user.
 - URL: <http://ecmedu01:9080/CaseBuilder>
 - User name: P8Admin
 - Password: IBMFileNetP8
2. In Case Manager Builder, open Lab Claims Solution by clicking the Edit link.
Hover the mouse over the solution to see the links.
3. Create a custom page:
 - a. Open the Pages tab and expand the Solution Pages.
 - b. Hover the mouse over the Cases page name.
 - c. Select the Copy icon on the right side of the page.
 - d. In the resulting page, edit the name to Custom Page for your new page and click OK to create the copy.
 - e. Save your work by clicking Save at the top of the page.
4. Edit the page in Page Designer to add the custom widget.
 - a. In the Pages tab, double-click Custom Page.
 - b. Drag your “Custom Page Widget” (under the “Custom Widgets” section) from widget palette on the left column to the page on the right.
 - c. Place the widget below the existing default Search widget.



5. Optionally, configure the properties.
 - a. Click the “Edit Settings” icon of the Custom Page Widget.



- b. In the settings tab, enter values for the properties fields as shown in the screen capture.

Custom Page Widget

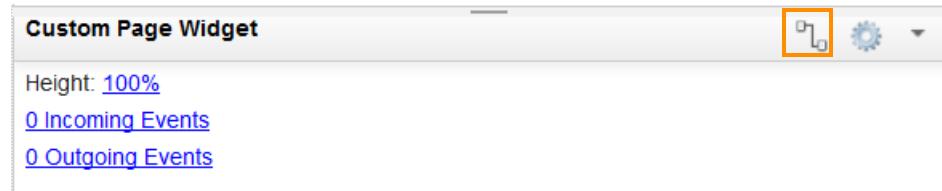
Settings menu toolbar

String property:
http://www.ibm.com

Integer Property:
50

Boolean property

- c. Click OK to close the page.
6. Edit the wiring for the widget:
- Click the “Edit Wiring” icon of the Custom Page Widget.



- b. In the Wire Events page > Event Wiring tab > “Incoming Events for the Custom Page Widget” section, complete the wiring with the data in the following table.

Field	Value
Source widget	Search
Outgoing event	Search cases
Incoming event	Custom Event 1

- c. Click Add Wire.

- d. The completed page looks like the one in the following screen capture.

Wire Events

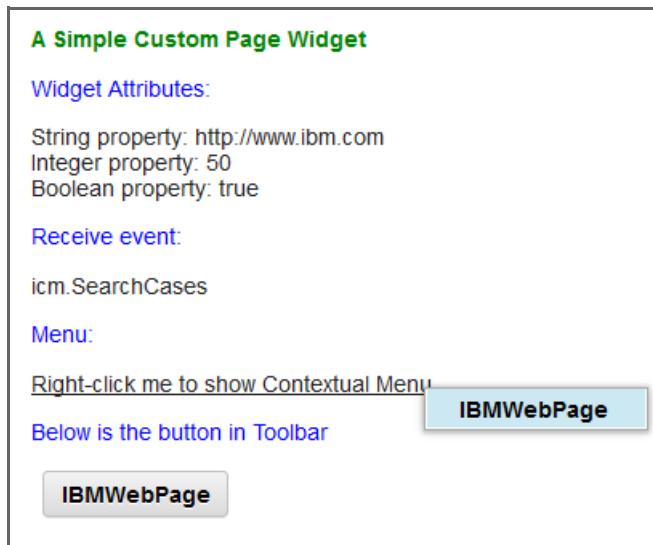
The screenshot shows the 'Wire Events' configuration for a 'Custom Page Widget'. At the top, a dropdown menu shows 'Widget' and 'Custom Page Widget'. Below it, two tabs are visible: 'Event Wiring' (which is selected) and 'Event Broadcasting'. A note below the tabs says: 'Add wires to establish communication between widgets. An asterisk (*) identifies an event related to an action.' The main area is titled 'Incoming Events for the Custom Page Widget'. It contains a table with four columns: 'Source', 'Event', 'Target', and 'Event'. One row in the table is highlighted with an orange border. The highlighted row shows 'Search' under 'Source', 'Search cases' under 'Event', 'Custom Page Widget' under 'Target', and 'Custom Event 1' under 'Event'. There is also an 'Outgoing event:' column with 'Search cases' and an 'Incoming event:' column with 'Custom Event 1'. An 'Add Wire' button is located to the right of the table.

- e. Click OK to close the page.
- f. Click Save and then Close to save your work and close Page Designer.
7. Assign the custom page to a role.
- Open the Roles tab.
 - Click the Customer Service Rep role link.
 - Open the Pages subtab.
8. Remove the “Custom Search” page that you created in the previous lesson.
- Select the page, hover over, and click the Remove (trash can) icon.
9. Assign the new page that you created.
- Click Assign Page.
 - Select Custom Page.
 - Click OK to close the dialog window.
10. Verify that your page is listed in the Pages tab.
- Click OK and then “OK All” to accept the changes to the role.
 - Click “Save and Close” at the top of the page to exit the solution editor.
11. Redeploy the solution.
- In the Manage Solutions page, select Lab Claims Solution and hover the mouse over.
 - Click Deploy.
 - In the Confirmation page, make sure that “Commit my changes and make them available for deployment” is selected, and click Deploy.
 - Wait for the green check mark to appear next to the solution.

12. Select Lab Claims Solution and hover the mouse over again.
 - a. Click Test to open Case Manager Client.

Procedure 5: Test the custom widget

1. In the Case Manager Client, select the Custom page tab to open your custom page.
2. Verify that the custom widget is displayed below the Search on the left pane.
 - a. Under the “Widget Attributes” section, a list of properties that you configured in the widget definition file is shown.
3. Test the menu.
 - a. Right-click the menu and check that the menu is shown.
4. Verify that the toolbar button is shown.
 - a. Click the “IBMWebPage” button, and check that it opens the web page.
5. Verify that the event is displayed.
 - a. Click the Search button to create a “Search Cases” event.
 - b. Verify that the custom widget shows the `icm.SearchCases` event.



6. Logout of the applications and close the browser.

Exercise 3.6.2: Troubleshooting

Introduction



Troubleshooting

If you get errors when you test your custom widget in the Case Manager Client, do the following steps.

1. Check your code and if needed, replace your code with the solution files.
 2. Package the code again into a ZIP file.
 - a. Refer to Procedure 1: Create your custom widget package, page 78 for more details.
 3. Redeploy and register your new package.
 - a. Refer to Procedure 2: Deploy and register the widget, page 79 for more details.
 4. Optionally, verify the deployment and registration.
 - a. Refer to Procedure 2: Deploy and register the widget, page 79 for more details.
 5. Verify the Content Navigator plug-in.
 - a. Refer to Procedure 3: Verify the Content Navigator plug-in registration, page 48 for more details.
 6. Clear the browser cache.
 - a. The student image is configured to clear the cache when you close the browser.
 7. Test your widget.
 - a. Refer to Procedure 5: Test the custom widget, page 87 for more details.
-

LESSON 3.7: Update an existing widget package

What this lesson is about

This lesson describes how to create a case comments widget, and update an existing widget package with new widgets. You can use the similar steps to update an existing widget in the package.

What you should be able to do

After completing this lesson, you should be able to:

- Create a case comments widget.
- Update an existing package with a new widget.

How you will check your progress?

- Hands on labs.

Lab Solution files

- The solution files for this lesson are included in the following folder:
C:\ICM\Widgets\CommentPageWidget\Solution Files

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

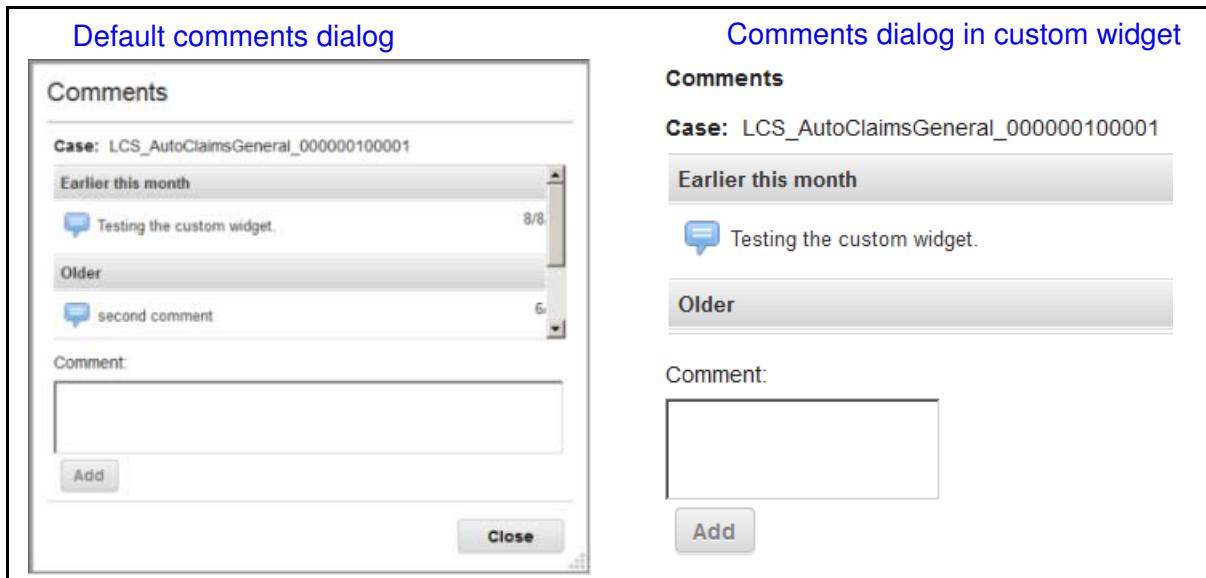
- <http://www.ibm.com/developerworks/library/>
-

Custom case comment widget

Default comments dialog dijit

The following screen capture shows the comments dialog interface both in the default IBM Case Manager Client and in the custom widget that you develop in this lesson.

- The default dialog opens, when you click the Comments button in the Case Details page.
- You add your custom widget in the Case Details page, and the comments dialog is available when you open the case in Case Details page.



- The custom case comment page widget (`icm/custom/pgwidget/commentWidget/CommentWidget.js`) that is used in this lesson embeds the IBM Case Manager-provided comment dialog dijit.

A screenshot of a code editor window titled 'CommentWidget.js'. The code is a RequireJS module definition:

```
1 define(["dojo/_base/declare",
2     "dojo/_base/lang",
3     "icm/base/Constants",
4     "icm/base/BasePageWidget",
5     "icm/base/_BaseWidget",
6     "icm/dialog/addcommentdialog/dijit/CommentContentPane",
7     "dojo/text!./templates/commentWidget.html",
8     "icm/base/Constants"]
```

The line 'icm/dialog/addcommentdialog/dijit/CommentContentPane' is highlighted with a yellow box.

- At run time, the case comment page widget coordinates with case toolbar page widget to save the unsaved case comment automatically when you save the case.

Update a custom widget package

Update an existing custom widget

If you update the code for your custom widget, use the following steps to redeploy the custom widget. Some of the steps are optional while others are mandatory.

1. In Case Manager Builder > Page Designer, remove the custom widget from all the pages where you added the widget. (optional)
2. After you update and package the widget, run the “Deploy and Register Widgets” task in IBM Case Manager Configuration Tool with the updated ZIP file for the custom widget. (mandatory)*
3. Clear the browser cache and cookies. (mandatory)*
4. In Case Manager Builder > Page Designer, add the new custom widget to the pages (optional: do this step only if you did the step 1)
 - a. Commit the changes and deploy the solution with your new widget.(optional)



Important

Steps 2-3 are sufficient for simple changes to the widget JavaScript files such as message outputs. If you change the widget definition JSON file that includes events, or wiring of the widget, then you must do all steps.

Update an existing custom widget package with new widgets

If you add a widget to an existing widget package, use the following steps to redeploy the custom widget package.

1. In Eclipse (or any other development tool), create and add a widget to your existing project.
 - a. Build the project to create a final ZIP file.
2. Run the “Deploy and Register Widgets” task in IBM Case Manager configuration tool with the updated ZIP file for the new custom widget. The tasks update the following items:
 - The web application in WebSphere Application Server
 - Registration for the custom widget package
 - The IBM Content Navigator plug-in
3. In Case Manager Builder > Page Designer, add the new custom widget to a page.
 - a. Commit the changes and deploy the solution with your new widget.

Exercise 3.7.1: Update an existing package with new widgets

Introduction

In this exercise, you create a comment widget, and add it to your existing package that you created in the previous lesson. You must redeploy the updated widget package to be able to use the new widget.

Procedures

Procedure 1: Create registry files for the custom widget, page 92

Procedure 2: Implement the CommentWidget, page 94

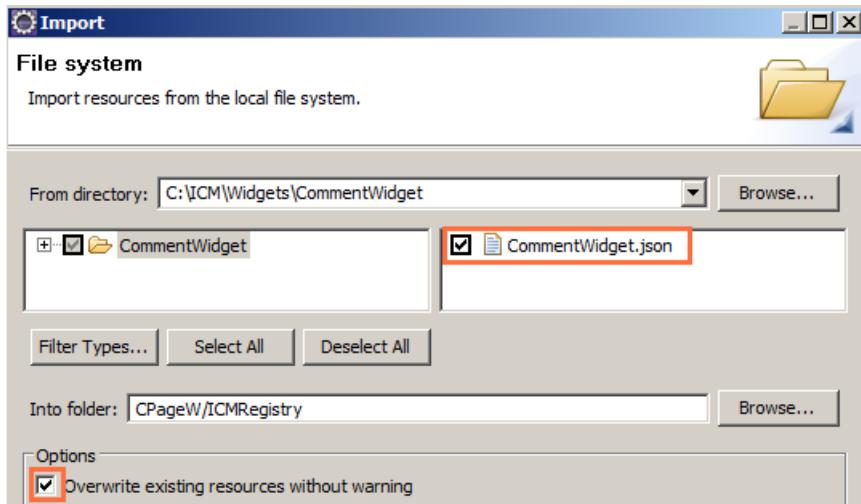
Procedure 3: Build and deploy the widget package, page 95

Procedure 4: Create a custom page to add the widget, page 95

Procedure 5: Test the custom widget, page 98

Procedure 1: Create registry files for the custom widget

1. If it is not already opened, start Eclipse by double-clicking the Eclipse icon in your desktop.
 - a. In the Workspace Launcher page, leave the default workspace directory (`C:\ICM\workspace_Eclipse`) and click OK.
2. Import the `CommentWidget.json` file into your project.
 - a. In Eclipse > Package Explorer > CPageW, right-click the `ICMRegistry` folder and click Import from the list.
 - b. In the Import page, expand General, select “File System”, and click Next.
 - c. In the Import > File system page, click Browse.
 - d. In the “Import from directory” page, expand `C:\ICM\Widgets` folder, select the `CommentPageWidget` folder, and click OK.
 - e. Back in the Import > File system page, select `CommentWidget.json` in the right pane.
 - f. Make sure that the “Into folder” field has the following value: `CPageW/ICMRegistry`
 - g. Select the “Overwrite existing resources without warning” option.
 - h. Click Finish.



3. Open the CommentWidget.json.
- a. The contents of the file is similar to the widget definition file of page widget that you created in the previous lesson.
- b. Check that it has “properties” and “events” section.

This widget handles two events: Select case and Send case information.

The two methods that are specified in this file (handleICM_SelectCaseEvent and handleICM_SendCaseInfoEvent) are defined in the JavaScript file.

```
"events": [
  {
    "id": "icm.SelectCase",
    "title": "Select case",
    "functionName": "handleICM_SelectCaseEvent",
    "direction": "subscribed",
    "description": "Display the case information for the case"
  },
  {
    "id": "icm.SendCaseInfo",
    "title": "Send case information",
    "functionName": "handleICM_SendCaseInfoEvent",
    "direction": "subscribed",
    "description": "Display the case that is specified in the"
  }
]
```

- c. Copy the lines 2-11 to paste it in the catalog file.
4. Edit the Catalog.json file to include the CommentWidget.
 - a. Open the Catalog.json file that you added for the custom page widget.
 - b. Check that in the “Widgets” section, you have a block for “CustomPageWidget”.
 - c. In line 24, after the closed curly braces (before the square bracket) add a comma.

- d. In the next line, add an open curly brace.
- e. Paste the lines that you copied from the `CommentWidget.json`.
- f. Remove the comma at the end of line 35.
- g. In line 36, add a closed curly brace (before the square bracket).
- h. The completed code must look like the following screen capture. The required code is highlighted.

```
23     "previewThumbnail": "images/customwidget_thumb.gif"
24 },
25 {
26     "id": "CommentWidget",
27     "title": "Comment Page Widget",
28     "category": "CustomWidgets",
29     "description": "This widget show comments added for a case",
30     "definition": "CommentWidget.json",
31     "preview": "images/customwidget_preview.png",
32     "icon": "images/customwidget_icon.png",
33     "runtimeClassName": "icm.custom.pgwidget.commentWidget.CommentWidget",
34     "help": "",
35     "previewThumbnail": "images/customwidget_thumb.gif"
36 }
37 ]
38 }
```

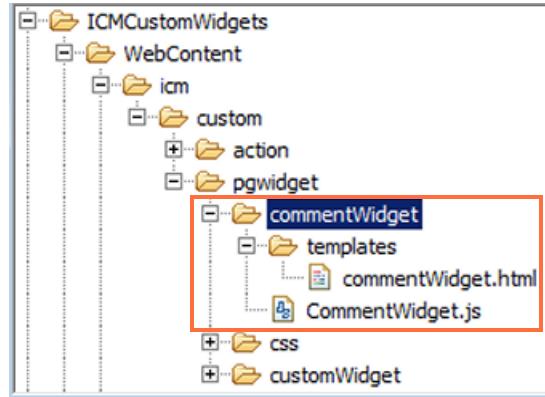
A red rectangular box highlights the entire JSON object for the `CommentWidget`. It includes the opening brace on line 24, the properties on lines 26 through 35, and the closing brace on line 36. The JSON object starts on line 23 and ends on line 38.

5. Save and close the files.

Procedure 2: Implement the CommentWidget

The files that are required for the `CommentWidget` are included in the student system. In this procedure, you copy these files into your project.

1. Copy the folders for your project.
 - a. In Windows Explorer, go to the `C:\ICM\Widgets\CommentPageWidget` folder.
 - b. Right-click the `commentWidget` folder and select `Copy`.
 - c. In Eclipse > Package Explorer > CPageW, expand `ICMCustomWidgets` > `WebContent` > `icm` > `custom` > `pgwidget`
 - d. Right-click `pgwidget` and select `Paste`.
 - e. Expand the `commentWidget` folder to see the contents.
 - f. The directory must look like the following screen capture:



Procedure 3: Build and deploy the widget package

In this procedure, you build and deploy the widget package to make your new widget available in the IBM Case Manager.

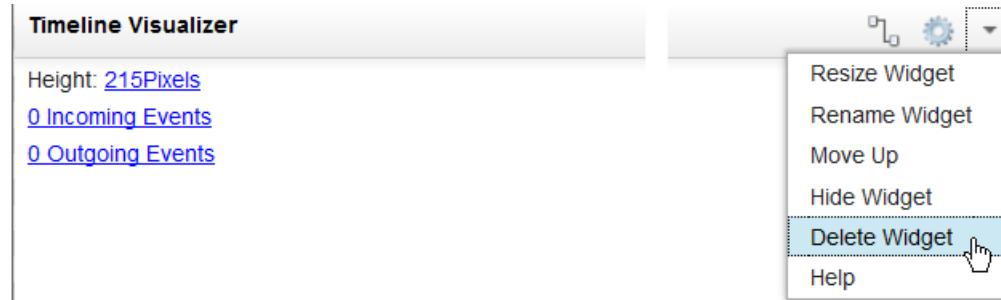
1. Build the widget package (as a Zip file)
 - a. See the Procedure 1: Create your custom widget package, page 78.
2. Deploy and register the widget package.
 - a. See the Procedure 2: Deploy and register the widget, page 79.
 - b. You can reuse the “Custom_Deploy and Register Custom Widgets” task.
3. Verify the deployment and registration.
 - a. See the Procedure 2: Register the custom widget, page 46.

Procedure 4: Create a custom page to add the widget

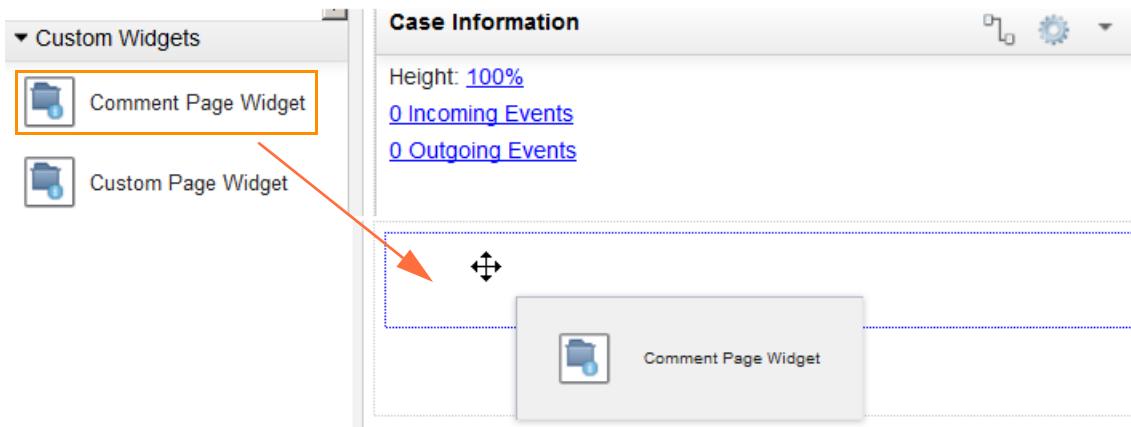
The steps to test the custom widget are similar to the steps in Procedure 1: Create a custom page, page 50.

1. In Firefox, log on to IBM Case Manager Builder as an administrative user.
 - URL: <http://ecmedu01:9080/CaseBuilder>
 - User name: P8Admin
 - Password: IBMFileNetP8
2. In Case Manager Builder, open Lab Claims Solution by clicking the Edit link. Hover the mouse over the solution to see the links.
3. Create a custom page:
 - a. Open the Pages tab and expand the Case Details Pages.
 - b. Hover the mouse over the Case Details page name.
 - c. Select the Copy icon on the right side of the page.

- d. In the resulting page, edit the name to **Comments Page** for your new page and click OK to create the copy.
 - e. Save your work by clicking **Save** at the top of the page.
4. Edit the page in Page Designer to add the custom widget.
 - a. In the Pages tab, double-click **Comments Page**.
 - b. In Page Designer > **Comments Page**, remove the Timeline Visualizer widget to make room for your widget.
 - c. Click the down-arrow in the toolbar of your widget, and select “Delete Widget”.
 - d. If the toolbar controls are not visible, expand the area for the widget.



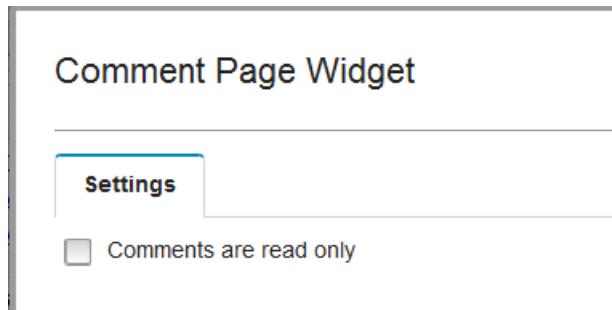
5. Add Comment Page Widget to the page.
 - a. Verify that the widget is added to the Custom Widgets category in the widget palette on the left pane .
 - b. Drag “Comment Page Widget” to the page.
 - c. Place the widget below the existing Case Information and Properties widgets.



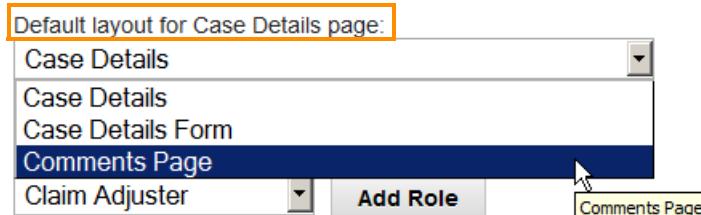
6. Check the configuration page for Comments Page Widget.
 - a. Click the “Edit Settings” icon.



- b. In the settings tab, notice the option to make comments “read only” as shown in the screen capture.



- c. Click OK to close the page.
7. Click Save to save your changes and Close to close Page Designer.
8. Assign the new page to a case type.
 - a. Select the Case Types tab.
 - b. Select Auto Claims General.
 - c. In the Case Type page, select Comments Page for “Default layout for Case Details page”.



9. Click Save and Close to save the changes to the solution.
10. Redeploy the solution.
 - a. In the Manage Solutions page, commit the changes to the Lab Claims Solution and Deploy it.
 - b. Wait for the green check mark to appear next to the solution.
11. Hover the mouse over the solution and click Test to open Case Manager Client.

Procedure 5: Test the custom widget

1. In the Case Manager Client, select the Cases tab.
 - a. Do a Search (criteria: Policy Family Name = Smith)
The value is case-sensitive; make sure “S” is uppercase.

Search:

Policy Family Name

Smith

Search [Advanced Search](#)



2. Click the Title link for the case in the right pane to open the Case details in the Comments Page.
3. In the Comments Page tab, verify that the Comments widget is shown at the end of the page.
 - a. Enter any text in the “Comment” input field to add a comment for the case.
 - b. Click Add. Verify that the comment is listed.

Comments

Case: LCS_AutoClaimsGeneral_000000100001

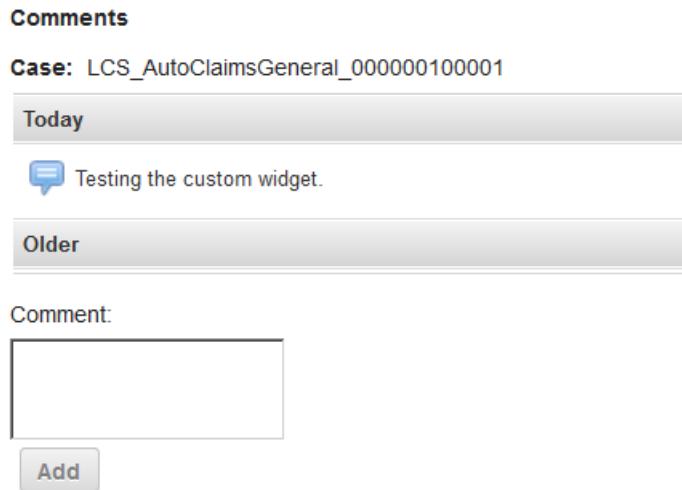
Today

Testing the custom widget.

Older

Comment:

Add



- c. Click the Comments button at the top of the page. A page similar to the custom Comments widget dialog opens.
The custom widget implements the IBM Case Manager-provided comment dialog dijit that the default Comments widget also implements.
 - d. Click Close to close the Case Details (Comments Page) page.
4. Logout of the applications and close the browser.

Appendix: Uninstall a custom widget in IBM Case Manager

Introduction

If you require to remove the custom widgets from the IBM Case Manager system, you must do the procedures in this section.

Procedures

Procedure 1: Remove the custom widget, page 99

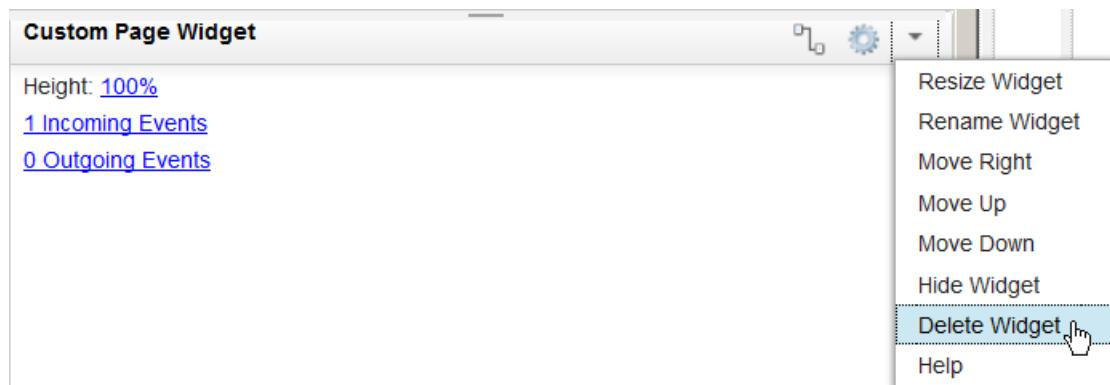
Procedure 2: Unregister the custom widget package, page 100

Procedure 3: Delete the IBM Content Navigator plug-in, page 101

Procedure 4: Uninstall the web application, page 101

Procedure 1: Remove the custom widget

1. In Firefox, log on to IBM Case Manager Builder as an administrative user.
 - URL: <http://ecmedu01:9080/CaseBuilder>
 - User name: P8Admin
 - Password: IBMFileNetP8
2. In Case Manager Builder, open Lab Claims Solution by clicking the Edit link.
Hover the mouse over the solution to see the links.
3. Open the custom page in Page Designer to edit it:
 - a. Select the Pages tab and expand the Solution Pages.
 - b. Double-click your custom page (Custom Page).
4. In the toolbar of your Custom Page Widget, click the down-arrow and select “Delete Widget”.
 - a. If the toolbar controls are not visible, expand the area for the widget.



5. Click Save to save your changes.
 - a. Click Close to close Page Designer.
 - b. Click Save and Close to save the changes to the solution.
6. Log out of IBM Case Manager Builder and close the browser.

**Note**

If a page contains multiple instances of the custom widget, you must delete all of them. If multiple pages include the custom widget, you must delete it from each page.

Procedure 2: Unregister the custom widget package

In this procedure, you unregister the widget package in the IBM Case Manager admin tool.

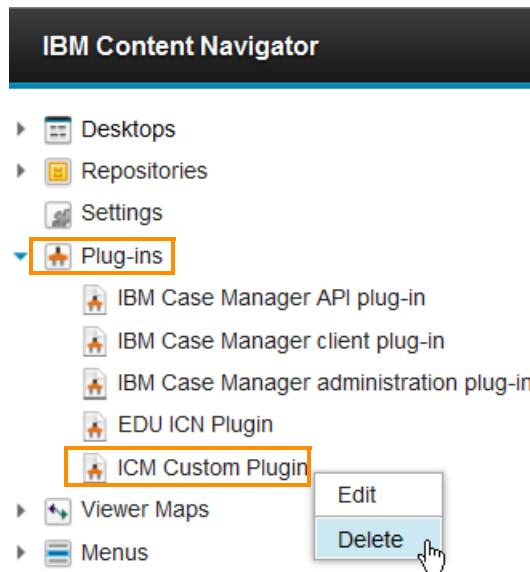
1. Start the IBM Case Manager administration client.
 - URL: <http://ecmedu01:9080/navigator/?desktop=icmadmin>
 - User name: P8admin
 - Password: IBMFileNetP8
2. Open the Design Object Store.
 - a. Click P8Domain > Object Stores > DevDOS in the left pane.
3. In the DevDOS tab, expand the DevDOS and select “Widget Packages” in the left pane.
 - a. In the Widget Packages tab, right-click your “ICM Custom Page Widget Package” and select “Unregister Custom Widgets”.

Name	Version	Description
EDU Custom Search Widget Package	1.0	IBM Case Manager custom search widget package
IBM Case Manager Widget package	5.2	IBM Case Manager Widget package
ICM Custom Page Widget Package		Unregister Custom Widgets

4. In the “Unregister Custom Widgets” tab, click Finish.
 - a. When you get the message that the package is removed, click Close.
 - b. Verify that your widget package is removed from the list in the Widgets Packages tab.
5. Log out of the IBM Case Manager admin tool and close the browser.

Procedure 3: Delete the IBM Content Navigator plug-in

1. Start the IBM Content Navigator Administration Desktop.
 - URL: <http://ecmedu01:9080/navigator/?desktop=admin>
 - User name: P8Admin
 - Password: IBMFileNetP8
2. Expand “Plug-ins” in the left pane.
3. Right-click your plug-in (ICM Custom Plugin) and click Delete.
4. Click Delete when you are prompted to confirm.



5. Verify that your plug-in is deleted from the list.
6. Log out of the IBM Content Navigator Administration Desktop and close the browser.

Procedure 4: Uninstall the web application

If the widget package contains a web module for your widgets (EAR file), you uninstall the web application for your custom widget package.

1. Start the WebSphere Integrated Solutions Console.
 - URL: <http://ecmedu01:9043/ibm/console/logon.jsp>
 - User name: P8Admin
 - Password: IBMFileNetP8
- a. In the left pane, expand Applications > Application Types and select WebSphere enterprise applications.
- b. Select ICMCustomWidgets from the list and click Uninstall.

The screenshot shows the 'Enterprise Applications' page in the WebSphere Integrated Solutions Console. At the top, there's a toolbar with buttons for Start, Stop, Install, Uninstall, Update, Rollout Update, Remove File, and Export. The 'Uninstall' button is highlighted with an orange box. Below the toolbar is a section for managing resources, showing icons for file operations like copy, move, and delete. A table lists applications under 'You can administer the following resources':

	Name	Application Status
<input type="checkbox"/>	CaseBuilder	
<input checked="" type="checkbox"/>	ICMCustomWidgets	

- c. In the “Uninstall Application” page, click OK.
 - d. In the “Enterprise Applications” page, click Save in the Messages section to save the changes to master configuration.
 - e. Verify that the application for your widget package is removed from the list.
2. Log out of WebSphere Integrated Solutions Console and close the browser.

 **Important** _____
You must restart the WebSphere Application Server, and clear the browser cache.

4

Implement External Data Services

This unit provides guidance for implementing an External Data Service (EDS). External data integration is a feature of IBM Case Manager that allows certain information about case properties, such as choice lists, to come from an external data source.

LESSON 4.1: Implement sample external data service

What this lesson is about?

This lesson shows some of the capabilities of the sample External Data Service (EDS) that IBM Case Manager provides. You can obtain certain information about case properties, such as choice lists, from a different repository, or an external data source.

What you should be able to do?

After completing this lesson, you should be able to use the sample External Data Services (EDS) for the following tasks:

- Change the field status dynamically for a property.
- Create choice lists and dependent choice lists.
- Implement case property validation.
- Prefill an initial value for a case property.
- Set the case property field status as required.

How you will check your progress?

- Hands on labs.

References

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Case data from an external data source

Introduction

- IBM Case Manager stores case data in Content Platform Engine.
- You can use an external data service with a solution to access data from a different repository or other data source.
 - The external data is incorporated into the case and stored with the rest of the case data in Content Platform Engine.

When to use an external data service?

- Get property values.
 - Example: Your customer records are stored in a database.
 - When a caseworker enters the serial number for a customer, the external data service can get the name and address of the customer from that database.
 - These values are then incorporated into the case data and stored in Content Platform Engine.
- Modify property attributes such as minimum value or maximum value.
 - The external data service must work within any constraints that are placed on the property attributes in Content Platform Engine.
 - For example, if a minimum value is specified for the property in Content Platform Engine, the external data service can set the minimum only within the allowed value for the Content Platform Engine.
- Define dependencies between properties to implement dynamic behavior in your solution.
 - Example: You can specify a dependency between a state property and a city property.
 - When a caseworker selects a state, the choice list that is associated with the city property contains only cities that are in that state.

How to get data from an external data source?

- Use the IBM Case Manager APIs to implement a service to extract case data from the external data source.
- Register the external data service in the IBM Case Manager administration client to use with your solution.
 - You can register only one external data service for a solution.

- Deploy or redeploy the solution.

**Note**

For this lesson, you use the IBM Case Manager provided sample EDS application to get data from an external source.

How is the data from the external data service handled?

- After you register the external data service, Case Manager Client communicates with the service to get case data whenever caseworker create cases or modify cases.
 - This communication is handled automatically through the IBM Case Manager APIs.
- Important: You use the external data service only for retrieving data from an external source.
 - If the caseworker modifies the data, Case Manager Client does not update the corresponding data in the external data source.
 - Case Manager Client saves the data that was received from the external data source only in Content Platform Engine.

Sample external data service

Sample External Data Service package

IBM Case Manager provides a sample External Data Service (EDS) that you can use it as a starting point for your application to implement EDS.



Note

The Sample External Data Service package is available from the developerWorks website. You can download it at this URL:

https://www.ibm.com/developerworks/community/blogs/e8206aad-10e2-4c49-b00c-fee572815374/entry/sample_external_data_service_for_ibm_case_manager_by_dave_hanson6?lang=en_us

Top-level files and directories

- The sample external data service gets data from an XML configuration file.
 - The sample is a web application that must be deployed to WebSphere Application Server.
- Sample contains the following top-level files and directories:
 - build.xml
 - An Apache Ant script file that builds the sample application.
 - env.properties
 - A file that defines specific file locations and other values that the build.xml script references. This file must be modified to match your specific build environment.
 - src/
 - This directory contains the Java source files.
 - config/
 - This directory contains the configuration files that are included in the built application.



Note

"Appendix: Steps to set up the sample External Data Service, page 25" at the end of this lesson contains the details about setting up the sample EDS application for IBM Case Manager.

Java code in the sample EDS service

Packages

The Java code of the sample service consists of two packages:

- `com.ibm.casemgmt.sampexterndata.rest`
 - Implements the RESTful interface to the sample external data service.
 - Processes the HTTP requests, for example, converting to and from JSON.
- `com.ibm.casemgmt.sampexterndata.api`
 - Implements a lower-level API layer that is responsible for processing the external data.
 - Parses the XML and process the configuration data for the various properties.

Main classes in the `com.ibm.casemgmt.sampexterndata.rest` package

- `SampleServiceServlet`
 - This class is the main servlet.
 - It derives from `javax.servlet.http.HttpServlet` and overrides various servlet methods.
 - Example: `init()`, `doGet()`, `doPost()`
- `SampleServiceHandler`
 - It encapsulates the actual processing of a particular request.
 - There is only one handler class in this sample, and it processes both the GET method to the `.../types` path and the POST method to the `.../type` path.
- `ResourceRequest`
 - This class represents the HTTP request.
 - It encapsulates the `HttpServletRequest` object that is passed to the servlet, and it provides additional information to the handler.
 - • Several classes are used to represent the
- `PropertiesJSONConverter`
 - This class is a utility class that has methods to convert a list of properties from the JSON passed as input to the external data service or to the JSON returned as the result.

Main classes in the `com.ibm.casemgmt.sampexterndata.api` package

- `FileBasedExternalData`
 - This class parses the XML file and generates a hierarchy of objects that corresponds to that XML file.

External data configuration in the sample

XML configuration file structure

The sample external data service is deployed with an XML file that defines what case types and properties the sample data service manages.

- You can edit this XML file to control the external data service.
- The sample XML file in this unit defines the following properties of a single case type:
 - PropOne
 - An integer property with static information such as minimum and maximum values.
 - State
 - A string property that controls the information of the City property.
 - State has a choice list with the possible states that can be selected.
 - City
 - A string property that changes depending on the value of the State property.
 - The City property has a different choice list for each value of the State property.
 - MVInt
 - A multi-value integer property that controls the information of the MVString property.
 - MVString
 - A multi-value string property that changes depending on the value of MVInt.

Modifying the external data configuration

The ExternalData.xml file is embedded in the WAR file that is deployed. This file can be modified to match the sample external data that you want to configure.

- To update the file after the application is deployed, you can rebuild the application, repackage the WAR, and then update the application from the WebSphere Application Server administrative console.
 - Alternatively, you can update the file at the following paths on the application server:
 - <was-root>/profiles/<profile-name>/config/cells/<cell-name>/applications/<application-name>.ear/deployments/<application-name>/sampleexternaldataservice.war/WEB-INF
 - <was-root>/profiles/<profile-name>/installedApps/<cell-name>/<application-name>.ear/sampleexternaldataservice.war/WEB-INF
 - After the files are updated, you must restart the application from the administrative console.
-

Exercise 4.1.1: Check the sample EDS configuration

Procedures

Procedure 1: Verify the sample EDS deployment, page 8

Procedure 2: Check the ExternalData.xml file in Eclipse project, page 9

Procedure 1: Verify the sample EDS deployment

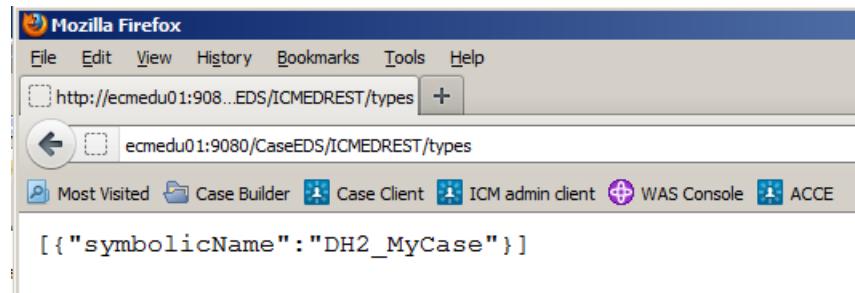
The External Data Services (EDS) web application must be deployed to use the service.



Note

Your student system is already configured with the sample external data service. For "how to configure sample", see Appendix: Steps to set up the sample External Data Service, page 25.

1. If it is not already started, start the WebSphere Application Server.
 - a. Click Start > All Programs > IBM WebSphere > IBM WebSphere Application Server V8.5 > Profiles > AppSrv01 > Start the server.
 - You can use Start Server1.bat in the WebSphere Admin folder on the desktop.
 - b. Wait for the Start the server page to close.
2. In a Firefox browser, go to <http://ecmedu01:9080/CaseEDS/ICMEDREST/types> (The URL is case-sensitive) or use the bookmark "CaseEDS" in the Bookmarks toolbar.
 - a. Verify that the browser shows a response with the case type in your solution. (MyCase).
Your student system contains an IBM Case Manager solution with a case type: "MyCase".

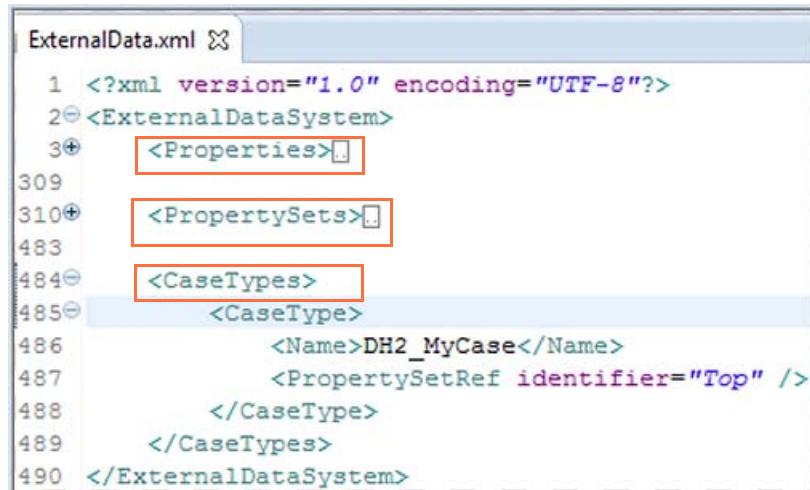


3. Optionally, check the case properties for the "EDS" solution > "MyCase" case type in IBM Case Manager Builder.
 - URL: <http://ecmedu01:9080/CaseBuilder>
 - User name: P8admin
 - Password: IBMFileNetP8

Procedure 2: Check the ExternalData.xml file in Eclipse project

An XML file in the sample external data service defines what case types and properties the sample data service manages.

1. Open Eclipse by double-clicking the Eclipse icon in your desktop.
 - a. In the Workspace Launcher page, leave the default workspace directory (C:\ICM\workspace_Eclipse) and click OK.
2. Open the SampleExternalDataService project.
 - a. In Package Explorer, expand the project > config folder.
 - b. Double-click the file `ExternalData.xml` to open it.
3. Observe that the file contains Properties, PropertySets, and CaseTypes top-level nodes.
 - a. In this exercise, `DH2_MyCase` case type is used.



```
ExternalData.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2<ExternalDataSystem>
3<Properties>
309
310<PropertySets>
483
484<CaseTypes>
485<CaseType>
486    <Name>DH2_MyCase</Name>
487    <PropertySetRef identifier="Top" />
488</CaseType>
489</CaseTypes>
490</ExternalDataSystem>
```



Note

See the ConfiguringSampleExternalDataXML.pdf file in the sample package for information about the structure and options of this XML file. The Sample External Data Service package is available from the developerWorks website. You can download it at this URL:

https://www.ibm.com/developerworks/community/blogs/e8206aad-10e2-4c49-b00c-fee572815374/entry/sample_external_data_service_for_ibm_case_manager_by_dave_hanson6?lang=en_us

4. Leave the file open in Eclipse for this lesson to compare the configuration with the results in the Case Manager Client.

Exercise 4.1.2: Change the field status dynamically for a property

Introduction

You can dynamically change the field status for a case property by editing the ExternalData.XML in the sample application.

In this example, you verify the field status for the City property of your case type ("MyCase"). This case property is configured to be "hidden" or "not hidden" depending on the value of the State property (value is selected from a choice list) of your case type.

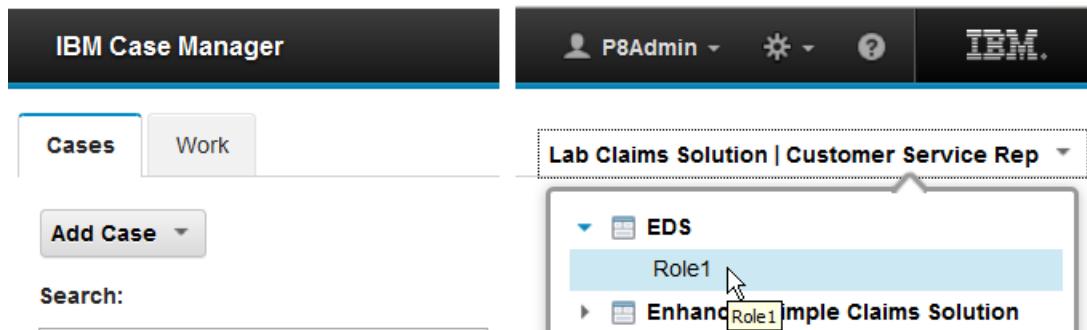
Procedures

Procedure 1: Test the EDS Service for the field status, page 10

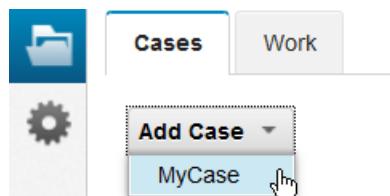
Procedure 2: Check the configuration in the XML file, page 12

Procedure 1: Test the EDS Service for the field status

1. Log in to IBM Case Manager Client in a browser.
 - URL: <http://ecmedu01:9080/navigator/?desktop=icm>
 - User name: P8admin
 - Password: IBMFileNetP8
2. Select EDS > Role1 from the upper right of the page from "Solution and Roles" list.



3. In the Cases tab, click "Add Case" > My Case to start the Add Case wizard.



4. Check that the “City” property is not visible in the Add Case wizard.
 - a. For “State”, select “California” from the list.
 - b. Verify that the “City” field is now visible, and it has a choice list with choices of cities in California.
 - c. Select “Los Angeles” or any other value for “City”.
5. Check the “City” property validation.
 - a. For “State”, select “New York” from the list.
 - b. Notice that the “City” property flags an error since the value that you selected in the previous step is no longer valid.

A screenshot of the IBM Case Manager Client interface. A dropdown menu is open under the 'City' label, showing 'Los Angeles' as the selected value. A red 'X' icon is positioned next to the dropdown, indicating an error. A tooltip message 'The value is not a member of the externally specified choice list.' is displayed in a grey box with a close button.

- c. Select “Buffalo” or any other value from the list for “City”. Verify that the error flag is not visible.
6. Check the “City” property status.
 - a. For “State”, select “Nevada” from the list.
 - b. Observe that the “City” property is not visible.
7. The following screen captures show the field status for “City” under different conditions.
 - If the value of the State property is “Nevada”, or nothing, then the City property is hidden from the display when you add a case in the IBM Case Manager Client. For other values, the City property is shown.

The image contains three side-by-side screenshots of the 'MyCase' form in the IBM Case Manager Client, illustrating the behavior of the 'City' property under different 'State' values:

- No Value for State:** The 'State' dropdown is empty. The 'City' field is visible but empty.
- Other values for the State:** The 'State' dropdown shows 'New York'. The 'City' field is visible and contains 'Buffalo'.
- Nevada for State:** The 'State' dropdown shows 'Nevada'. The 'City' field is hidden and not visible on the screen.

8. Leave the Case Manager Client open for this lesson.

Procedure 2: Check the configuration in the XML file

1. In Eclipse, if the file is not already opened, open the file.
 - a. In Package Explorer, expand SampleExternalDataService > config
 - b. Double-click the file ExternalData.xml to open it.
2. Observe that the property block for City from line 102-131.
3. Every property in the Properties section has a symbolic name as defined in the solution (DH2_City), Property data type (String), and Cardinality (single).
4. The “Hidden” element can be set true or false.
 - a. Notice that it is false when the State name is NY as defined in the Property identifier (“Ct@St=NY”).
 - b. This value makes it possible for the City field to show when New York is selected.

```
102      <!-- Dynamic info -->
103      <!-- City, when State=NY -->
104      <Property identifier="Ct@St=NY">
105          <SymbolicName>DH2_City</SymbolicName>
106          <PropertyType>string</PropertyType>
107          <Cardinality>single</Cardinality>
108          <Hidden>false</Hidden>
109
110      <HasDependentProperties>false</HasDependentProperties>
111      <ChoiceList>[]
112          <!--<ValueIfNewOrInvalid>New York</ValueIfNewOrInvalid>-->
113          <ValueIfNew>New York</ValueIfNew>
114          <!--<ValueIfInvalid handling="forceOnConfigChange">New York</ValueIfInvalid>
115      </Property>
```

5. Check the other elements for the property such as <HasDependentProperties>, <ChoiceList>, <ValueIfNew>.
6. Observe that there are many blocks of City property in this file. In each block, the Property identifier (“Ct@St=NY”) is different.
 - a. In line 134, the Property identifier is “Ct@St=CA” for the state California.
 - b. For this block also, the “Hidden” element is set false.
 - c. This value makes it possible for the City field to show when California is selected.
7. In line 171, the Property identifier is “Ct@St=NV” for the state Nevada.
 - a. For this block, the “Hidden” element is set true.
 - b. This value makes the City field not to show when Nevada is selected.
8. Locate line 450, under the following element structure.
 - a. Verify that the default “Hidden” value for City (when there is no State value) is “true”.
 - b. This value makes the City field not to show when no State is selected.

```
<ExternalDataSystem>
  <PropertySets>
    <PropertySet identifier="Top">
      <DynamicPropertySets>
        <DynamicPropertySet>
          <DefaultPropertySet>
```

```
445<
446
447
448
449
450
  <Property>
    <SymbolicName>DH2_City</SymbolicName>
    <PropertyType>string</PropertyType>
    <Cardinality>single</Cardinality>
    <!--<RenderedReadOnlyValue/>-->
    <Hidden>true</Hidden>
```

Exercise 4.1.3: Add an external choice list with EDS

Introduction

The choice lists are pre-configured so that users avoid any errors from typing the data. You can add a choice list to a case property in Case Manager Builder when you design a solution (internal choice lists).

In this lab exercise, you review how to add an external choice list with EDS.

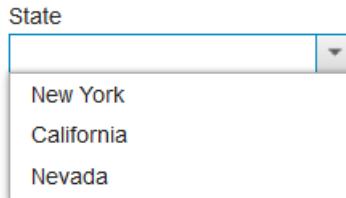
Procedures

Procedure 1: Check the EDS Service for a choice list, page 14

Procedure 2: Check the configuration in the XML file, page 14

Procedure 1: Check the EDS Service for a choice list

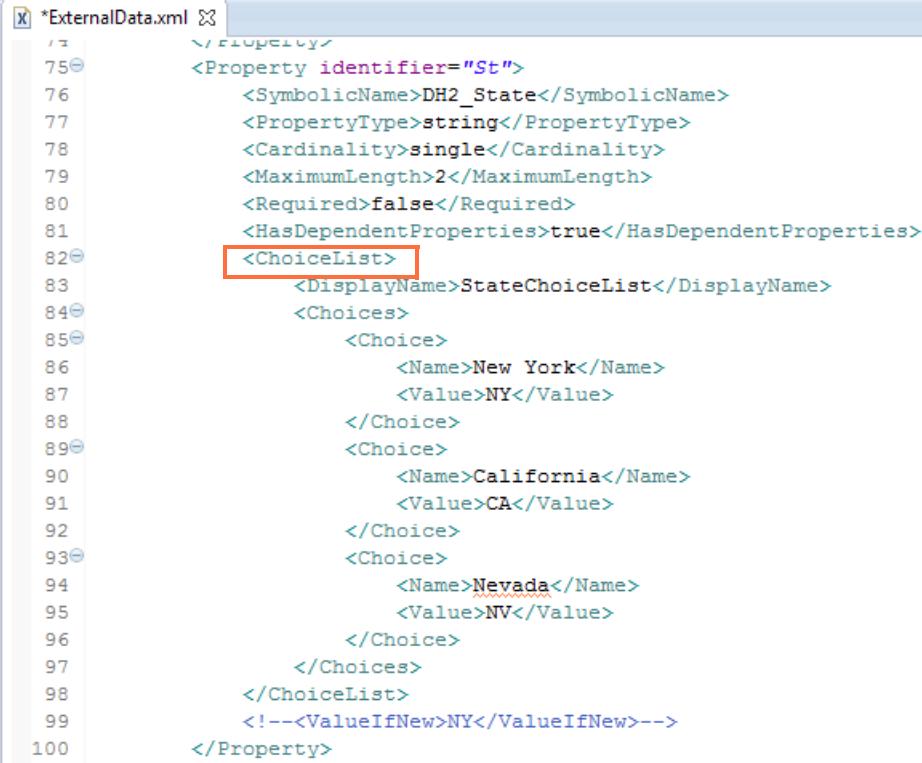
1. In IBM Case Manager Client, the “Add Case” tab is opened.
 - a. If it is not opened, repeat steps 1-3 in Ex.5.1.2. > Procedure 1: Test the EDS Service for the field status, page 10 to add a case in the Case Manager Client.
2. Recall the State property from the previous lab exercise.
 - a. The choice list for this property is implemented with the sample EDS.



Procedure 2: Check the configuration in the XML file

1. In Eclipse, if the file is not already opened, open the file.
 - a. In Package Explorer, expand SampleExternalDataService > config
 - b. Double-click the file ExternalData.xml to open it.
2. Observe the property block for State <Property identifier="St"> from line 75-100.

3. The symbolic name for this property as defined in the solution is DH2_State.
 - a. Property data type is String.
4. In line 82 - 98, a choice list is defined for this property.
 - a. Within the <ChoiceList> element, there is a <Choices> element.
 - b. With the Choices, there are individual <Choice> elements with a name and value.
 - c. These Names are displayed for the choice list in the Case Manager Client.



```
*ExternalData.xml
  ...
  <Property identifier="St">
    ...
    <ChoiceList>
      ...
      <Choices>
        ...
        <Choice>
          ...
        </Choice>
        ...
        <Choice>
          ...
        </Choice>
        ...
        <Choice>
          ...
        </Choice>
      </Choices>
    </ChoiceList>
    ...
  </Property>
```

Exercise 4.1.4: Create dependent choice lists with EDS

Introduction

If the values of one choice list depend on the selection of a value for another property, the two choice lists are called as “Dependent choice lists”.

Recall the State and City example from the previous lab exercises. When you select a State from the choice list, the choice list values for the City are updated automatically to match the State value. The State and the City represent dependent choice lists, and are implemented with the EDS.

Procedures

Procedure 1: Check the EDS Service for dependent choice lists, page 16

Procedure 2: Check the configuration in the XML file, page 17

Procedure 1: Check the EDS Service for dependent choice lists

1. In IBM Case Manager Client, the “Add Case” tab is opened.
 - a. If it is not opened, repeat steps 1-3 in Ex.5.1.2. > Procedure 1: Test the EDS Service for the field status, page 10 to add a case in the Case Manager Client.
2. Recall the State and City properties from the previous lab exercise.
 - a. If the value of the State property is “Nevada”, then the City property is hidden from the display when you add a case in the IBM Case Manager Client.
 - b. When you select New York or California as a value for State, the values for the City property choice list dynamically change to match the State.

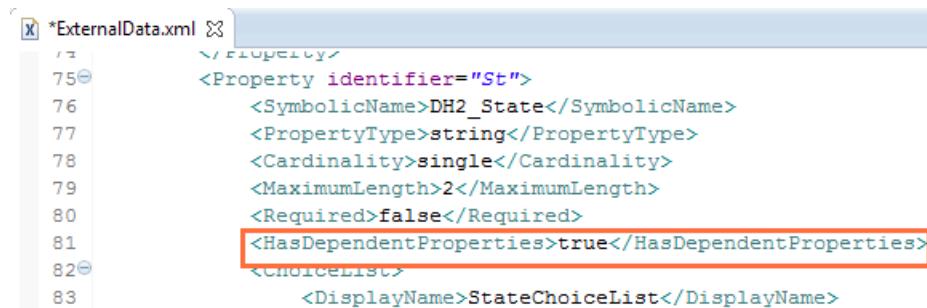
The figure consists of two side-by-side screenshots of the IBM Case Manager Client interface, specifically the "Add Case" tab.

Screenshot 1 (Left): This screenshot shows the "City" property. The dropdown menu is open, displaying three options: Buffalo, New York, and Rochester. Below the dropdown, there is a note: "MVInt 0, 100" and a dropdown menu set to "One". Further down, there is a "PropOne" input field and a "State" dropdown menu. The "State" dropdown is currently set to "New York".

Screenshot 2 (Right): This screenshot shows the same "City" property after a selection has been made. The dropdown menu is now closed, and the selected value "Los Angeles" is displayed in the input field. The "PropOne" input field and the "State" dropdown menu are also visible. The "State" dropdown is now set to "California".

Procedure 2: Check the configuration in the XML file

1. In Eclipse, if the file is not already opened, open the file.
 - a. In Package Explorer, expand SampleExternalDataService > config
 - b. Double-click the file ExternalData.xml to open it.
2. Observe the property block for State <Property identifier="St"> from line 75.
3. In line 81, <HasDependentProperties> element with the value "true" defines that other property values change depending on the value of this property.



```
*ExternalData.xml
75<Property identifier="St">
76    <SymbolicName>DH2_State</SymbolicName>
77    <PropertyType>string</PropertyType>
78    <Cardinality>single</Cardinality>
79    <MaximumLength>2</MaximumLength>
80    <Required>false</Required>
81    <HasDependentProperties>true</HasDependentProperties>
82    <ChoiceList>
83        <DisplayName>StateChoiceList</DisplayName>
```

4. In line 171, the Property identifier is "Ct@St=NV" for the state Nevada.
 - a. For this block, the "Hidden" element is set true.
 - b. This value makes the City field not to show when Nevada is selected.
5. Locate line 399, under the following element structure.

```
<ExternalDataSystem>
    <PropertySets>
        <PropertySet identifier="Top">
            <DynamicPropertySets>
                <DynamicPropertySet>
                    <ConditionalPropertyName>DH2_State</ConditionalPropertyName>
                    <ConditionalPropertySets>
                        <ConditionalPropertySet>
                            a. Check that the condition is "<Equals> NY </Equals>" for the State property and it is linked to "<PropertyRef identifier="Ct@St=NY" />".
                            b. This condition instructs the client to use the choice list with New York cities.

```

```

392<DynamicPropertySets>
393<DynamicPropertySet>
394    <ConditionalPropertyName>DH2_State</ConditionalPropertyName>
395    <ConditionalPropertySets>
396        <ConditionalPropertySet>
397            <!-- One of: -->
398            <!-- Equals, can be single or multi-value -->
399            <Equals>NY</Equals>
400
401            <!-- Or, Between. Only for single value. -->
402            <!-- Between>
403                <LowerValue>some_value1</LowerValue>
404                <UpperValue>some_value2</UpperValue>
405            < / Between -->
406
407            <!-- Or, Includes. Only for a multi-value prop
408                but specifies a single value to test against. -->
409            <!-- Includes>some_value</Includes -->
410
411            <!-- Inline set to include city property when state=NY
412            <PropertySet>
413                <StaticProperties>
414                    <PropertyRef identifier="Ct@St=NY" />
415                </StaticProperties>
416            </PropertySet>
417
418        </ConditionalPropertySet>

```

- c. "<Property identifier="Ct@St=NY">" choice list is defined in lines 104 -131.
d. This condition makes the City field to list the cities in New York.

```

102    <!-- Dynamic info -->
103    <!-- City, when State=NY -->
104    <Property identifier="Ct@St=NY">
105        <SymbolicName>DH2_City</SymbolicName>
106        <PropertyType>string</PropertyType>

```

6. For California, the city values are defined in lines 134-168, and for Nevada in lines 171-180.

Exercise 4.1.5: Implement case property validation

Introduction

In this exercise, you implement a case property validation in the IBM Case Manager Client with the sample EDS.

Procedures

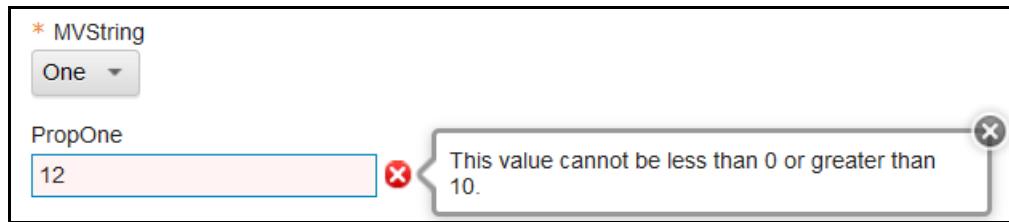
Procedure 1: Check the EDS Service for the case property validation, page 19

Procedure 2: Check the configuration in the XML file, page 19

Procedure 1: Check the EDS Service for the case property validation

The PropOne case property in the MyCase case type uses a data validation feature from the sample EDS.

1. In IBM Case Manager Client, the “Add Case” tab is opened.
 - a. If it is not opened, repeat steps 1-3 in Ex.5.1.2. > Procedure 1: Test the EDS Service for the field status, page 10 to add a case in the Case Manager Client.
2. For the PropOne field, enter a number that is greater than 10.
 - a. Notice that an error message shows that the value must be 1 - 10.



- b. If you enter any number 1 - 10, verify that there are no errors.

Procedure 2: Check the configuration in the XML file

1. In Eclipse, if the file is not already opened, open the file.
 - a. In Package Explorer, expand SampleExternalDataService > config
 - b. Double-click the file ExternalData.xml to open it.
2. Observe that the property block for P1 <Property identifier="P1"> from line 5-74.

3. The symbolic name for this property as defined in the solution is DH2_PropOne.
 - a. Property data type is integer.
4. In line 21 and 24, notice that maximum and minimum values are defined for this property.
 - a. These two values define the data validation for the PropOne field.

```
*ExternalData.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ExternalDataSystem>
3 <Properties>
4     <!-- Static info -->
5     <Property identifier="P1">
6         <SymbolicName>DH2_PropOne</SymbolicName>
7
8         <!-- integer/string/id/boolean/float/datetime -->
9         <PropertyType>integer</PropertyType>
10
11        <!-- single/multi -->
12        <Cardinality>single</Cardinality>
13
14        <!-- optional. Always return this value in the payload with
15            a display mode of readonly. No content means null.
16            This element takes precedence over any other value
17            handling element (ValueIfNew, ValueIfInvalid, etc.) -->
18        <!-- RenderedReadOnlyValue -->
19
20        <!-- optional -->
21        <MaximumValue>10</MaximumValue>
22
23        <!-- optional -->
24        <MinimumValue>0</MinimumValue>
25
26        <!-- optional -->
27        <!-- MaximumLength -->
```

Exercise 4.1.6: Prefill an initial value for a case property

Introduction

In this lab exercise, you prefill an initial value for a case property in the IBM Case Manager Client with the sample EDS application.

Procedures

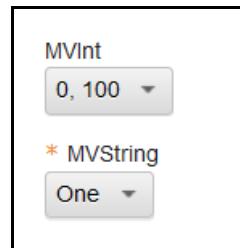
Procedure 1: Check the EDS Service for an initial value of a case property, page 21

Procedure 2: Check the configuration in the XML file, page 21

Procedure 1: Check the EDS Service for an initial value of a case property

The MVInt and MVString case properties in the MyCase case type has initial values from the sample EDS when you open the Add Case wizard.

1. In IBM Case Manager Client, the “Add Case” tab is opened.
 - a. If it is not opened, repeat steps 1-3 in Ex.5.1.2. > Procedure 1: Test the EDS Service for the field status, page 10 to add a case in the Case Manager Client.
2. Verify that the MVInt and MVString fields have the following initial values.
 - MVInt: 0, 100
 - MVString: One



Procedure 2: Check the configuration in the XML file

1. In Eclipse, if the file is not already opened, open the file.
 - a. In Package Explorer, expand SampleExternalDataService > config
 - b. Double-click the file `ExternalData.xml` to open it.
2. Observe the property block for `DH2_MVInt` from line 378 – 389.

3. This property has Cardinality as “multi” and so it can have multiple values.
 - a. Property data type is integer.
4. In line 385-388, the `<ValueIfNewOrInvalid>` element defines the initial value for this property.
 - a. Because it is a multi-value property, two values are assigned.

```
378④      <Property>
379          <SymbolicName>DH2_MVInt</SymbolicName>
380          <PropertyType>integer</PropertyType>
381          <Cardinality>multi</Cardinality>
382          <MaximumValue>1000</MaximumValue>
383          <MinimumValue>0</MinimumValue>
384          <HasDependentProperties>true</HasDependentProperties>
385④          <ValueIfNewOrInvalid>
386              <Value>0</Value>
387              <Value>100</Value>
388          </ValueIfNewOrInvalid>
389      </Property>
```



Hint

The properties in this file are defined in the “`<Properties>`” section, and conditions are defined in the “`<PropertySets>`” section. To show as an alternative method, the `DH2_MVInt` property is defined in the “`PropertySets`” section

5. Observe the property block for `DH2_MVString` from line 184 – 225.
6. This property also has Cardinality as “multi” and so it can have multiple values.
 - a. Property data type is String.
7. In line 221-223, the `<ValueIfNew>` element defines the initial value for this property.

```
220          </ChoiceList>
221④          <ValueIfNew>
222              <Value>One</Value>
223          </ValueIfNew>
224          <ValueIfInvalid handling="replaceValue" />
225      </Property>
```

Exercise 4.1.7: Set the case property field status as required

Procedures

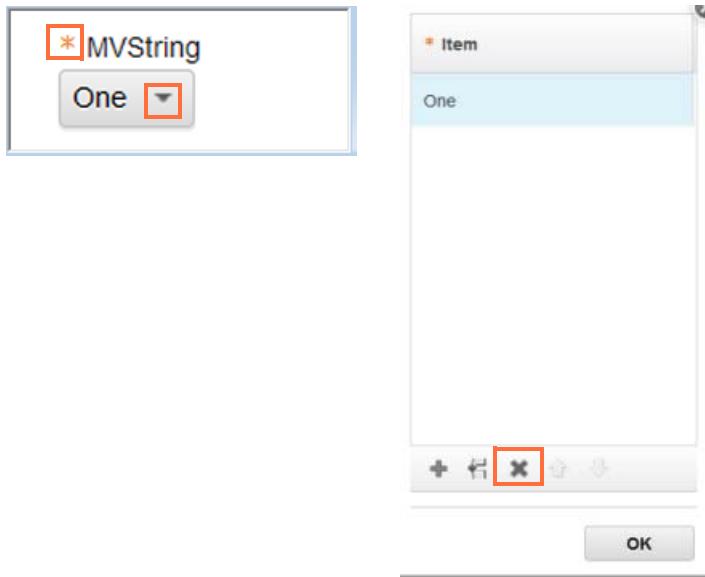
Procedure 1: Check the EDS Service for the case property field status (required), page 23

Procedure 2: Check the configuration in the XML file, page 24

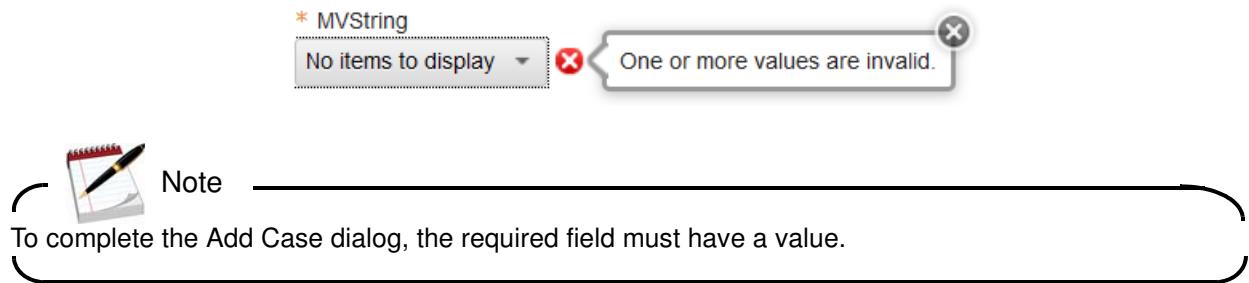
Procedure 1: Check the EDS Service for the case property field status (required)

The MVString case property in the MyCase case type has a field status as “required” through the sample EDS when you open the Add Case wizard.

1. In IBM Case Manager Client, the “Add Case” tab is opened.
 - a. If it is not opened, repeat steps 1-3 in Ex.5.1.2. > Procedure 1: Test the EDS Service for the field status, page 10 to add a case in the Case Manager Client.
2. Verify that the MVString field name has a red star icon. This icon indicates that this field requires a value.
3. Since it has a prefilled value, remove the value to test its requirement.
 - a. Click the down-arrow next to the value.
 - b. In the Item window, select “one”, and click the X icon to delete it.



- c. Click OK.
4. Verify that you get an error since the required field is empty.



Procedure 2: Check the configuration in the XML file

1. In Eclipse, if the file is not already opened, open the file.
 - a. In Package Explorer, expand SampleExternalDataService > config
 - b. Double-click the file ExternalData.xml to open it.
2. Observe the property block for DH2_MVString from line 184 – 225.
3. In line 190, the <Required> element with “true” as a value defines the “Required” status for this property.

```
184      <!-- MVString when MVInt = [0,100] -->
185      <Property identifier="MVs@mvI=0_100">
186          <SymbolicName>DH2_MVString</SymbolicName>
187          <PropertyType>string</PropertyType>
188          <Cardinality>multi</Cardinality>
189          <MaximumLength>24</MaximumLength>
190          <Required>true</Required>
191          <HasDependentProperties>false</HasDependentProperties>
```

The screenshot shows a code editor with XML code. Line 190 is highlighted with a red rectangle, specifically the '<Required>true</Required>' part. The code defines a property named 'DH2_MVString' with various attributes like type ('string'), cardinality ('multi'), and maximum length ('24'). The 'Required' attribute is set to 'true'.

4. Log out of the IBM Case Manager Client and close the Browser.
 - a. Exit Eclipse.

Appendix: Steps to set up the sample External Data Service

Introduction

The sample external data service for IBM Case Manager is distributed as a self-contained ZIP file. The ZIP file includes:

- Java source files, configuration files, and an Apache Ant script to build the application.
- Technical documents that describe how to build and use the sample.



Note

The Sample External Data Service package is available from the developerWorks website. You can download it at this URL:

https://www.ibm.com/developerworks/community/blogs/e8206aad-10e2-4c49-b00c-fee572815374/entry/sample_external_data_service_for_ibm_case_manager_by_dave_hanson6?lang=en_us

The sample is a Java servlet-based web application. You must complete the following steps:

- Deploy it to a WebSphere Application Server environment.
- After the deployment, register the URI of the service to a solution in IBM Case Manager.

In this appendix, you create a WAR file with the sample files, deploy, and register the service.

Procedures

Procedure 1: Create a WAR file for the sample in Eclipse, page 25

Procedure 2: Install the sample external data service, page 28

Procedure 3: Register the external data source, page 29

Procedure 4: Verify the external data source registration, page 30

Procedure 1: Create a WAR file for the sample in Eclipse

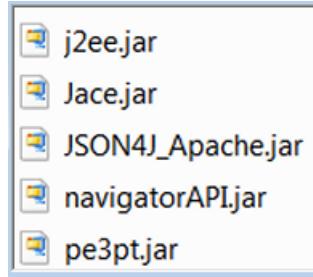
1. Open Eclipse by double-clicking the Eclipse icon in your desktop.
 - a. In the Workspace Launcher page, leave the default workspace directory (`C:\ICM\workspace_Eclipse`) and click OK.
2. Open the Java Project creation wizard.
 - a. In Eclipse, click File > New > Java Project.
 - b. In the New Java Project page, enter a name (`SampleExternalDataService`).
 - c. Click Next.

3. Add the required libraries.

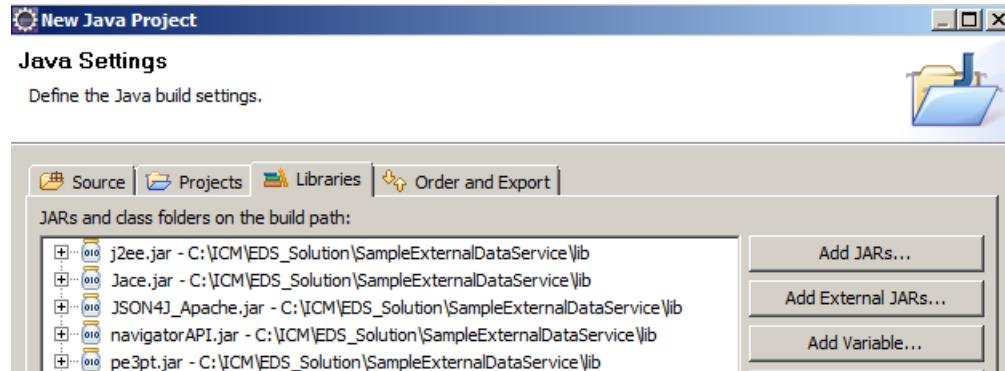
- In the Java Settings page, select the Libraries tab.
- Click “Add External JARs”.
- In the “JAR Selection” window, go to a directory where you have the following libraries:

 Note

The library files are in the following folder on the system where IBM Case Manager is installed:
C:/Program Files/IBM/CaseManagement/configure/exploded_apps/pi_was/WEB-INF/lib. For the student image, these files are copied to the lib folder in the Sample External Data Service sample.



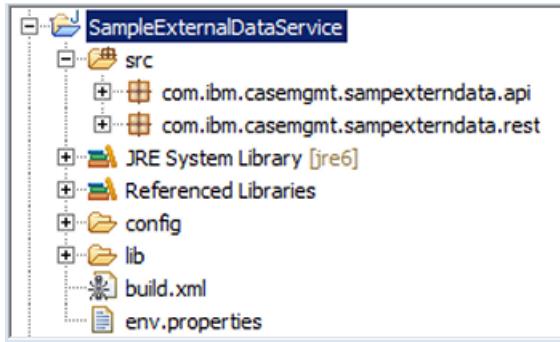
- The completed page looks like the one in the screen capture.



- Click Finish to complete the project wizard.
- Copy the sample EDS files from windows explorer and paste it in your project.

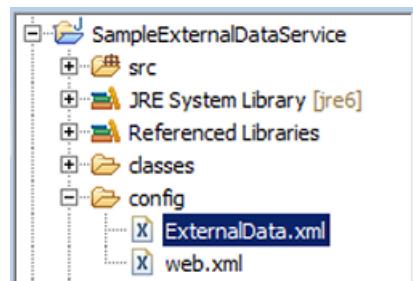
Name	Date modified
config	8/12/2014 7:43 PM
src	8/12/2014 7:43 PM
build.xml	8/12/2014 7:42 PM
ConfiguringSampleExternalDataXML.pdf	8/12/2014 7:42 PM
env.properties	8/12/2014 7:42 PM
UsingSampleExternalDataService.pdf	8/12/2014 7:42 PM

Your project structure looks like the following screen capture.



 Requirements

An XML file in the sample external data service defines what case types and properties the sample data service manages. You edit the ExternalData.XML file to add the case properties in your solution to control the external data service. See the ConfiguringSampleExternalDataXML.pdf file in the developerworks download package for information about the structure and options of this XML file.



6. The build.xml file that packages your project into a WAR file is included in the sample. Verify the library file location and edit them to point to the correct location.
 - a. Right-click the build.xml and select Run As > “2.Ant Build” from the list.
 - b. Validate that the console tab at the bottom pane displays that the Build was successful.
 - c. Verify that the message indicates that the WAR file is created.
 - d. Ignore the “includeantruntime” warning.

Procedure 2: Install the sample external data service

1. Start the WebSphere Integrated Solutions Console.
 - URL: <http://ecmedu01:9043/ibm/console/logon.jsp>
 - User name: P8Admin
 - Password: IBMFileNetP8
2. Expand Applications and select New Application.
3. Select the externaldataservice.war file and 'Open'.
4. Click Next and accept defaults in the wizard excepting for the following steps:
 - a. Edit the application name.
 - b. Specify a ContextRoot (Example: "CaseEDS").
You use this URL path to register the service.
5. After the wizard completes, select "Save to master configuration" to complete the installation and configuration.
6. Start the external data service (CaseEDS).
7. Check to ensure that context root property is set properly.
 - a. Click the external data service and click "Context Root for Web Modules".
 - b. The Context Root must be "CaseEDS" as you configured.
8. Enter the following URL in a browser to test the service:
<http://ecmedu01:9080/CaseEDS/ICMEDREST/types>
 - a. Verify that the browser shows a response with the case type in your solution. (MyCase)



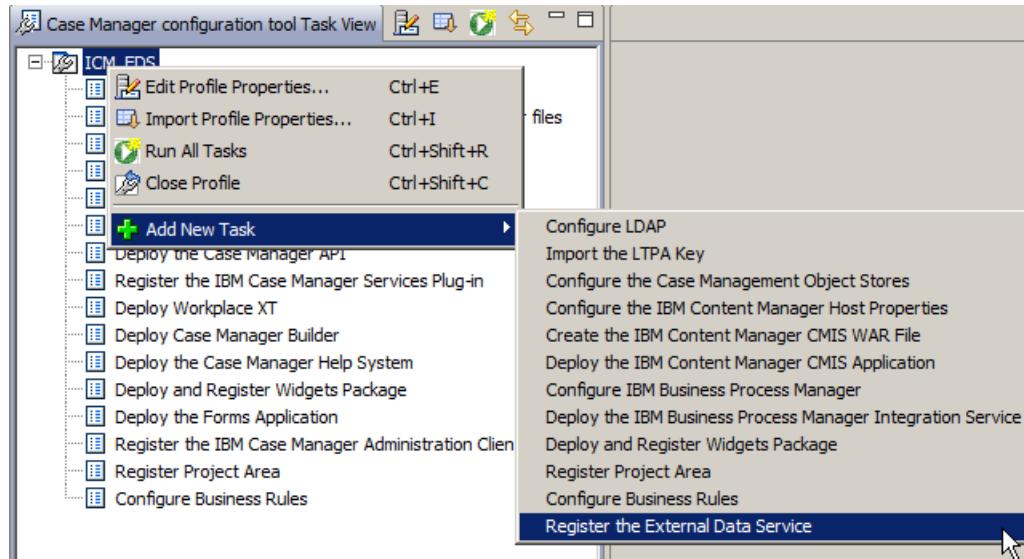
Note

Your student system already has a case Solution that contains a case type with the name "MyCase".

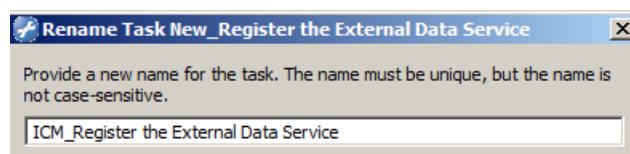
- b. Note down the URL to use it in the next procedure.

Procedure 3: Register the external data source

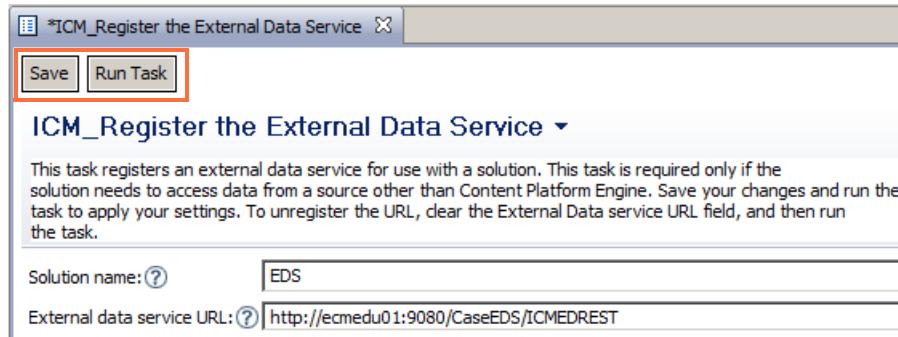
1. In Case Manager Builder, create a solution. You deploy it later.
2. Double-click the Case Manager Configuration tool icon on the desktop.
 - a. You can also start it from Windows Start > All Programs > IBM Case Manager > Case Manager Configuration Tool.
3. Select File > Open Profile to open the configuration profile.
 - a. In the open window, go to the C:\Program Files (x86)\IBM\CaseManagement\configure\profiles\ICM Lab folder.
 - b. Select ICM Lab.cfgp and click Open.
4. Expand the ICM Lab node.
5. Right-click on any task, and select Add New Task > Register the External Data Service.



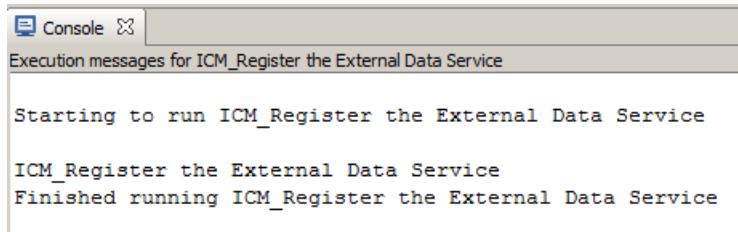
6. Optionally, right-click and rename the newly created task.



7. Double-click the task to edit it:
 - a. In the right pane, select EDS from the list for Solution name.
 - b. External data service URL: <http://ecmedu01:9080/CaseEDS/ICMEDREST>
8. Save the changes. Run the task. It takes a few moments.

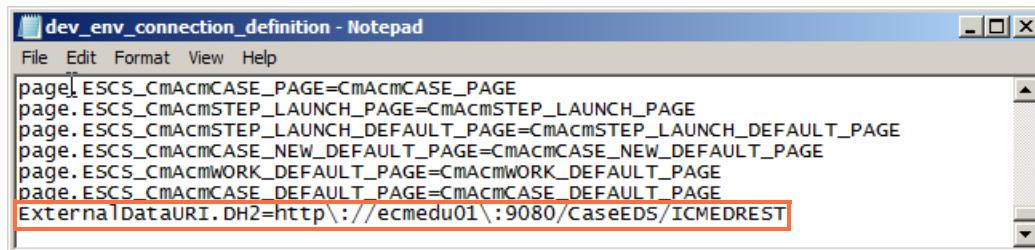


9. Click Yes for the warning to enable and run the task.
10. Validate that the task ran successfully in the console below the task.



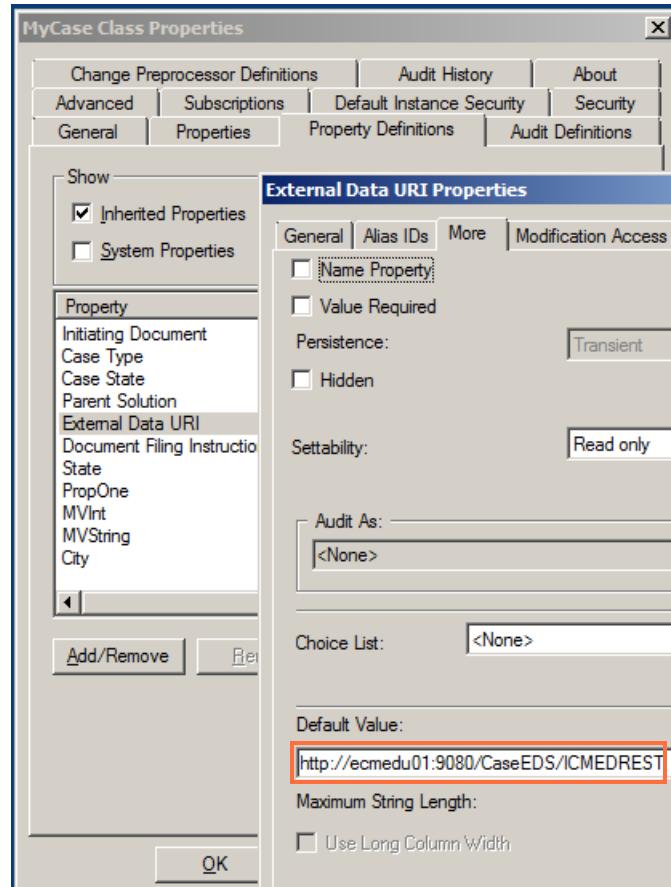
Procedure 4: Verify the external data source registration

1. Double-click the FileNet Enterprise Manager icon on the desktop.
 - a. Go the development object store (DevDOS).
 - b. Expand DevDOS > Root Folder > IBM Case Manager > Connection Definitions
2. Verify that the URL is added to the folder properties.
 - a. Open the connection definition file in notepad.
 - b. Scroll to the end of the file to confirm that the ExternalDataURI property is set.



3. In Case Manager Builder, deploy your solution.

4. In the FileNet Enterprise Manager, go to your target object store (DevTOS).
 - a. Go to DevTOS > Other Classes > Folder > Base Case > Case Folder > MyCase.
 - b. Right-click your case and select Properties.
 - c. In the Properties window, select the “Property Definitions” tab.
 - d. Select Show > Inherited Properties.
5. Scroll down, and select “External Data URI” Property.
 - a. Click Edit.
 - b. In the “External Data URI Properties” window, select the “More” tab.
 - c. Verify that the “Default Value” field on the bottom of the dialog contains the URL that entered earlier in the Case Manager Configuration tool.



6. Close all the applications.

5

Appendix

This Appendix contains supplemental material not included in the main course material.

© Copyright IBM Corp. 2014

Course materials may not be reproduced in whole or in part
without the prior written permission of IBM.

APPENDIX A: Start and Stop System Components

Procedures

- Procedure 1: Start WebSphere Application Server
- Procedure 2: View the Content Engine Startup Context page
- Procedure 3: Open the Content Platform Engine log files
- Procedure 4: View web applications
- Procedure 5: Shutdown individual components
- Procedure 6: Restart the system components

Procedure 1: Start WebSphere Application Server

WebSphere Application Server hosts the following applications:

- Tivoli Directory Server Administration tool
 - Content Platform Engine
 - IBM Content Navigator
 - IBM Case Manager applications
 - Administration Console for Content Platform Engine
1. Click Start > All Programs > IBM WebSphere > IBM WebSphere Application Server V8.5 > Profiles > AppSrv01 > Start the server.
 2. Wait for the Start the server window to close.



Note

You can also use the Start_Server1.bat file in the WebSphere Admin folder on the desktop to start WebSphere Application Server.

Procedure 2: View the Content Engine Startup Context page

The Content Engine Startup Context (Ping Page) provides detailed information about the Content Platform Engine. You can use the Ping Page to quickly determine whether the Content Platform Engine is running.

1. Use Firefox to open the Content Engine Startup Context page.
 - <http://ecmedu01:9080/FileNet/Engine>
 - In general, the URL is <http://<server>:<port>/FileNet/Engine>.
2. Verify the following information:

- Product name: P8 Content Platform Engine
- Log file location: C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSvr01\FileNet\server1
- P8 Domain: P8Domain
- JDBC Driver: IBM DB2 JDBC Universal Driver

Procedure 3: Open the Content Platform Engine log files

If startup problems occur, the Content Platform Engine log files provide information for troubleshooting.

1. Go to the location of the log files that you identified in the previous procedure and view them in WordPad or Notepad++.

- p8_server_error.log
- p8_server_trace.log
- pesvr_system.log
- pesvr_trace.log



Note

The Content Platform Engine has two main services: Content Services and Process Services. Both services create log files in the same location. Content Service log files begin with “p8,” while Process Service log files begin with “pesvr.”

Procedure 4: View web applications

You can view the applications that are running in the administrative console. From there, you can also shut down and restart any web application individually.

1. Use Firefox to open the administrative console:
 - <https://ecmedu01:9043/ibm/console>
 - In general, the URL is `https://<server>:<port>/ibm/console`
2. Log in to WebSphere Integrated Solutions Console as the `P8Admin` user.
 - Password: `IBMFFileNetP8`
3. In the navigation pane, go to Applications > Application Types > WebSphere enterprise applications.
4. Verify that the following applications are running (shows a green arrow in the Application Status column):

Application name	Purpose
FileNetEngine	Content Platform Engine
IDSWebApp	Directory Services Web Administration console.

CaseManager	IBM Case Manager Client
CaseBuilder	IBM Case Manager Builder
ivtApp	WebSphere installation verification test application
navigator	IBM Content Navigator
query	WebSphere query service
CaseEDS	A sample external data service

Procedure 5: Shutdown individual components

In this procedure, you shut down the components individually. You are viewing the Enterprise Applications in the WebSphere Integrated Solutions Console.

1. Stop IBM Case Manager applications:
 - a. In WebSphere Integrated Solutions Console, select a Case Manager application.
 - b. Click Stop.
 - c. Wait for the Application Status to show that it stopped.
2. Stop FileNetEngine.
3. Log out of WebSphere integrated Solutions Console.
4. Stop WebSphere Application Server:
 - a. Click Start > All Programs > IBM WebSphere > IBM WebSphere Application Server V8.5 > Profiles > AppSrv01 > Stop the server.
 - b. Enter the following user name and password to stop the server:
 - User name: P8Admin
 - Password: IBMFileNetP8



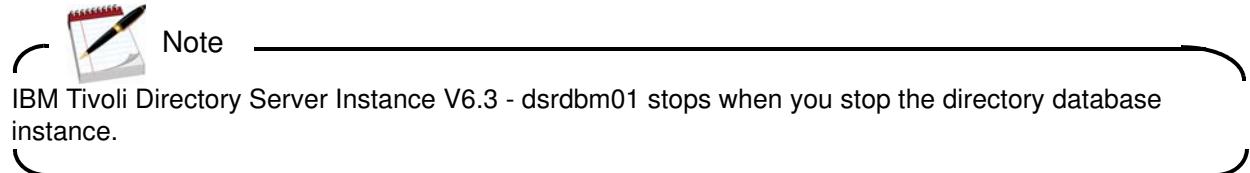
Note

You can also use Stop the Server.bat in the WebSphere Admin folder on the desktop. The batch file enters the user name and password for you.

5. Stop the database server software for the Content Platform Engine:
 - a. Open the Windows Services Console: Click Start > Services.
 - b. Locate the two DB2 server instances:

DB2 - TDSV63DB2 - DB2TDS63-0	Allows appl... Started Automatic .\P8Admin
DB2 - TDSV63DB2 - DSRDBM01	Allows appl... Started Automatic Local System

- c. Stop these two database services.



Procedure 6: Restart the system components

1. Start the Directory Server and Content Platform Engine database DB2 server instances in this order:
 - a. DB2 - TDSV63DB2 - DSRDBM01 (Directory Services DB2 instance)
 - b. DB2-TDSV63DB2 - DB2TDS63-0 (Content Platform Engine DB2 instance)
 - c. IBM Tivoli Directory Server Instance V6.3 - dsrdbm01 (Directory server)
 2. Start WebSphere Application Server.
-

APPENDIX B: Debugging and troubleshooting

This section describes troubleshooting techniques that can be used when you customize and extend IBM Case Manager. Use the information along with IBM Case Manager documentation for troubleshooting.

Topics

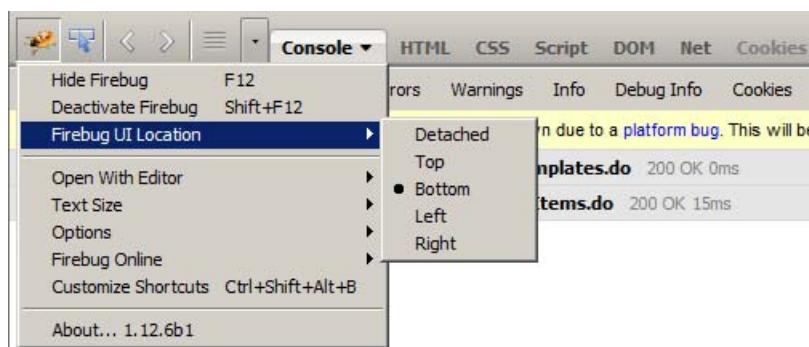
Debugging in the Firefox browser on page 5-6

Configure logging in WebSphere Application Server on page 5-7

Debugging in the Firefox browser

Firebug (a powerful client debugging tool for Firefox) is installed as a Firefox add-on in your student system.

1. Press F12 to launch the Firebug in the browser.
2. In the Firebug options, it can be set to any location of the browser or separately in an individual window.



3. The following panels in Firebug are useful for different purposes.
 - Console: Show logs and all requests and response content.
 - HTML: Show page HTML source
 - CSS: Show CSS of current page
 - Script: Show scripts. You can debug JavaScript code. You can set breakpoints, watch variable values, and check the whole stack.
 - DOM: Show all dom information.
 - Net: Show network transmission information, such as URL, status, size, and timeline.

- Cookie: Show all cookie information.



Note

For more information about Firebug, see: <http://addons.mozilla.org/en-US/firefox/addon/firebug>

Configure logging in WebSphere Application Server

Case Manager Builder and the IBM Case Manager API application log information to the WebSphere Application Server main log and trace log.

- You can redirect the WebSphere Application Server log to record information for these components in the IBM Case Manager log location.
- You can also change the logging levels in WebSphere Application Server.



Note

For more information about logging, see IBM Case Manager V5.2 documentation at this URL:
http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/com.ibm.casemgmt.design.doc/acmta004.htm

APPENDIX C: Additional information and links

IBM Case Manager Redbook

- IBM Case Manager Redbooks publication introduces the case management concept.
 - Title: Advanced Case Management with IBM Case Manager
 - URL: <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>
- This book:
 - Includes the reason for and benefits of case management, and why it is different from the traditional business process management or content management.
 - Addresses how you can design and build a case management solution with IBM Case Manager and integrate that solution with external products and components.
 - Is intended to provide IT architects and IT specialists with the high-level concepts of case management and the capabilities of IBM Case Manager.
 - Serves as a practical guide for IT professionals who are responsible for designing, building, customizing, and deploying IBM Case Manager solutions.

Links to Information Centers and Resources

IBM Case Manager Version 5.2 Information Center

- http://www.ibm.com/support/knowledgecenter/SSCTJ4_5.2.0/

IBM Redbooks publication: Advanced Case Management with IBM Case Manager

- <http://www.redbooks.ibm.com/abstracts/sg247929.html?Open>

IBM Developer Works Technical Library

- <http://www.ibm.com/developerworks/library/>

Course Wrap Up

Course Wrap Up provides you with a list of training options offered by IBM, identify additional help, and enable you to submit an evaluation of the class.

Completed Objectives

Completed Course Objectives

This course was designed to enable you to:

- Customize the Case Manager Client User Interface
 - Customize the banner and the login page
 - Associate a custom icon for a MIME type
 - Modify labels in the IBM Case Manager Client
 - Create a Viewer Map for PDF files
 - View the Microsoft Word documents in the FileNet Viewer
 - Edit the existing help topics
 - Create a help plug-in
 - Customize the toolbar to implement actions
 - Add a custom action as a menu item
- Use Scripts to Customize Case Manager Client
 - Use Script Adapter to customize the Case client
 - Create a toolbar button to start a task
 - Dynamically add a choice list to a case property
 - Validate the data based on a step response
 - Create a case custom workbench page
 - Configure your system for the workbench page
 - Add a Script Adapter to filter In-baskets
 - Add a Script Adapter to filter cases
- Develop Custom Widgets
 - Create an IBM Content Navigator plug-in project in Eclipse
 - Create catalog and widget definition JSON files
 - Implement a custom widget (Custom search)
 - Build and register the widget package
 - Test the custom widget
 - Create a Java project in Eclipse for a widget package
 - Implement toolbar and menu for your widget
 - Define widget properties, and add event handling for your widget
 - Build and deploy the widget package with an EAR file
 - Troubleshoot
 - Update an existing widgets package with new widgets
(Custom case comment widget as an example)
 - Uninstall a custom widget package in IBM Case Manager

- Implement External Data Services (EDS)
 - Check the sample EDS configuration
 - Change the field status dynamically for a property
 - Add an external choice list with EDS
 - Create dependent choice lists with EDS
 - Implement case property validation
 - Prefill an initial value for a case property
 - Set the case property field status as required
 - Appendix: Steps to set up the sample External Data Service
-

Course Evaluation

Completing the Evaluation

To ensure that the “IBM Case Manager 5.2: Customize and Extend the Features” course is as effective as possible and meets the needs of customers, please share your feedback. Feedback on the site, curriculum, and instructor tell us what was good about the class and what can be improved.

Take the time to complete the online course evaluation on the IBM Training website:

https://survey6.spss-asp.com/mriweb/mriweb.dll?i.project=ibmlearningsrvs_learningservicesbasesurveyjuly2013

You will need the Course Code in order for you to complete the online course evaluation.

Thank you for your valuable inputs.

If you are interested in additional services, please go to the IBM Training website:

<http://ibm.com/training/>

IBM
®