

Course Guide

IBM Case Foundation 5.2.1: Routing Work

Course code F240 ERC 1.0



March 2016 edition

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

Trademarks

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

© Copyright International Business Machines Corporation 2011, 2016.

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

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Trademarks v	
Course description vi	
Agenda viii	
Conditional routing 1	
Why is this lesson important to you? 2	
Unit objectives 3	
Conditional routing 4	
Step responses 6	
Route properties 8	
Step routing information 11	
Example: Route based on participant responses 13	
Demonstration 14	
Unit summary 15	
Exercise: Use conditional routing in a workflow 16	
Exercise introduction 17	
Parallel processing 1	
Why is this lesson important to you? 2	
Unit objectives 3	
Modeling parallel execution 4	
Example: Simple parallel processing 6	
Collection and data merge types 8	
How data merging works 9	
Example: Data merging in a multi-participant step 11	
Rules for creating valid workflow maps (1) 12	
Rules for creating valid workflow maps (2) 13	
Use of TerminateBranch system function 14	
Use of an unassigned step 16	
Demonstration 18	
Unit summary 19	
Exercise: Define a parallel process in a workflow 20	
Exercise introduction 21	
Manage participation 1	
Why is this lesson important to you? 2	
Unit objectives 3	
Workflow participants 4	
Workflow groups 5	
Example: Dynamic assignment of workflow groups 7	
System-provided workflow group: F_Trackers 8	
System-provided workflow group: F_Originator 9	
Set participant privileges 11	
Participant voting 13	
Example: Counting responses 15	
Demonstration 17	
Unit summary 18	
Exercise: Control how users participate in a workflow 19	
Exercise introduction 20	
Work with submaps 1	
Why is this lesson important to you? 2	

Unit objectives	3
Using submap steps	4
Define a new submap	6
Example: Using a submap step	7
Implicit and explicit return from a called map	8
Default system maps	10
Workflow termination	12
Override a system map	13
Demonstration	14
Unit summary	15
Review questions	16
Review questions	17
Review answers	18
Review answers	19
Exercise: Use workflow submaps	20
Exercise introduction	21
Handle workflow exceptions	1
Why is this lesson important to you?	2
Unit objectives	3
Business process exceptions	4
Malfunction map and the Conductor queue	5
Override the Malfunction system map	6
Using Return in a Malfunction map	8
Workflow troubleshooting and recovery	10
Example: Workflow exception	12
Demonstration	13
Demonstration(2)	14
Unit summary	15
Exercise: Resolve business process exceptions	16
Exercise introduction	17
Workflow inheritance	1
Why is this lesson important to you?	2
Unit objectives	3
Using workflow inheritance	4
Inherited elements	6
The base workflow	8
Workflow Preferences for inheritance	9
Create a workflow hierarchy	11
Override an inherited map	12
Unit summary	14
Exercise: Use workflow inheritance	15
Exercise introduction	16

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

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

Bluemix®
SPSS®

Cognitive Era™
Watson Avatar®

Cognos®
Worklight®

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

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

Other product and service names might be trademarks of IBM or other companies.

Course description

IBM Case Foundation 5.2.1: Routing Work

Duration: 1 day

Audience

This course is for workflow authors who are responsible for planning, designing, creating, testing workflow solutions in a development environment.

Prerequisites

- Familiarity with Windows 2008 operating systems. General knowledge of P8 Platform security concepts.
- General workflow terminology:
 - Workflow
 - Workflow definitions
 - Queues
 - Rosters
- Start a P8 Platform system.
- Familiarity with P8 Platform administration interfaces, including:
 - Administration Console for Content Platform Engine
 - IBM Content Navigator
 - Process Designer

Objectives

- Use conditions to direct the flow of work.
- Create a parallel process in a workflow.
- Create workflow group.
- Define participant voting for a step.
- Define participant voting for a step.
- Use a submap
- Identify system default submaps
- Resolve business process exceptions
- Create a workflow hierarchy

- Override an inherited map

Contents

- Conditional routing
- Parallel processing
- Manage participation
- Work with submaps
- Handle workflow exceptions
- Workflow inheritance

Curriculum relationship

To learn about related courses, visit the IBM authorized training website:

<http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=page&c=Y678448H04759K32>

Agenda

**Note**

The following unit and exercise durations are estimates, and might not reflect every class experience.

Day 1

(00:15) Course introduction

(00:30) Unit 1. Conditional routing

(00:30) Exercise 1. Use conditional routing in a workflow

(0:35) Unit 2. Parallel processing

(00:35) Exercise 2. Define a parallel process in a workflow

(00:40) Unit 3. Manage participation

(00:50) Exercise 3. Control how users participate in a workflow

(00:30) Unit 4. Work with submaps

(00:40) Exercise 4. Use workflow submaps

(00:30) Unit 5. Handle workflow exceptions

(00:45) Exercise 5. Resolve business process exceptions

(00:30) Unit 6. Workflow inheritance

(00:30) Exercise 6. Use workflow inheritance

Unit 1. Conditional routing

Estimated time

00:30

Overview

This unit shows how to add step responses and conditional routing to a workflow definition using Process Designer.

How you will check your progress

- Successfully complete the lesson exercises.

References

IBM Knowledge Center

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8toc.doc/welcome_p8.htm

Why is this lesson important to you?

- You are designing a workflow application. At a point in the workflow, the process flow depends on the response given by a workflow participant at a step. You need to add step responses and conditional routing to your workflow definition.

Conditional routing

© Copyright IBM Corporation 2011, 2016

Figure 1-1. Why is this lesson important to you?

Unit objectives

- Use conditions to direct the flow of work

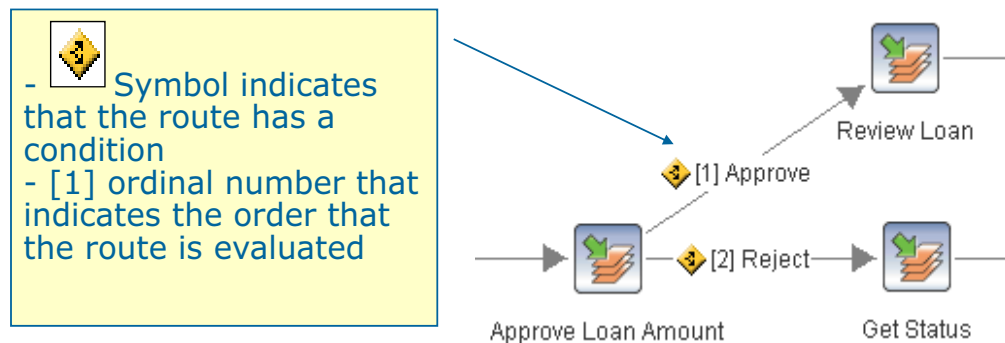
Conditional routing

Figure 1-2. Unit objectives

© Copyright IBM Corporation 2011, 2016

Conditional routing

- For a step, multiple outbound routes can be based on conditions, such as
 - One or more participant responses at the step
 - The value of an expression by using one or more workflow fields
 - A combination of responses and expressions
- Route conditions are evaluated in order.
 - You control the order in the step routing properties.
- Best practice: Label routes with meaningful names.



Conditional routing

© Copyright IBM Corporation 2011, 2016

Figure 1-3. Conditional routing

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About routing

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh019.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About routing>Create a route between steps

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh039.htm

Routes

A route, represented by an arrow symbol between two steps, defines the sequence in which steps are processed. A workflow starts from the launch step. Every step must have one or more routes that lead from it, except for the last step on the map. In a valid workflow map, every step in the workflow definition must be able to be reached from the launch step.

Multiple outbound routes

A step can have multiple outbound routes. The workflow designer defines the conditions that control which route to follow. Multiple outbound routes from a step can be based on the following types of conditions.

- The responses given by one or more participants at the step
- The value of an expression that is evaluated by using one or more workflow data fields
- A combination of responses and expressions

You can use a combination of step properties and route properties to specify conditional routing in a workflow.

Best practice for routing

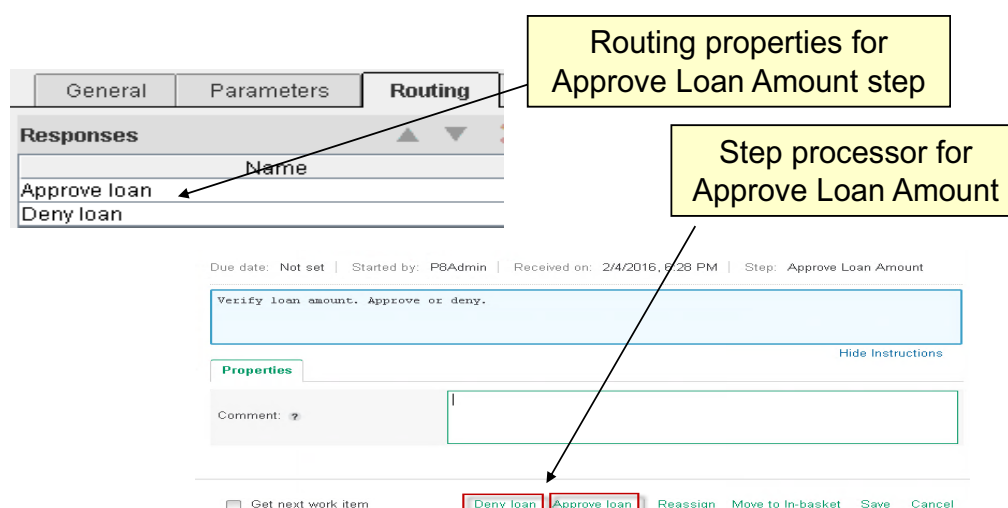
It is a good idea to assign meaningful, unique names to all multiple outbound, and inbound routes. Use a distinctive route name to make it easier to identify individual routes and maintain and edit the workflow.

Example of conditional routing

The example diagram on this page shows conditional routing for the outgoing routes of the Approve Loan Amount step in a loan processing workflow. In this example, at the Approve Loan Amount step the Loan Manager verifies loan details and the loan amount. If the manager approves the loan, the workflow proceeds along the Approve route to the Review Loan step. If the manager denies the loan, the workflow proceeds along the Reject route to the Get Status step. The route conditions are evaluated in the order specified by the number displayed in front of the route name.

Step responses

- You can define responses for an Activity step.
 - At design time, responses are defined in the Routing tab of the step properties.
 - At run time, the participant selects a response.
- Example: Approve Loan Amount step has two responses.
 - Approve loan or Deny loan



Conditional routing

© Copyright IBM Corporation 2011, 2016

Figure 1-4. Step responses

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About steps>About Activity steps>Activity step - routing properties

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh068.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About routing>Associate a response with a route

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh041.htm

For Activity steps, you can specify responses that the workflow participant can select. In many workflows, the response that the participant selects determines which step is processed next. For a multi-participant step, all the participants must complete the step before any responses are evaluated. The participant is required to click a response in the step processor.

You specify the responses on the Routing tab in the step properties. You cannot define responses for step types other than an Activity step.

Step responses example

The screen capture on the left shows the participant responses defined in the Routing tab for the Approve Loan Amount step properties. The screen capture on the right shows the resulting display in the step processor at run time when the step is run. The participant clicks a response to process the step.

Two system fields exist that capture the values of the responses selected at run time.

F_Responses

F_Responses is a system field that is an integer array. The array lists each response in a step with a count of the number of participants that choose the response. Responses are positional, based on the order in which you defined the routing responses for the step.

F_Response

Is a system field that is a string. F_Response is derived from F_Responses and contains the string value of the response selected.

Both system fields are transient and must be assigned to a workflow data field after completion of the step where the response is selected.

Route properties

- Always true (no condition)
- Conditional Route based on responses
 - You must first define the responses for participants to select.
- Conditional Route based on data fields
 - Evaluates an expression you define

The image displays two screenshots of the 'Route' configuration window in the IBM P8 Platform. Both screenshots show the 'Route' name as 'Approve' and the 'Routing' type as 'Conditional Route'.

The left screenshot shows the 'Data Fields' tab selected. The 'Field' is 'status (String)', the 'Operator' is 'is equal', and the 'Value' is 'Approved'. A yellow callout box below the dialog states: 'Evaluates to true when status = "Approved"'. A red box highlights the 'Data Fields' tab.

The right screenshot shows the 'Responses' tab selected. The 'Condition' is 'ALL', the 'Response' is 'Approve loan', and the 'Operator' is 'is equal'. A yellow callout box below the dialog states: 'Evaluates to true when all participants select the "Approve loan" response'. A red box highlights the 'Responses' tab.

Conditional routing

© Copyright IBM Corporation 2011, 2016

Figure 1-5. Route properties

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About routing>Specify route properties

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh153.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About routing>Specify a routing expression

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh038.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About routing>Routing expression example

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh040.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About routing>Routing expression example - test for attachment

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh042.htm

For each route, you specify a name and indicate whether the route is always true or has conditions. Routes that are always true are not subject to evaluation. For conditional routes, you can specify an evaluation expression or a participant response.

Route name

Routes are automatically labeled with numbers, which are based on the order of creation in the workflow. Optionally, you can provide names for routes, which are displayed on the workflow map provided the workflow preferences are set to show route names. It is a good practice to label all multiple outbound routes with meaningful names that make it easier to maintain the workflow definition.

Route properties

For a route, you can select one of the following two options.

- Select *Always true* when the route does not have a condition and the workflow always takes this route.
- Select *Conditional Route* to specify a condition on this route, which can be based on a response or on a data field expression. You can build an expression by using the buttons and menus in the Responses or Data Fields tabs, or by typing an expression directly in the field.
 - If you specify Conditional Route based on responses, the responses that the participant selects must already be defined in the step properties.
 - If you specify Conditional Route based on a data field expression, the data fields that are used in the expression must already be defined in the workflow properties. If the expression evaluates to true, the route is taken.

You can create an expression in a conditional route that uses a combination of both responses and data fields.



Important

If no route that leads from a step evaluates to "true" and the work item is a child work item, it is deleted. If the work item is the root work item, it is terminated and the workflow is also terminated. To avoid this situation, ensure that at least one route of multiple outgoing routes evaluates to true. A good way to ensure that at least one valid route exists is to set the route properties for the last route to be evaluated to "Always true."

Conditional routing example

The examples that are shown on this page are a continuation of the Approve Loan Amount example introduced in the previous pages. The decision of which route to take is based on whether the loan is approved or not. The example screen captures show two different ways of evaluating whether to take the route called Approve.

- The screen capture on the left, tests the value of a data field, for example, status = "Approved."
- The screen capture on the right, uses participant responses, for example, ALL (Approve loan).

**Note**

When entering an expression with Boolean values, be aware that the Boolean values true and false are case-sensitive. These values are valid in lowercase characters only.

Step routing information

- Outgoing routing information
 - You specify which outgoing routes to take.
 - “All true conditions” or only the “First true condition” that is evaluated
 - Outgoing routes are evaluated in the order listed.
 - You can change the order of evaluation by using the arrows.
 - Click Details to view the routing conditions, if any.
- Incoming routing information
 - You can designate the step as a collector step.
 - Used for parallel processing

Conditional routing

© Copyright IBM Corporation 2011, 2016

Figure 1-6. Step routing information

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About steps>About Activity steps>Activity step - routing properties

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh068.htm

In addition to step responses, you can also define the routing options for both incoming and outgoing routes in the step properties. These properties are specified in the step properties Routing tab.

Outgoing routing information

By default, an outgoing route is always taken unless a condition is placed on it.

If you want only one valid route to be taken from a step, select the “First true condition” option in the step properties Routing tab. If only one outgoing route exists, it is taken. If multiple outgoing routes from a step exist, you design the decision for which route to take based on:

- Participant responses defined in the step properties
- Conditional expressions defined in the route properties
- A combination of both responses and expressions

You must specify the route conditions for each route. If multiple true route conditions are possible and you want all possible valid routes to follow this step, then specify All true conditions. This option is used to design a workflow with parallel processing.

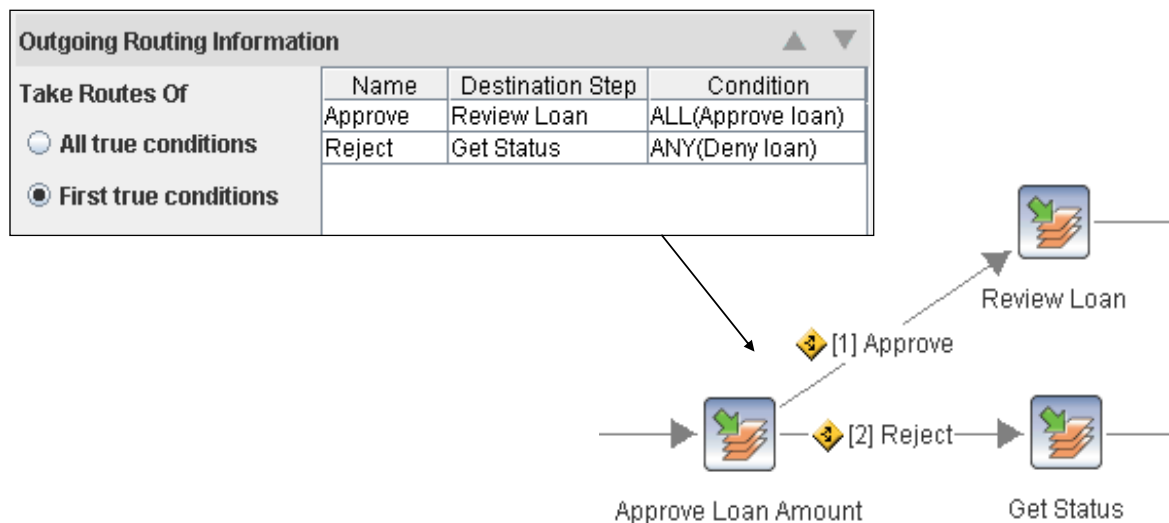
You can click Details to view the conditions for all routes, if any.

Incoming routing information

You can designate whether the step is a collector step. A collector step, also called an AND-join step, is used in modeling parallel processing.

Example: Route based on participant responses

- Outgoing routes from Approve Loan Amount step
 - If **all** responses are “Approve loan”, take the **[1] Approve route** to the Review Loan step.
 - If **any** response is “Deny loan”, take the **[2] Reject route** to the Get Status step.



Conditional routing

© Copyright IBM Corporation 2011, 2016

Figure 1-7. Example: Route based on participant responses

This example is a continuation of the example on the previous pages. The Approve Loan Amount step has two outgoing routes with conditions: Approve and Reject. When you open the Routing tab for the Approve Loan Amount step properties, the Outgoing Routing Information section shows the routing details, as shown in the screen capture.

- If all participants Approve the loan, take the [1] Approve route.
- If any participant denies the loan, take the [2] Reject route.

Demonstration



- Define a simple workflow with four activity steps.
- Define two data fields to store the values of F_Response and F_Responses.
- Add three responses to the first step.
 - Valid
 - Not Valid
 - Cancel
- Define conditional routes, based on the responses.
- Assign the system data fields F_Response and F_Responses to workflow data fields after completion.
- Test the workflow.
 - Select different responses so different routes are taken.

If you are taking this course as a self-paced online course, return to the main course menu and click the link to play the recorded demonstration.

Conditional routing

© Copyright IBM Corporation 2011, 2016

Figure 1-8. Demonstration

Unit summary

- Use conditions to direct the flow of work

Conditional routing

Figure 1-9. Unit summary

© Copyright IBM Corporation 2011, 2016

Exercise: Use conditional routing in a workflow

Conditional routing

© Copyright IBM Corporation 2011, 2016

Figure 1-10. Exercise: Use conditional routing in a workflow

Exercise introduction

- Define step responses
- Define conditional routes based on responses
- Define conditional routes based on data fields



Conditional routing

© Copyright IBM Corporation 2011, 2016

Figure 1-11. Exercise introduction

Unit 2. Parallel processing

Estimated time

0:35

Overview

This unit shows how to model parallel execution in a workflow definition.

How you will check your progress

- Successfully complete the lesson exercises.

References

IBM Knowledge Center

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8toc.doc/welcome_p8.htm

Why is this lesson important to you?

- You are designing a workflow application. Your process has several steps that are run in parallel to save time. You must model parallel execution in your workflow.

Parallel processing

© Copyright IBM Corporation 2011, 2016

Figure 2-1. Why is this lesson important to you?

Unit objectives

- Create a parallel process in a workflow

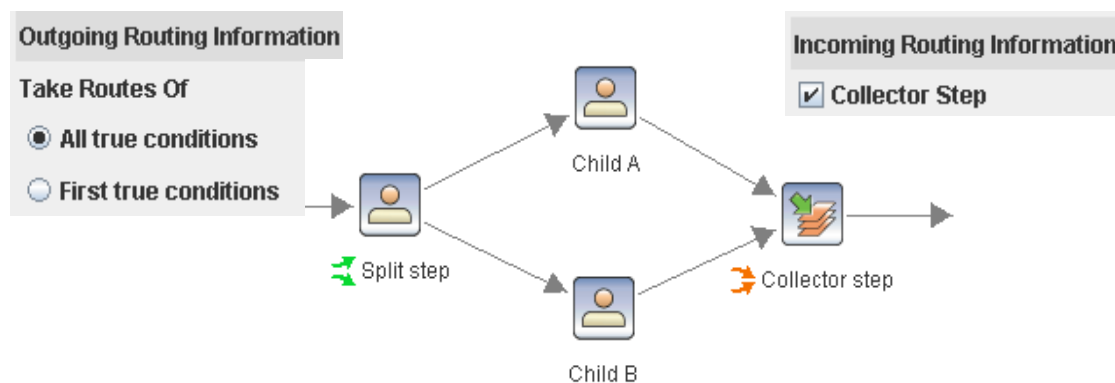
Parallel processing

© Copyright IBM Corporation 2011, 2016

Figure 2-2. Unit objectives

Modeling parallel execution

- Define an AND-split step
 - On the Step Properties > Routing tab, select the “All true conditions” option for outgoing routes.
 - Create multiple true outgoing routes from the step.
- Define a collector step (also known as an AND-join step)
 - Required to join multiple, parallel routes
 - Joins the work back into a single path at end of all true routes



Parallel processing

© Copyright IBM Corporation 2011, 2016

Figure 2-3. Modeling parallel execution

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About steps>About Activity steps>Activity step - routing properties

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh068.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About steps>About system steps>System step - routing properties

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh087.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Getting started in Design mode>Preferences - display options in Design mode

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh033.htm

A workflow can simultaneously follow multiple routes during its processing. For example, in an insurance claims processing workflow, a claims adjuster can report on property damage at the same time that a medical reviewer prepares a report of injuries originating from the claim. Because neither step depends on the other, these two activities can continue in parallel, which saves time.

You model this requirement for parallel execution by using AND-split and Collector (AND-join) steps. This feature splits the main work item into a number of concurrent processes with the AND-split, and then brings the workflow back to a single collection point with the Collector (AND-join) step.

Define an AND-split step

To model parallel processing, you define multiple outgoing routes from an Activity step, where more than one route can evaluate to true. On the Routing tab in the Properties pane, select the “All true conditions” option found in the Outgoing Routing Information area. A split icon is displayed next to the AND-split step name, if the Collector step icon is enabled in Process Designer display preferences (default setting is enabled). Work passed from this step continues along all true routes. Where more than one active step results from such a branch, the multiple steps are considered child steps of the split step. The numbers that show evaluation order for multiple routes when the “First true condition” option is selected are not displayed on the workflow map when the “All true conditions” option is selected.

Define a Collector (AND-join) step

If you have an AND-split step, you must define a subsequent Activity step in the path as a Collector step (also known as an AND-join step). The AND-join step can immediately follow the AND-split step, or one or more steps can be in between.

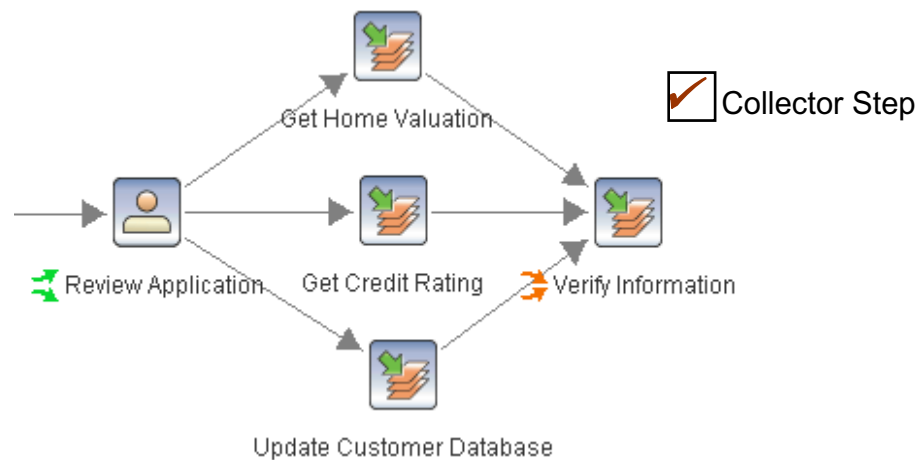
To specify an AND-join step, select the step on the workflow map. In the Routing tab of the Properties pane, select the Collector Step check box, which is found under Incoming Routing information. A collector icon is displayed next to the collector step name if the Collector step icon is enabled in Process Designer display preferences (default setting is enabled).

Example

The diagram on this page shows an example map that contains a parallel process. Workflow steps Child A and Child B are run in parallel. The Split step is an example of an AND-split step. The Collector step is an example of an AND-join step.

Example: Simple parallel processing

- The Review Application step specifies “All true conditions” for outgoing routes.
- Three steps are processed concurrently followed by the collector step.
- Three work items are created to follow each of the three paths.



Parallel processing

© Copyright IBM Corporation 2011, 2016

Figure 2-4. Example: Simple parallel processing

Simple example of parallel processing

In this example of parallel processing, a workflow is launched when a customer applies for a home mortgage loan. After the loan application is reviewed, three separate tasks run, which are independent of each other: update the customer database with basic information about the applicant, get a credit rating report for the applicant, and get a home valuation (an appraisal of the property).

In the workflow definition, the outgoing routing properties for the Review Application step are assigned by using the “All true conditions” option. In this example, the three outgoing routes are specified as unconditional routes that are always true. Three steps are processed concurrently: Update Customer Database, Get Credit Rating, and Get Home Valuation.

Three work items are created that follow each of the three separate paths. The root or parent work item remains in the Delay system queue until the three child work items complete processing.

When all three steps complete, the three work items are merged and processing continues with the Verify Information step. The Verify Information step is designated as a collector step in this example. It functions as the AND-join step to the AND-split that occurs at the Review Application step.

Extended example of parallel processing: Combining AND-split and conditional routes

In the example shown on this page, each of the three outgoing routes from the Review Application step is specified as an unconditional route that is always true. Conditional routes can also be used in an AND-split. For example, the route from Review Application to Update Customer Database can include a condition that depends on the value of a Boolean data field, `new_customer`. After the AND-split, the Update Customer Database path is taken only if the value of `new_customer` is true.

Collection and data merge types

- Collection occurs in
 - Collector step (AND-join step)
 - Multi-participant steps
- Collection is composed of:
 - The parent work item waits in the Delay system queue until child processing is complete.
 - Field merging of the child items' data fields to the parent work item
- Data merge type is defined in Workflow Properties for a data field and governs how the fields are merged.
 - Data merge types for simple data types: Default, Override, Add
 - Data merge types for arrays: Default, Override, Override Entry, Append, Append No Dup, Add

Parallel processing

© Copyright IBM Corporation 2011, 2016

Figure 2-5. Collection and data merge types

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Merge types

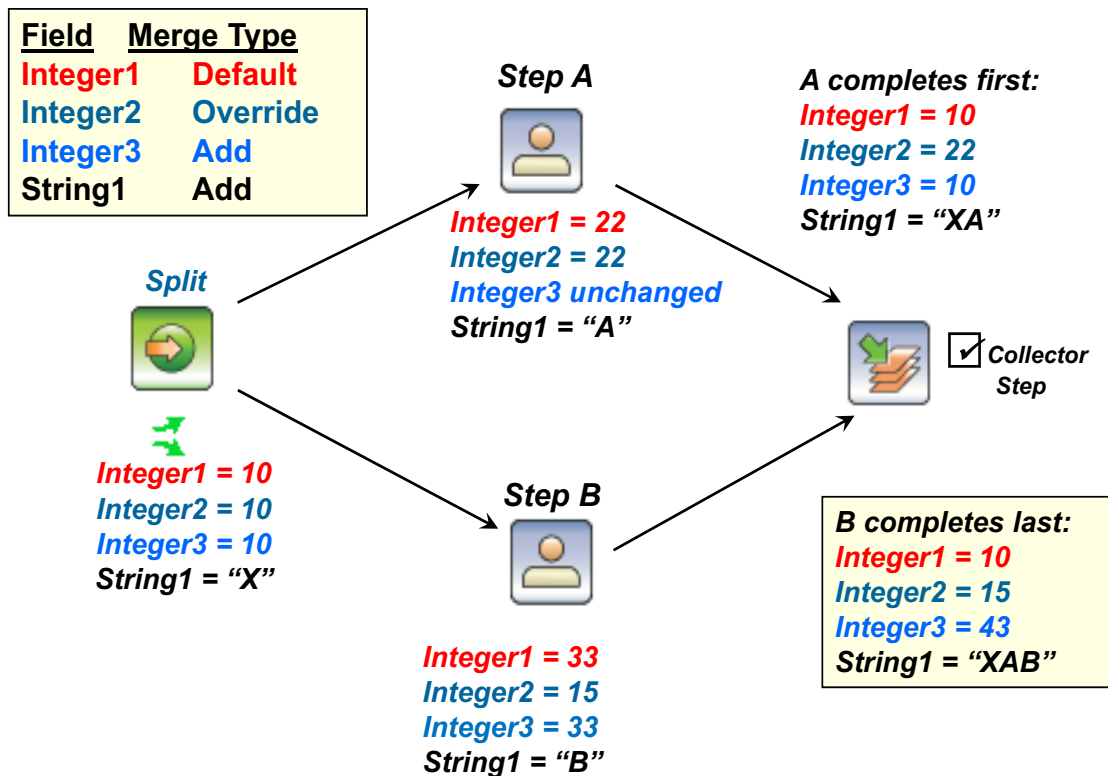
http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh027.htm

In the case of an AND-join or a multi-participant step, work items are processed concurrently. When data values vary due to the tasks run in each path, the Process Designer merge options determine how values are merged or which values are retained in the collection step.

Steps assigned to multiple participants consist of several work items, with one work item created per participant. Similarly, an AND-split creates copies of the root or parent work item, with one work item for each outgoing route after the AND-split. In both cases, the parent work item (also called the root work item) waits in the Delay system queue until all of its children finish processing. When all the work items of the AND-split or the multi-participant step finish processing, the fields from the child work items are merged with the parent work item.

How data fields values are merged in the parent work item depends on the merge type defined for the data field in Workflow Properties. Several merge types are available for both simple data types and for arrays. For an explanation of each of the merge types, see the FileNet P8 Help topic "Merge types."

How data merging works



Parallel processing

© Copyright IBM Corporation 2011, 2016

Figure 2-6. How data merging works

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Field merge integer array example

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh025.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Field merge string array example

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh026.htm

The diagram on this page illustrates how data merging works in a sample parallel process. The example is constructed to show various data types, merge types, and how they are merged.

Merge type examples for simple data types

When the Override merge type for a simple data type is selected, the values that are entered in the last step that completes before the collector step, override any previous values.

When the Add merge type for a string data field is selected, the strings are concatenated. For a float or integer field, the values are added.

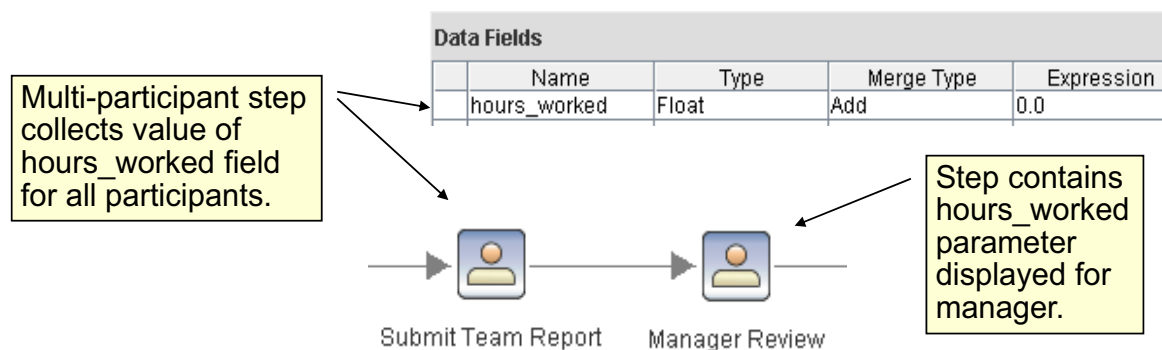
When the Default merge type is selected, the field does not change. The value of the field remains the same as it was set at the split.

The example workflow definition begins with a Launch step in which the outgoing routes are defined as parallel execution (All true conditions option is selected). One route goes to Step A, and one route goes to Step B. Steps A and B each have a single participant. Step A completes first, and then Step B completes.

The starting fields and merge types are shown in the box at the upper left of the page. The ending field values, after Step B completes, are shown in the box on the lower right.

Example: Data merging in a multi-participant step

- Use case scenario
 - Manager of a project team wants to review total hours worked.
- Example solution
 - Submit Team Report is a multi-participant step completed by each team member.
 - The hours_worked parameter is assigned a value by each team member.
 - Float field uses Add merge type.



Parallel processing

© Copyright IBM Corporation 2011, 2016

Figure 2-7. Example: Data merging in a multi-participant step

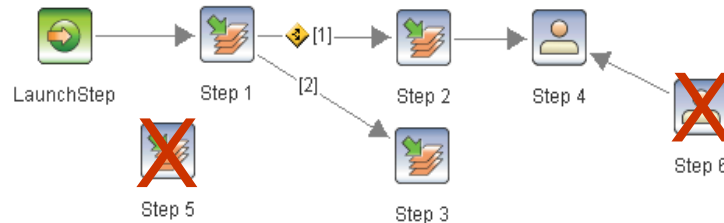
This slide contains an example of data merging in a multi-participant step. In the example map, you see two steps: the Submit Team Report step followed by the Manager Review step.

A manager for a project team requires weekly reports from team members on project activities. The manager wants to review the total hours worked on the project. The Submit Team Report step is a multi-participant step completed by all team members. Each team member provides a report and assigns a value to the hours_worked parameter. The hours_worked data field is a Float data type and Add data merge type is selected. After all team members complete the step, the system collects the values of hours_worked and adds the values. The manager views the total hours worked in the Manager Review step.

Rules for creating valid workflow maps (1)

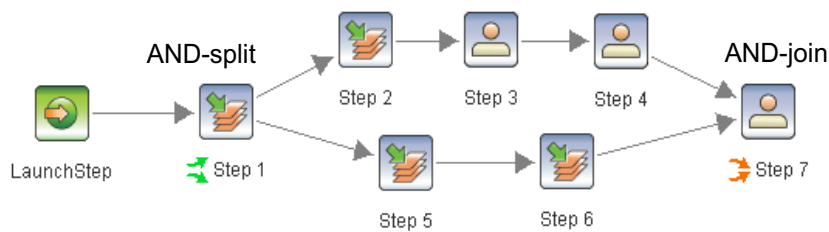
- All steps must be reachable from the Launch step.

- Example 1

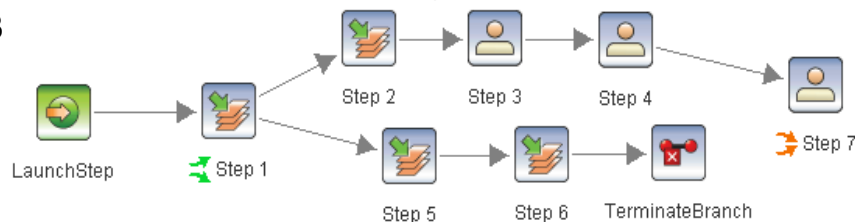


- AND-split must have one corresponding AND-join.

- Example 2



- Example 3



Parallel processing

© Copyright IBM Corporation 2011, 2016

Figure 2-8. Rules for creating valid workflow maps (1)

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Validate, transfer, and launch a workflow definition or workflow collection>Create a valid workflow map

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh133.htm

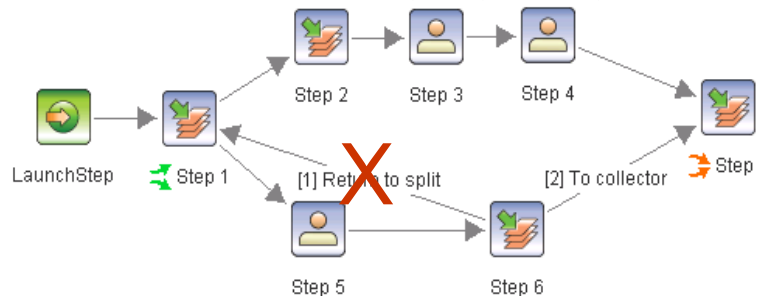
The following rules are enforced during workflow validation. Use these guidelines when creating your workflow maps. This slide shows examples of these rules.

- A workflow starts at the Launch step, and all steps must be reachable from the Launch step. Example 1 on this slide represents an invalid map. The routing used for both Step 5 and Step 6 is not valid because these steps cannot be reached in a path from the Launch step.
- An AND-split must have one corresponding AND-join. The AND-join can follow immediately, or one or more steps can be placed between the AND-split and the AND-join. Example 2 on this slide represents a valid map.
- Not all paths within a split need to join. One or more paths can terminate. Example 3 on this slide, also represents a valid map.

Rules for creating valid workflow maps (2)

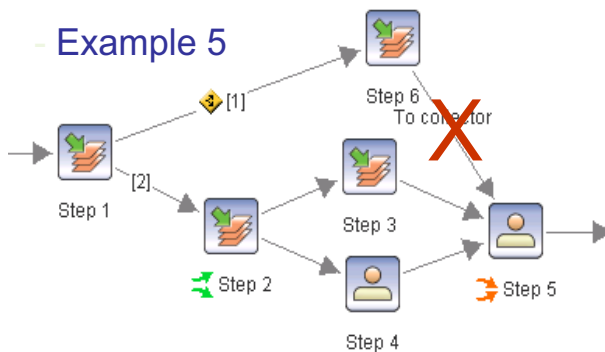
- Cannot return to AND-split without first going through collector

Example 4



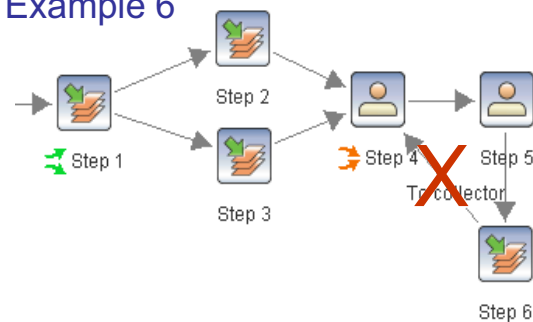
- Cannot go to AND-join without first going through AND-split

Example 5



Parallel processing

Example 6



© Copyright IBM Corporation 2011, 2016

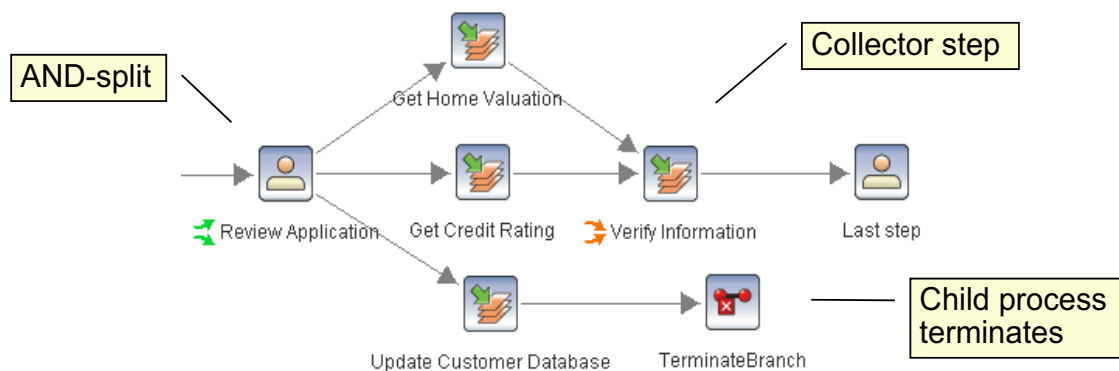
Figure 2-9. Rules for creating valid workflow maps (2)

This slide shows more rules that are enforced during workflow validation.

- A path that passes through an AND-split step cannot return to that step without first passing through the corresponding AND-join step. Example 4 represents an invalid map. The “Return to split” route from Step 6 to Step 1 is invalid because it does not first go through Step 7, the AND-join step.
- All paths that pass through an AND-join step must first pass through the corresponding AND-split step. Example 5 represents an invalid map. The “To collector” route from Step 6 to Step 5 is invalid because it does not first go through Step 2, the AND-split step.
- A path that passes through an AND-join step cannot return to that step without first passing through the corresponding AND-split step. Example 6 represents an invalid map. The “To collector” route from Step 6 to Step 4 is not a valid route.

Use of TerminateBranch system function

- Use case scenario
 - In a workflow with an AND-split, the main process cannot wait for one of the branches of the split to complete.
- Design solution example
 - Use TerminateBranch to end processing on one branch of the split.
 - Result:
 - Parent work item does not wait for Update Customer Database.
 - Data fields are merged at the collector step and processing continues.



Parallel processing

© Copyright IBM Corporation 2011, 2016

Figure 2-10. Use of TerminateBranch system function

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About steps>System functions>General step activity>TerminateBranch system function

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh114.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About steps>System functions>General step activity>TerminateProcess system function

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh115.htm

Use case scenario for the TerminateBranch system function

In a loan application processing workflow, three steps can be processed simultaneously: Get Home Valuation, Get Credit Rating, and Update Customer Database. On occasion the Update Customer Database does not need to complete and any data fields that are updated should be ignored.

To prevent the parent work item from waiting for the completion of the Update Customer Database step, you place a TerminateBranch system step (on the General System palette) after the Update Customer Database step. When TerminateBranch system function is run, that child process is terminated. The data field values of the Update Customer Database step are not merged at the collector step. The processing of the other two paths in the workflow continues.

At the collector step, Verify Information, the data field values from the Get Home Valuation and Get Credit Rating steps are merged. Processing continues with the Last step without waiting for completion of the Update Customer Database step.

Two system functions for terminating work items:

- TerminateBranch – prematurely ends the processing of a work item.
- TerminateProcess - ends the processing of all work items in a workflow and calls the Terminate system map.

For more details, see the FileNet P8 documentation topics.

Use of an unassigned step

- Not assigned to a work queue or a participant
- Can be used in some cases to facilitate flow
 - As a dummy step to evaluate routing conditions or perform calculations and assignments
 - As an AND-split step
 - As a placeholder for cycling back in a workflow

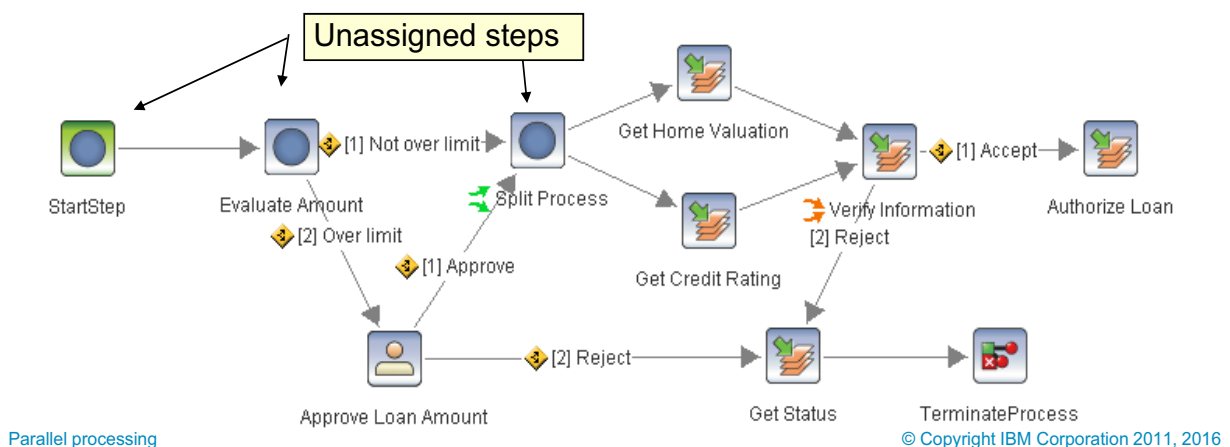


Figure 2-11. Use of an unassigned step

An unassigned step on the workflow map is a step that is not assigned to a work queue or a participant in the General properties pane.

An unassigned step can facilitate certain design cases:

- As a dummy step used to evaluate routing conditions to perform calculations, and perform assignments
- As an AND-split step in the case of parallel processing
- As a placeholder for cycling back to a previous step in a process flow

Processes that are performed in unassigned steps are not queued and cannot be viewed with administrative tools.

The example workflow map that is shown on this slide is for a loan processing workflow and contains three unassigned steps. The unassigned step, Evaluate Amount, is used to evaluate a routing condition that tests the loan amount and determines whether the amount is over limit or within limit. The routing condition is evaluated and is used to determine whether the next step is Split Process or Approve Loan Amount. The unassigned step, SplitProcess, is used as an AND-split step. The unassigned step on the left is a submap Start step.

Special cases

A Launch step is always unassigned because the workflow is not created until after the Launch step is completed.

A StartStep (the first step in a submap) is unassigned by default when you create the submap. You can assign the Start step to a participant or to a work queue.

Demonstration



- In Process Designer, on the Workflow map, define an AND-split step that is unassigned.
- Add two Activity steps that are intended to be run in parallel.
- Define a collector step.
- Validate and Transfer the workflow.
- Launch the workflow and process the Launch step.
- In Process Administrator, search for all the work items.
- Show the multiple child work items and the parent work item waiting in the Delay queue.
- Show that the F_WobNum is different for each work item and the F_WorkFlowNumber is the same.

If you are taking this course as a self-paced online course, return to the main course menu and click the link to play the recorded demonstration.

Parallel processing

© Copyright IBM Corporation 2011, 2016

Figure 2-12. Demonstration

Unit summary

- Create a parallel process in a workflow

Parallel processing

© Copyright IBM Corporation 2011, 2016

Figure 2-13. Unit summary

Exercise: Define a parallel process in a workflow

Parallel processing

© Copyright IBM Corporation 2011, 2016

Figure 2-14. Exercise: Define a parallel process in a workflow

Exercise introduction

- Add steps to define a parallel process
- Set a data field merge type
- Test the parallel processing



Parallel processing

© Copyright IBM Corporation 2011, 2016

Figure 2-15. Exercise introduction

Unit 3. Manage participation

Estimated time

00:40

Overview

This unit shows how to dynamically assign participants in a workflow at runtime and setup participant voting in a workflow.

How you will check your progress

- Successfully complete the lesson exercises.

References

IBM Knowledge Center

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8toc.doc/welcome_p8.htm

Why is this lesson important to you?

- You are designing a workflow application and you want to:
 - Control how users participate in a workflow.
 - Assign a user to track the progress of a workflow.
 - Design a workflow that uses dynamic assignment of participants at run time.
 - Use participant voting to determine the next step in the workflow.

Manage participation

© Copyright IBM Corporation 2016, 2011

Figure 3-1. Why is this lesson important to you?

Unit objectives

- Control how users participate in a workflow
- Define participant voting for a step

Manage participation

© Copyright IBM Corporation 2011, 2016

Figure 3-2. Unit objectives

Workflow participants

- A workflow participant is any user or group who is assigned to process steps in a workflow.
- When you assign a step to specific participant, the work assigned appears in the participant's Inbox.
 - Decision of who processes the work is made at **design time**.
- When you assign a step to a work queue, the work is processed by any one of the users or groups who have processing access rights to the queue.
 - Decision of who processes the work is made at **run time**.

Manage participation

© Copyright IBM Corporation 2016, 2011

Figure 3-3. Workflow participants

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Process applications concepts>Design and run workflows>Workflow participants

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.user.doc/bpfwd010.htm

Review:

A workflow participant is any user or group who is assigned to process steps in a workflow.

A workflow author, administrator, or tracker can:

- Assign any user or group as a participant for a particular step in a workflow. When the workflow is running, work assigned to a participant appears in that participant's Inbox.
- Assign a work queue to a step. Any users or groups that have processing access rights to the queue can process the step.

Workflow groups

- Placeholders for one or more users or groups
 - Assign group members at design time or dynamically at run time.
 - **Not** the same as an LDAP group
 - Can be the assigned participant for any step
- Use case and benefits
 - Provide flexibility to assign participants at design time or run time
 - Create a workflow group for a collection of users who perform a particular job function in a workflow
 - For example, supervisors or loan officers
- Define a workflow group in:
 - Workflow Properties > Workflow Groups tab.

Manage participation

© Copyright IBM Corporation 2016, 2011

Figure 3-4. Workflow groups

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Workflow properties - workflow groups

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh181.htm

Define workflow groups within a specific workflow definition, which are placeholders for one or more users or groups. The members of a workflow group typically perform a particular job function in a workflow; for example, claims adjustors or supervisors. A workflow group can be the assigned participant for any step.

For flexibility in defining a workflow, the author can either assign one or more specific participants to a workflow group, or allow the participants to be assigned later. The participants can be assigned as part of the launch process or at a particular step at runtime. This technique of including unspecified workflow groups in the workflow definition is useful when the participant for a step is likely to change each time the workflow runs.

For any step, the author can:

Allow workflow participants to see which users are assigned to workflow groups.

Allow workflow participants to assign different users to workflow groups.

A workflow group is defined within a particular workflow definition, which means that the same workflow group name can consist of different users in different workflow definitions.

Example use case

A payment processing workflow has a Verify Invoice step processed by an accounts payable clerk. However, the actual clerk can be different each time the workflow runs. The Accounts Payables department assigns its clerks to work with different vendors, so the clerk assigned to process an invoice depends on the vendor that sent the invoice. In the workflow definition, the workflow designer creates a workflow group named, `payment_clerks`, and assigns the group as the participant for the Verify Invoice step. When a workflow is launched, a user is assigned at run time to the `payment_clerks` workflow group, depending on the vendor that originated the invoice.

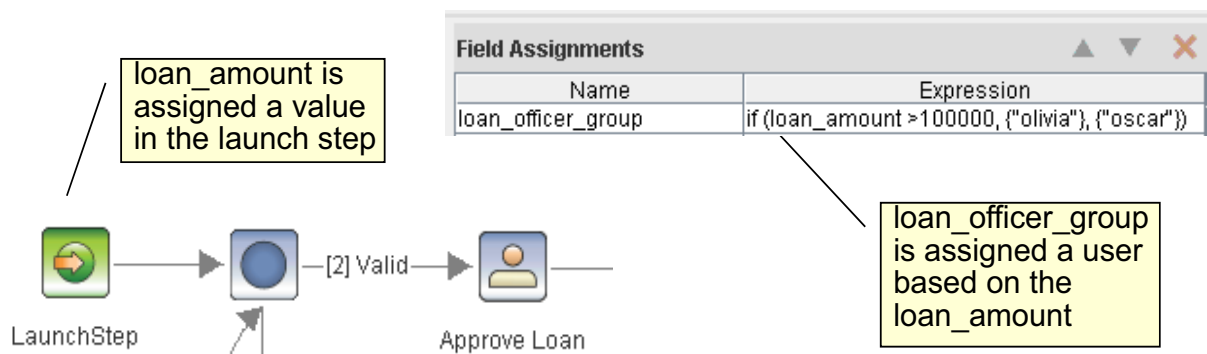


Important

If you change the name of a workflow group or you delete a workflow group, the change is updated in all steps where the workflow group is used as a parameter. However, the change is not reflected in expressions where the workflow group is used.

Example: Dynamic assignment of workflow groups

- Use case scenario
 - Loan processing workflow contains an Approve Loan step.
 - A specific loan officer is assigned to approve, based on the amount.
- Design solution example
 - The workflow group, loan_officer_group, is assigned as step destination for the Approve Loan Amount step.
 - Before execution of the step, a specific loan officer is assigned, based on the loan_amount data field value.



Manage participation

© Copyright IBM Corporation 2016, 2011

Figure 3-5. Example: Dynamic assignment of workflow groups

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Expression Builder

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh147.htm

This business use case is an example of dynamically assigning a workflow group based on the loan amount of a loan that is being processed. The loan processing workflow contains a workflow step called Approve Loan. At this step, a loan officer must approve a loan. A specific loan officer is assigned to approve the loan, based on the loan amount. If the loan value is more than 100,000, then Olivia must approve the loan. If the loan value is less than or equal to 100,000, then Oscar must approve the loan.

The diagrams on this page show the example workflow map with the Approve Loan step and its field assignment of the loan_officer_group before execution. If the loan_amount is greater than 100000, then Olivia is assigned to the loan_officer_group. Otherwise, Oscar is assigned to the loan_officer_group.

System-provided workflow group: F_Trackers

- Members are assigned in Workflow Properties > Workflow Groups.
- All assigned members are workflow trackers.
- Members can also be assigned at run time by using Process Administrator and the full access view of Process Tracker.
- Consider assigning at least one tracker for a workflow definition.
 - To monitor the workflow and help resolve runtime problems

Manage participation

© Copyright IBM Corporation 2016, 2011

Figure 3-6. System-provided workflow group: F_Trackers

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Assign Trackers

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh159.htm

All assigned members of the F_Trackers become workflow trackers, with access to the Tracker application. Although it is not required, for most processes it is useful to have one or more users or groups that are assigned as trackers to monitor events and to help resolve problems when the workflow is running.

To assign a tracker for a workflow definition, open the Workflow Properties window. In the Workflow Groups tab, select the predefined F_Trackers workflow group. On the right, click the pencil icon to modify the list of participants. Search for one or more participants and add them to the list.

System-provided workflow group: F_Originator

- Specifies the user who launched the workflow
- System automatically assigns value at launch time
- Can be used as the assigned participant for a step
- Not generally used for workflows launched by the system (for example, through a workflow subscription)
 - If you assign a step to F_Originator in a workflow that is launched by a workflow subscription, the work item is assigned to the Content Platform Engine administrative user.

Manage participation

© Copyright IBM Corporation 2016, 2011

Figure 3-7. System-provided workflow group: F_Originator

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Process applications concepts>Design and run workflows>Workflow participants>Selecting workflow participants (Scroll to the bottom of the help topic).

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.user.doc/bpfwd055.htm

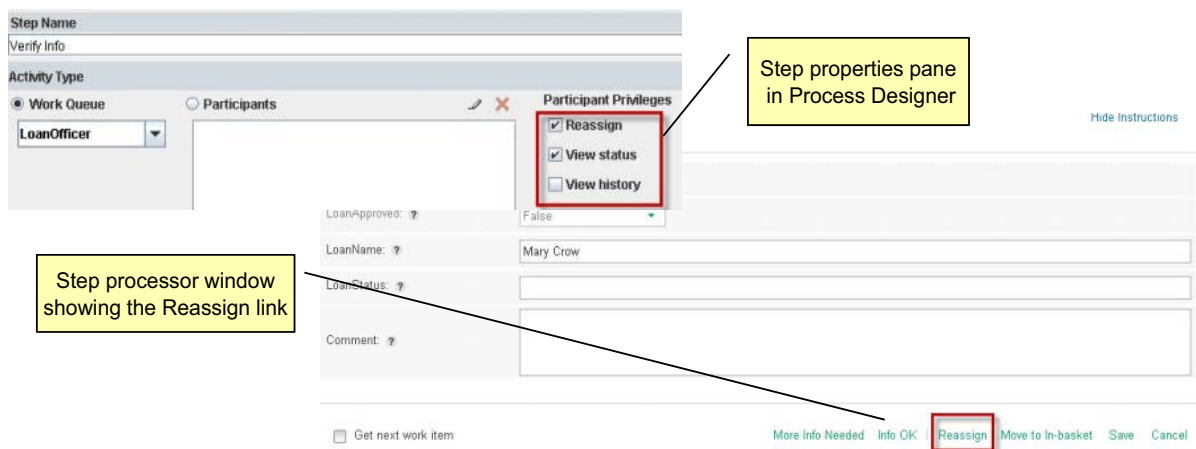
F_Originator is a special workflow group that represents the name of the user who launched the workflow. At launch time, the system assigns the name of the launch user to that workflow group. You can assign a step destination to the F_Originator workflow group. At run time, the system routes the work item to the user who launched the workflow.

- Do not use F_Originator for workflows that are launched in response to some event in the Content Platform Engine (for example, checking in a document of a particular document class). In this case, the system uses credentials from the Content Platform Engine server's JAAS context, which corresponds to a specific Content Platform Engine administrative user. Work items assigned to F_Originator go to that Content Platform Engine administrator's Inbox. This situation is not unrecoverable, but it is probably not the intended result.

- Do not use F_Originator in workflows that provide a Web Service with a Receive system step as the first step in the workflow. In this case, the workflow system launches the workflow, which has no associated user name or inbox, so work items assigned to F_Originator go to the Malfunction queue.

Set participant privileges

- In General step properties, you can control how a workflow participant interacts with the workflow at a step.
- For a particular step, you can allow a workflow participant to:
 - Reassign work to another user.
 - View status by using Process Tracker.
 - View history by using the History window in the step processor.



Manage participation

© Copyright IBM Corporation 2016, 2011

Figure 3-8. Set participant privileges

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About steps>About Activity steps>Activity step - general properties

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh066.htm

For a particular step, you can allow a workflow participant to:

- Reassign
 - Allow the participant to reassign work, at this step, to another user. At run time, a Reassign link appears at the bottom of the step processor.
- View status
 - Allow the participant to use Process Tracker to display the workflow map, show which participants processed previous steps, show responses, comments, milestone messages, and other information.
 - To view the Properties pane, the participant must be assigned as a tracker for the workflow.

- View history
 - Allow the participant to view the history of the running workflow. The History dialog box lists which participants processed previous steps, shows responses and comments that are entered by participants at previous steps, and provides other information. It does not display the workflow map.

The upper-left screen capture shows the General step properties pane, in Process Designer, with Reassign and View History checked.

The lower-right screen capture shows a step processor window, with the link, Reassign, included, since the check box is selected in the step properties.

Participant voting

- A multi-participant step with responses specified
 - All participants must complete the step before responses are evaluated.
 - You can base route conditions on the aggregated outcome of the participant responses.
- Use COUNT in the route conditions to test the number of participant responses.
 - Examples:
 - COUNT (Approve) > COUNT (Reject)
 - COUNT (Reject) >= 1

Manage participation

© Copyright IBM Corporation 2016, 2011

Figure 3-9. Participant voting

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About routing>Associate a response with a route

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh041.htm

You might need to specify a routing decision based on responses collected in a multi-participant step. A document approval process routes a document to three managers for approval. If two of the three managers approve the document, you want to publish the document.

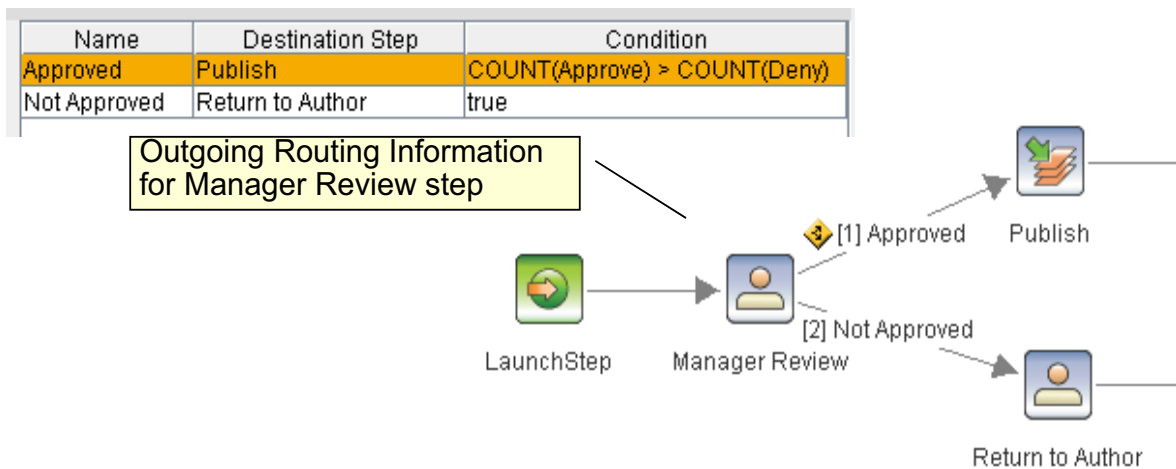
However, in a more general case you might not know the total number of managers that is asked to approve a document. Instead, you want to publish if the majority of managers that reviews a document responds with an approval.

To model the need for participant voting in a workflow, you can build route conditions that use COUNT to test the number of participants that select a particular response. For example, the route condition COUNT (Approve) > COUNT (Reject) evaluates to true if the number of participants that select the Approve response is greater than the number of participants that select the Reject response. If at least one participant selects the Reject response, the route condition COUNT (Reject) >= 1 evaluates to true.

In the conditional routing expression, ANY, ALL, and NONE are also available when constructing the route condition. For example: NONE (Approve)

Example: Counting responses

- Responses for the Manager Review step
 - Are specified in the order [1] Approve, [2] Deny
- Two outgoing routes from Manager Review step
 - If COUNT (Approve) > COUNT (Deny), [1] Approved route is taken
 - Otherwise, route [2] Not Approved is taken



Manage participation

© Copyright IBM Corporation 2016, 2011

Figure 3-10. Example: Counting responses

In this example, a multi-participant step called Manager Review in which managers review a document and select 1 of 2 responses: Approve or Deny. If a majority of the managers approve the document, the next step to be executed is the Publish step. Otherwise, the next step to be processed is the Return to Author step in which the document is returned to the author for revision.

To model this workflow logic, use the COUNT function to build the outgoing route conditions. The step routing properties for the two routes, Approved and Not Approved, are shown on this page. The step responses for the Manager Review step are specified in the following order: Approve and Deny. If the number of managers that select the Approve response is greater than the number of managers that select the Deny response (that is, Count(Approve) > COUNT(Deny)), then the Approved route is taken to the Publish step. Otherwise, the Not Approved route is taken to the Return to Author step.

The upper screen capture, shows the Outgoing Routing Information for the Manager Review step. The lower diagram shows an example workflow map with the two outgoing routes.

F_Responses system field

You can also use the F_Responses system field in a routing condition to test for participant responses at a step. F_Responses is a system field that is an integer array. The array lists each response in a step with a count of the number of participants that choose the response. Responses are positional, based on the order in which you defined the routing responses for the step.

For example, the expression `F_Responses[1] > F_Responses[2]` can be used to test whether the number of users who selected the first response is greater than the number of users who selected the second response.

Demonstration



- In a prepared workflow definition file, open the Workflow Properties window.
- Assign a user to track the progress of the workflow.
- Define a new workflow group.
- Select a participant step on the workflow map.
- Assign a user to the workflow group by using a dynamic assignment before execution of the participant step.

If you are taking this course as a self-paced online course, return to the main course menu and click the link to play the recorded demonstration.

Manage participation

Figure 3-11. Demonstration

© Copyright IBM Corporation 2011, 2016

Unit summary

- Control how users participate in a workflow
- Define participant voting for a step

Manage participation

© Copyright IBM Corporation 2011, 2016

Figure 3-12. Unit summary

Exercise: Control how users participate in a workflow

Manage participation

© Copyright IBM Corporation 2011, 2016

Figure 3-13. Exercise: Control how users participate in a workflow

Exercise introduction

- Assign a Tracker to a workflow
- Add a dynamically assigned workflow group
- Define participant voting with multi-participant steps



Manage participation

© Copyright IBM Corporation 2011, 2016

Figure 3-14. Exercise introduction

Unit 4. Work with submaps

Estimated time

00:30

Overview

This unit shows how to efficiently organize steps into submaps in a workflow definition and how to define post-processing steps that are always processed immediately before workflow termination.

How you will check your progress

- Successfully complete the lesson exercises.

References

IBM Knowledge Center

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8toc.doc/welcome_p8.htm

Why is this lesson important to you?

- You are designing a workflow application. You need an efficient way of organizing steps in a workflow definition. You want to define a set of post-processing steps, which always run immediately before workflow termination.

Work with submaps

© Copyright IBM Corporation 2016, 2011

Figure 4-1. Why is this lesson important to you?

Unit objectives

- Use a submap in a workflow
- Identify system default submaps

Work with submaps

© Copyright IBM Corporation 2011, 2016

Figure 4-2. Unit objectives

Using submap steps

- Submap step
 - Single step that represents a call to another map in the same workflow definition.
 - Control returns to the calling step when the steps in the submap are completed.
 - Can include on any map in a workflow definition.
 - Advantage: Makes workflow definition more modular
- Submap
 - Workflow map that is called from another map
 - Begins with a StartStep that is automatically included.
- Map toolbar
 - Used to list, display, create, and delete maps.

Work with submaps

© Copyright IBM Corporation 2016, 2011

Figure 4-3. Using submap steps

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Process applications concepts>Design and run workflows>About workflow maps

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.user.doc/bpfwd019.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About steps>About Submap steps

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh005.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Display a map

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh149.htm

Why use workflow submaps?

For modeling a more complex business process, you can find it more efficient to use submaps in a workflow definition. Submaps make a workflow definition more modular. A submap can define a single process that is run multiple times, rather than defining the process in multiple places. The submap is called when needed. In addition to reducing redundancy, submaps keep the workflow

map area uncluttered in the same way that folders are used to organize files in a computer file system.

Workflow submaps

Each workflow definition has a main map, called Workflow (Main Map), which runs when the workflow is launched. You can choose to place all of the workflow steps and routes on this one map. However, for a more complex business process, it is better to place the steps for the primary process on the main map and use submaps for secondary processes.

A submap is a workflow map that is called from another map in the same workflow definition. You use a submap step on a map to call a submap. Map recursion is allowed. Think of a submap as a subroutine in the context of a computer program.

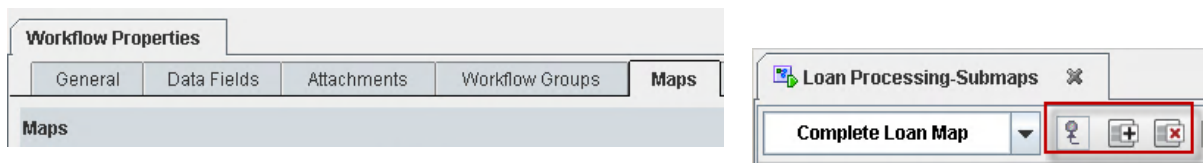
You create new submaps, and they appear in the list of maps on the map toolbar. Tools on the map toolbar are available to view, create, delete, and list maps. Each submap always begins with a StartStep.

Two other ways to display a submap:

1. Double-click the submap step on the calling map.
2. Click Actions > Go to Map, and select the map from the list.

Define a new submap

- Click Create Map on the map toolbar.
 - Assign a unique name to the map.
 - Add steps and routes to the submap.
 - Submap begins with a StartStep, which you can edit.
- Use a submap by assigning it to a submap step.
 - In submap step General properties, select the submap from the list.
- Workflow Properties > Maps displays a list of all maps in the current workflow definition.
 - Map Usage pane shows where maps are used in the workflow.
 - You can create and delete maps.



Work with submaps

© Copyright IBM Corporation 2016, 2011

Figure 4-4. Define a new submap

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Create a map

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh150.htm

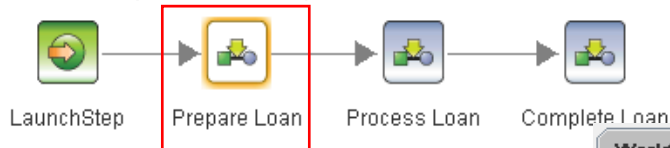
FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Map usage

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh157.htm

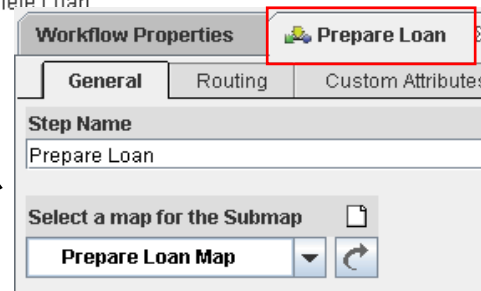
After creating a new submap, you use it by assigning the submap to a submap step. You place a submap step on the calling map. In the submap step properties, you select the map to call from the list. You can assign other properties to the submap step, as needed.

Example: Using a submap step

- Workflow (main map) contains a Prepare Loan submap step.



- The Prepare Loan step properties specify the map called.



- When the Prepare Loan step runs, the Prepare Loan Map submap is called and its steps are run.



Work with submaps

© Copyright IBM Corporation 2016, 2011

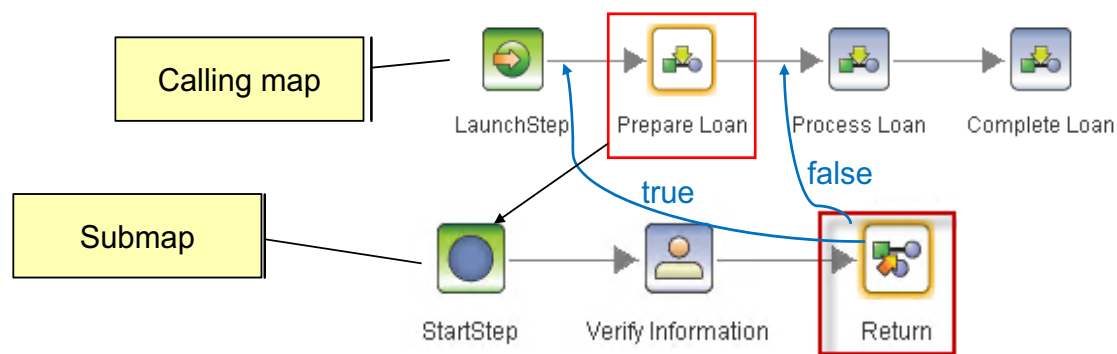
Figure 4-5. Example: Using a submap step

The diagrams on this page show a Workflow (main map) with a submap step, Prepare Loan. The submap step, Prepare Loan, is assigned to the Prepare Loan Map submap. At run time, when the submap step, Prepare Loan, is encountered, the steps in the submap, Prepare Loan Map, are run. When the submap, Prepare Loan Map, completes, control is returned to the calling map, Workflow (main map), and the next step in the Workflow map is processed (in this case, the submap step, Process Loan).

The diagrams on this page show a Workflow (main map) with a submap step, Prepare Loan. The submap step, Prepare Loan, is assigned to the Prepare Loan Map submap. At run time, when the submap step, Prepare Loan, is encountered, the steps in the submap, Prepare Loan Map, are executed. When the submap, Prepare Loan Map, completes, control is returned to the calling map, Workflow (main map), and the next step in the Workflow map is processed (in this case, the submap step, Process Loan).

Implicit and explicit return from a called map

- Every submap has a return to the calling map.
 - Implicit (automatic) return at the end.
 - By default, processing continues in the calling map at the stage immediately after the submap was called.
- Add explicit Return system step to change this behavior.
 - Return system step:
 - Allows control of stage to return in the calling map.
 - Useful for exception handling.



Work with submaps

© Copyright IBM Corporation 2016, 2011

Figure 4-6. Implicit and explicit return from a called map

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About steps>About Submap steps>About return from a called workflow map

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh003.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About steps>System functions>General step activity>Return system function

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh111.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Process applications concepts>Design and run workflows>About workflow maps>About step states

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.user.doc/bpfwd014.htm

Every submap has an automatic implicit return to the calling map. An implicit return at the end of the submap (no Return system function), returns to the calling map immediately after the step that called the submap. You can change this default behavior by adding a Return system function to your submap. With the Return system function, you can immediately exit from a called submap and return to the calling map. You can specify a Return expression (a valid Boolean expression). If the expression evaluates to true, you return to the calling map and repeat the action that called the

submap. If the expression evaluates to false, you return to the calling map, skip the action that called the submap, and resume processing at the next step.

The diagram shows two maps, a calling map and a submap. The calling map gets to the Process Loan submap step, which calls the submap. The submap starts processing, when the Return system function is reached, the Boolean expression that is configured is evaluated. If the result of the Boolean expression is true, the submap returns to the point just before the Process Loan map was called and repeats the step. If the result of the Boolean expression is false, the submap returns to the point just after the calling submap and processes the next step in the calling map.

Default system maps

- System-provided maps are in every workflow definition:
 - Workflow (main map)
 - Terminate
 - Malfunction
- Each workflow definition has its own copy of these maps.
- Workflow system runs these maps as needed.
- Workflow (main map).
 - It is always the main map.
 - Execution of a workflow always starts with its launch step.
- Terminate map
 - Called automatically at the end of the Workflow (main map).
- Malfunction map
 - Runs when an error occurs during workflow processing.

Work with submaps

© Copyright IBM Corporation 2016, 2011

Figure 4-7. Default system maps

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Process applications concepts>Design and run workflows>About workflow maps>About system maps

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.user.doc/bpfwd015.htm

In addition to workflow maps defined by the process designer, system maps exist that the workflow system runs automatically under certain conditions.

All workflow definitions include these FileNet P8-supplied system maps:

- Workflow
- Terminate
- Malfunction

The workflow system runs the system maps as needed during standard or exception processing. Each workflow definition has its own copies of the system maps.

The Workflow system map

The main map in each workflow definition. When you create or open a workflow definition, Process Designer automatically starts with the Workflow map displayed. When a workflow is launched, processing begins with the execution of the LaunchStep on the Workflow system map.

The Terminate map

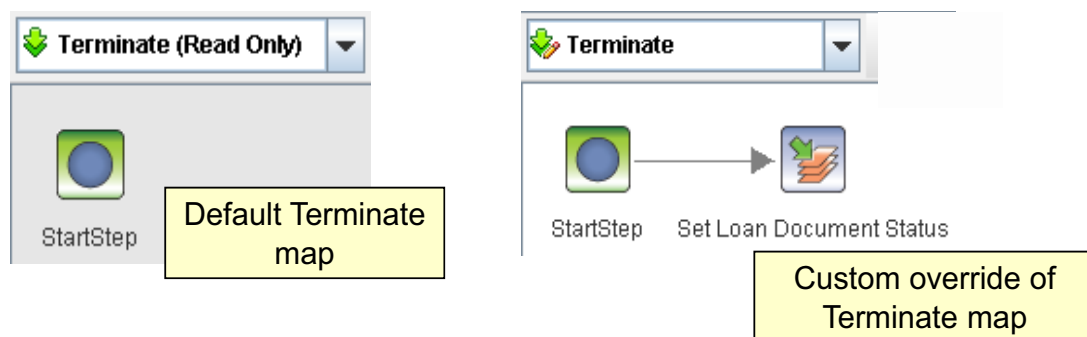
The Terminate map is called automatically at the end of the Workflow (main map). By default the Terminate has no steps; it removes the work item from the system.

The Malfunction system maps

The Malfunction map runs when an error occurs during workflow processing.

Workflow termination

- By default, the Terminate system map is run after a workflow completes processing.
- The default Terminate map
 - Contains only a submap StartStep
 - Therefore, the workflow is terminated immediately.
- You can override the default Terminate map behavior.
 - Add post-processing steps to the Terminate map.
 - Workflow terminates after the custom Terminate map completes.



Work with submaps

© Copyright IBM Corporation 2016, 2011

Figure 4-8. Workflow termination

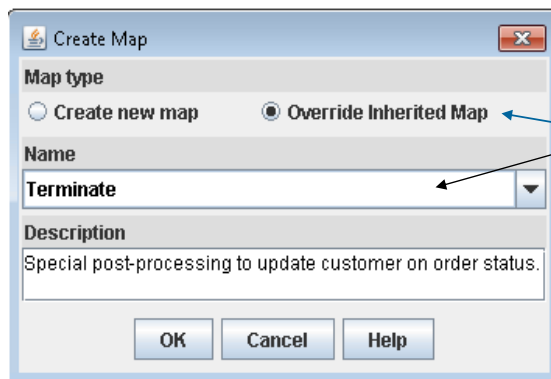
The Terminate system map runs when a parent work item terminates. When a child work item reaches the end of its Workflow map, the work item is automatically removed from the system, but the Terminate system map is not called.

You can manually cause the Terminate map to run by using the TerminateProcess system function in a workflow definition or by terminating a work item in Process Administrator.

The screen capture on the left shows the default Terminate map. The screen capture on the right, shows the default Terminate map overridden. The step, Set Loan Document Status is added. At run time, the Set Loan Document Status step is run before the workflow terminates.

Override a system map

- The default Terminate and Malfunction system maps are displayed as Read Only in Process Designer.
- Override the default maps by using the Create Map tool.
- Use care when overriding system maps.
 - If you redefine the map, be aware that the conditions under which the map is run remain the same.



To override a map, select "Override Inherited Map" and then select the map.

Work with submaps

© Copyright IBM Corporation 2016, 2011

Figure 4-9. Override a system map



Important

Use care when overriding the system-provided default maps. The conditions under which a default system map is run remain the same. You must design the overridden maps to handle any unexpected problems in their execution. For example, if you modify the Terminate system map to send work items to a work queue rather than removing them, you delete the primary mechanism for removing completed work items from the system.

The screen capture on this page shows an example of the Create Map tool, the Override Inherited Map option is checked, and the Terminate map is selected.

Demonstration



- Create a submap.
- Add a submap step to a map.
- Assign the submap step properties to call the new map.

If you are taking this course as a self-paced online course, return to the main course menu and click the link to play the recorded demonstration.

Work with submaps

© Copyright IBM Corporation 2011, 2016

Figure 4-10. Demonstration

Unit summary

- Use a submap in a workflow
- Identify system default submaps

Work with submaps

© Copyright IBM Corporation 2011, 2016

Figure 4-11. Unit summary

Review questions



1. Which of the system maps listed, contains the LaunchStep?
 - A. Workflow system map.
 - B. Terminate system map.
 - C. Exception system map
 - D. Start system map
2. Which system map is called when an error occurs?
 - A. Start system map.
 - B. Terminate system map.
 - C. Malfunction system map
 - D. Exception system map
3. Which system map is called when a child workflow terminates?
 - A. Workflow map.
 - B. Terminate map.
 - C. Exception map
 - D. None of the above

Work with submaps

© Copyright IBM Corporation 2011, 2016

Figure 4-12. Review questions

Write your answers here:

- 1.
- 2.
- 3.

Review questions



4. Which map cannot be overridden?
- A. Workflow map
 - B. Terminate map
 - C. Start map
 - D. Exception map

Work with submaps

© Copyright IBM Corporation 2011, 2016

Figure 4-13. Review questions

Write your answers here:

4.

Review answers



1. Which of the system maps listed, contains the LaunchStep?

- A. [Workflow system map.](#)
- B. Terminate system map.
- C. Exception system map
- D. Start system map

The answer is A.

2. Which system map is called when an error occurs?

- A. Start system map.
- B. Terminate system map
- C. Malfunction system map
- D. [Exception system map](#)

The answer is D.

Review answers



3. Which system map is called when a child workflow terminates?

- A. Workflow map.
- B. [Terminate map.](#)
- C. Exception map
- D. None of the above

The answer is [B](#).

4. Which map cannot be overridden?

- A. Workflow map
- B. Terminate map
- C. [Start map](#)
- D. Exception map

The answer is [C](#).

Exercise: Use workflow submaps

Work with submaps

© Copyright IBM Corporation 2011, 2016

Figure 4-16. Exercise: Use workflow submaps

Exercise introduction

- Create submaps
- Use submaps in a workflow
- Override the Terminate map



Work with submaps

© Copyright IBM Corporation 2011, 2016

Figure 4-17. Exercise introduction

Unit 5. Handle workflow exceptions

Estimated time

00:30

Overview

This unit shows how to handle the resolution of possible business process exceptions in a workflow definition.

How you will check your progress

- Successfully complete the lesson exercises.

References

Knowledge Center

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8toc.doc/welcome_p8.htm

Why is this lesson important to you?

- You are designing a workflow application. You must handle the resolution of possible business process exceptions in your workflow definition. You need to override the Malfunction system map.

Handle workflow exceptions

© Copyright IBM Corporation 2016, 2011

Figure 5-1. Why is this lesson important to you?

Unit objectives

- Resolve business process exceptions

Handle workflow exceptions

Figure 5-2. Unit objectives

© Copyright IBM Corporation 2011, 2016

Business process exceptions

- When a problem occurs and processing cannot continue in a running workflow, a workflow exception occurs.
 - Work item is sent to the Malfunction system map.
 - By default, the work item is sent to the Conductor queue.
- You must design the workflow to handle business processing exceptions.
 - Avoid errors by using a combination of specialized error handling submaps and custom step processors.
 - Work together with the application developer to identify potential exception situations and create custom solutions.

Handle workflow exceptions

© Copyright IBM Corporation 2016, 2011

Figure 5-3. Business process exceptions

When a problem occurs and processing cannot continue, a workflow exception occurs. These errors are business processing errors and user errors, not system integrity errors. For example, if an empty workflow group is assigned as the participant for a step, a workflow exception occurs when processing reaches that step. Workflow exceptions are caused by events, such as division by zero, invalid date/time mask, or an invalid participant in a workflow group. When an exception occurs, the Malfunction submap runs and the work item by default goes into the Conductor queue.

Several situations can result in a Malfunction, which causes the work item to be placed in the Conductor queue. In addition, a software application developer can define a custom step processor or other application to explicitly call the Malfunction system map when specific errors occur. The default Malfunction map moves the work item to the Conductor system queue with a Review step.

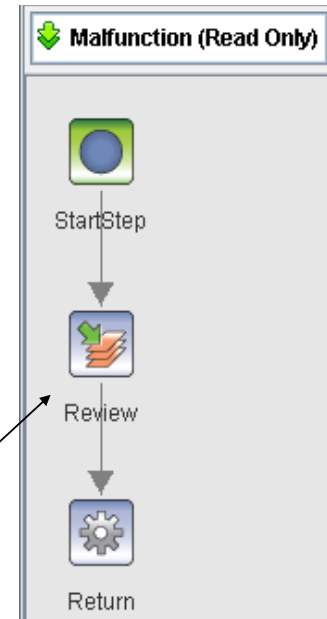
Generally, you must make a provision to handle business processing exceptions in a workflow. You can choose to override the default behavior in the Malfunction submap. For example, the work item can be sent to an exception handling queue where a specialized step processor handles the error condition or sent to a user who corrects the error.

Typically in a production workflow environment, work items are not sent to the Conductor queue if the workflow design accounts for these exceptional business situations. The workflow author generally works together with the application developer to identify potential exception cases and create custom solutions.

Malfunction map and the Conductor queue

- Conductor queue
 - A queue for work items in an exception state
 - Default queue where a work item goes when a process exception occurs
- Items require administrative action.
 - System cannot process work items in the Conductor queue without intervention.
 - Use Process Administrator to access Conductor queue and view information about exceptions.

Conductor Review step is in the default Malfunction map, unless overridden.



Handle workflow exceptions

© Copyright IBM Corporation 2016, 2011

Figure 5-4. Malfunction map and the Conductor queue

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Process applications concepts>Design and run workflows>About workflow maps>About system maps

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.user.doc/bpfwd015.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Process applications concepts>Administration and configuration>Troubleshooting

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.user.doc/bpfcg014.htm

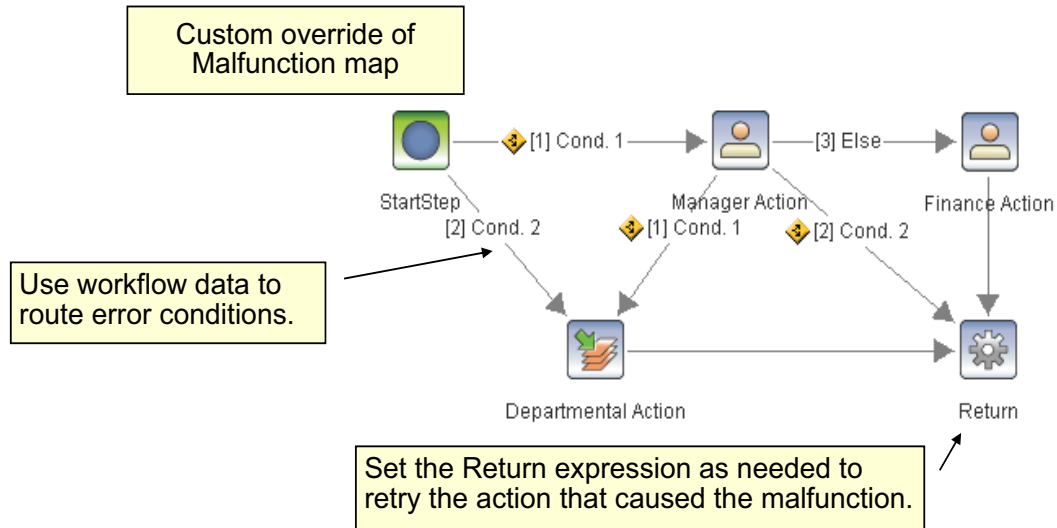
When a workflow exception occurs in a running workflow, the Malfunction map moves the work item to the Conductor system queue by default. The Conductor queue is a system-provided and system-maintained work queue that collects work items in an exception state. Work items remain in the Conductor queue indefinitely until an administrative user intervenes.

In Process Tracker and Process Administrator, you can examine a running workflow and determine the location and nature of the error. You can view the work item information stack in Process Administrator to display any messages associated with a work item in an exception state.

The diagram on the right shows an example of the default Malfunction system map, which contains StartStep, Conductor Review, and Return steps.

Override the Malfunction system map

- You can override the default Malfunction map to handle workflow exceptions.
 - Work item returns to referring step after Malfunction submap completes.
 - If an exception occurs within overridden Malfunction submap, the work item goes to the Conductor queue.



Handle workflow exceptions

© Copyright IBM Corporation 2016, 2011

Figure 5-5. Override the Malfunction system map

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Create a map

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh150.htm

You can override the Malfunction system map by defining an overriding submap in the Create Map window. You can add your own custom steps to the map, which are run when the Malfunction map is called.

If the work item completes the custom Malfunction map without encountering a `TerminateBranch` or `TerminateProcess` step, the custom Malfunction map returns the work item to the calling step.

If you use a `Return` system function in the Malfunction map, you set the `Return` expression as needed to retry the action that caused the malfunction or not. Exceptions within an overridden Malfunction map go to the Conductor queue.

While you cannot alter the circumstances under which the Workflow system runs the Malfunction system map, you can create your own exception-handling workflow maps and call them as needed in your business process. For example, in the code for a custom step processor, a software developer can include a call to run a user-defined map if a certain error occurs when the step processor is running.

The diagram on this page shows an example of a custom override of the default Malfunction system map. It contains steps for actions to take by a manager, finance, and department personnel if a workflow malfunction occurs.

Using Return in a Malfunction map

- Return system function in a called submap
 - Return system function has a Boolean Return expression (also called the retry option).
 - If the Return expression is true, return to the calling map and repeat the action that called the submap.
 - If the Return expression is false, return to the calling map, skip the action that called the submap, and resume at the next step.
- The Return retry option is useful for exception handling.
 - You can override the default Malfunction map to correct the error and return to the calling map to retry the step that failed.

Handle workflow exceptions

© Copyright IBM Corporation 2016, 2011

Figure 5-6. Using Return in a Malfunction map

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About steps>About Submap steps>About return from a called workflow map

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh003.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>About steps>System functions>General step activity>Return system function

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh111.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Process applications concepts>Design and run workflows>About workflow maps>About step states

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.user.doc/bpfwd014.htm

The Return system function is generally used to return from a submap that overrides the Malfunction map.


The Return system function has a Boolean Return expression. The value of the expression affects how processing continues on return to the calling map.

- If the expression evaluates to true, then processing returns to the calling map and repeats the action that caused the exception.

- If the expression evaluates to false, then processing returns to the calling map, skips the action that caused the exception, and resumes processing at the next step.

When you are designing exception handling maps, the Return expression in the Return system function is useful. You can override the default Malfunction map to correct a business processing error and return to the calling map to retry the step that failed.

Workflow troubleshooting and recovery

- Search for work items in the Conductor queue or the custom queue.
 - Process Administrator search results show work items in an exception state.
 - Work items are flagged with the exception symbol. 
- View information about the exception.
 - Use Process Administrator to view the information stack and workflow fields values to determine the cause of the error.
 - Use Process Tracker to review the workflow history and workflow maps.
- Design and implement a solution that eliminates the cause.
- Useful system data fields:
 - F_LastErrorNumber
 - F_LastErrorText

Handle workflow exceptions

© Copyright IBM Corporation 2016, 2011

Figure 5-7. Workflow troubleshooting and recovery

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Process applications concepts>Administration and configuration>Troubleshooting

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.user.doc/bpfcg014.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Process applications concepts>Administration and configuration>Troubleshooting>Displaying workflow exceptions

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.user.doc/bpfcg015.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Process applications concepts>Design and run workflows>System fields definitions

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.user.doc/bpfwd039.htm

A workflow administrator or a workflow author, troubleshooting a workflow, can use Process Administrator, and Process Tracker to examine a running workflow and determine the location and nature of the error.

Display the information stack

You view the information stack to display any messages that are associated with a workflow exception.

- In Process Administrator, build and run a search for work items in the Conductor queue with Edit mode selected. If your application uses a custom error-handling queue as defined in an overridden Malfunction submap, you run a search of that custom queue.
- In the returned list of work items, select the item of interest. Right-click the item and click View Information Stack. Or, from the menu bar, click View > Information Stack.
- In the Information Stack window, click the work item to display the information stack for that item. The information stack lists the workflow map and step where the error occurred and displays a message that describes the error.

Troubleshooting and recovering running workflows

In some cases, a running workflow that generates an exception can be corrected and dispatched to complete its processing. In a production environment, the workflow administrator uses Process Administrator to take several types of corrective actions to recover a running workflow. Possible corrective actions include:

- Delete or terminate workflows and work items.
- Modify workflow properties, such as data field values and workflow groups.
- Open work in Process Tracker or step processors to modify attachments.
- Complete the work.
- Reassign the work.

When a workflow author is troubleshooting a workflow definition, the goal is to identify the **cause** of the workflow exception and adjust the workflow definition design so that the exception is avoided in future running workflows. Designing workflow applications might involve coordination with application developers and creation of special custom step processors to handle business process exceptions.

System fields that contain error information

Certain system fields in a work item might be helpful in troubleshooting a workflow. The F_LastErrorNumber field contains the error number that is associated with the last exception that occurred. The F_LastErrorText field contains the text that is associated with the last exception that occurred. View these system field values for a work item in Process Administrator.

Example: Workflow exception

- Malfunction occurs and work item goes to Conductor queue
 - View the information stack in Process Administrator.
 - Determine cause by reviewing the workflow map and data field values by using Process Administrator and Process Tracker.
 - Example: No user is assigned to the step destination workflow group.

Information Stack:

Map	Step	Error Number	Error Message
Malfunction	Review Exception	0x0	
PrepareLoan	Confirm Information	0xd5240069	Malfunction: [Err=d5240069] Step: Invalid Participant. []

- Possible design approaches
 - Use dynamic assignment of workflow groups and check for null or invalid users before the step that uses the workflow group.
 - Override Malfunction map and send work items to a workflow administrator who can assign a valid user.

Handle workflow exceptions

© Copyright IBM Corporation 2016, 2011

Figure 5-8. Example: Workflow exception

This slide shows an example information stack from Process Administrator for a work item that was sent to the Malfunction map. The information stack shows that the work item was in the Confirm Information step on the Prepare Loan map when the error occurred. The step had an invalid participant.

By reviewing the work item history and data field values in Process Administrator and Process Tracker, you can determine the cause of the invalid participant. For example, the workflow group assigned to the step destination was empty. Therefore, the system did not find a valid participant to process the work item.

Possible ways to handle this exception are:

- Check for null or invalid users before running a step.
- Override the Malfunction map with a submap that sends work items, with invalid users, to a workflow administrator who can assign a valid user.

Demonstration



Troubleshoot a workflow exception

- Launch a workflow that generates a workflow exception.
- Open Process Administrator and search the roster for the malfunctioning work item.
 - Show the work item in the Conductor queue.
 - View it with Process Tracker
- Obtain information about the workflow exception.
- Correct the problem in Process Administrator.
- Complete the work item to continue processing the workflow.

If you are taking this course as a self-paced online course, return to the main course menu and click the link to play the recorded demonstration.

Handle workflow exceptions

Figure 5-9. Demonstration

© Copyright IBM Corporation 2011, 2016

Demonstration(2)



Handle workflow exceptions in a workflow

- Open a workflow definition in Process Designer.
- Override the Malfunction Map.
- Modify the Review step.
- Modify the default behavior of the Return step.
- Capture the error message and error number before execution of the StartStep.

If you are taking this course as a self-paced online course, return to the main course menu and click the link to play the pre-recorded demonstration.

Handle workflow exceptions

© Copyright IBM Corporation 2011, 2016

Figure 5-10. Demonstration(2)

Unit summary

- Resolve business process exceptions

Handle workflow exceptions

Figure 5-11. Unit summary

© Copyright IBM Corporation 2011, 2016

Exercise: Resolve business process exceptions

Handle workflow exceptions

© Copyright IBM Corporation 2011, 2016

Figure 5-12. Exercise: Resolve business process exceptions

Exercise introduction

- Determine the cause of a workflow malfunction
- Override the Malfunction map
- Test business process exception handling



Handle workflow exceptions

© Copyright IBM Corporation 2011, 2016

Figure 5-13. Exercise introduction

Unit 6. Workflow inheritance

Estimated time

00:30

Overview

This unit shows how to take advantage of process reusability and use inheritance in a workflow definition.

How you will check your progress

- Successfully complete the lesson exercises.

References

IBM Knowledge Center

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8toc.doc/welcome_p8.htm

Why is this lesson important to you?

- You are designing a workflow application. You want to take advantage of process reusability and use inheritance in a workflow definition. You need to create a workflow hierarchy and override an inherited map.

Workflow inheritance

© Copyright IBM Corporation 2016, 2011

Figure 6-1. Why is this lesson important to you?

Unit objectives

- Create a workflow hierarchy
- Override an inherited map

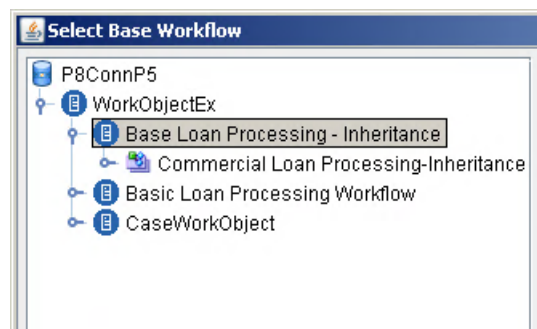
Workflow inheritance

© Copyright IBM Corporation 2011, 2016

Figure 6-2. Unit objectives

Using workflow inheritance

- You can create a workflow definition that is derived from and reuses elements from a base workflow.
 - Define common characteristics at a higher level in the class hierarchy.
 - Properties are automatically passed to derived workflows.
- Advantages of creating a hierarchy of workflow definitions:
 - Consistent processing across related workflows
 - More modular implementation with reusability
 - Expedite creation of workflow definitions across a group of related processes
- Example: loan origination in financial services company



Workflow inheritance

© Copyright IBM Corporation 2016, 2011

Figure 6-3. Using workflow inheritance

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Process applications concepts>Design and run workflows>Workflow inheritance

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.user.doc/bpfwd003.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Process applications concepts>Design and run workflows>Workflow inheritance>Inheritance example

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.user.doc/bpfwd033.htm

You can create workflow definitions that inherit certain elements from a previously defined workflow definition, including submaps. In other words, you can create a hierarchy of workflow definitions in which common characteristics are defined in a workflow definition at a high level in the class hierarchy. Characteristics are automatically passed to subsequently derived workflow definitions and reused at the lower levels in the class hierarchy.

Advantages of using workflow inheritance

If you use a hierarchy of workflow definitions, you can:

- Ensure consistent processing across related workflows.
- Establish a more modular implementation of your business process application.

- Take advantage of reusability of workflow elements.
- Take advantage of ease of maintenance with a single location for inherited submaps.
- Maintain better overall control of workflow definition files and changes to them.

Example of workflow inheritance in financial services industry

A financial services company offers loans to both commercial and residential customers. The organization has business units that handle these two lines of business. Among other operational goals, the company wants to streamline and standardize certain loan origination procedures across the two business units. The company implements an IBM FileNet workflow application for loan origination.

In its workflow implementation, the company creates the following workflow definition hierarchy:

- Base Loan Workflow contains common elements to all loans, such as borrower name, loan amount, loan ID, and maps such as Documentation Review, Underwriter Assignment, Credit Reporting, and Document Archiving.
 - Commercial Loan Workflow is inherited from Base Loan Workflow and contains more data fields and maps, such as business name and tax number, and maps such as Commercial Underwriting. It overrides the Credit Reporting submap to add steps for business credit reporting.
 - Mortgage Loan Workflow is inherited from Base Loan Workflow and contains more workflow elements, such as individual taxpayer number, property address, Residential Underwriting, Property Appraisal, and Valuation.

Inherited elements

- When you create a new workflow definition based on another workflow definition, the new workflow inherits characteristics from the base workflow, such as:
 - Workflow main map
 - Submaps
 - Data fields, attachments, workflow groups
 - Deadlines and reminders
 - Milestones
 - Event log and workflow roster
- These elements are designated Read Only in the new, derived workflow.
 - Unless you override the inherited item by redefining it

Workflow inheritance

© Copyright IBM Corporation 2016, 2011

Figure 6-4. Inherited elements

A new workflow inherits the following elements from its base workflow.

- Workflow main map.
- Submaps
- Data field, attachment, and workflow group definitions. You cannot delete these inherited items, but you can change the descriptions and initial values.
- Deadlines and reminders
- Milestones
- Event log and workflow roster, which can be changed and overridden until the new workflow is transferred.
- Condition identifier.
- Partner links and XML schema (cannot be overridden, Read Only).
- Rule sets (cannot be overridden, Read Only)
- Email notification preference

Inherited items in a derived workflow definition are designated as read-only. However, you can override an inherited item by redefining it. For example, to override an inherited map, you create a

new map of the same name in the derived workflow. If you later delete the overriding map, the inherited map is reactivated.

Workflow main map

The Workflow main map is a special case. In a new workflow definition, the Workflow main map is overridden by a blank map with only a launch step. To reactivate the inherited Workflow map, you must delete the main map in the current, derived workflow.

The base workflow

- To use inheritance, you specify a base workflow from which to inherit maps and properties.
 - Multiple levels of workflow hierarchy are possible
- Before you can select a base workflow, it must already be transferred into the isolated region.
- If you modify the base workflow definition
 - You must retransfer both the base workflow and the derived workflow to implement changes.
- The default base workflow is called WorkObjectEx
 - Contains system data fields and empty system maps
 - All workflows are derived from WorkObjectEX
 - Special case: workflows for IBM Case Manager solutions

Workflow inheritance

© Copyright IBM Corporation 2016, 2011

Figure 6-5. The base workflow

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Select base workflow

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh046.htm

To create a workflow definition that inherits submaps and other items that are defined in another workflow definition, you select a previously defined workflow definition as the base workflow.

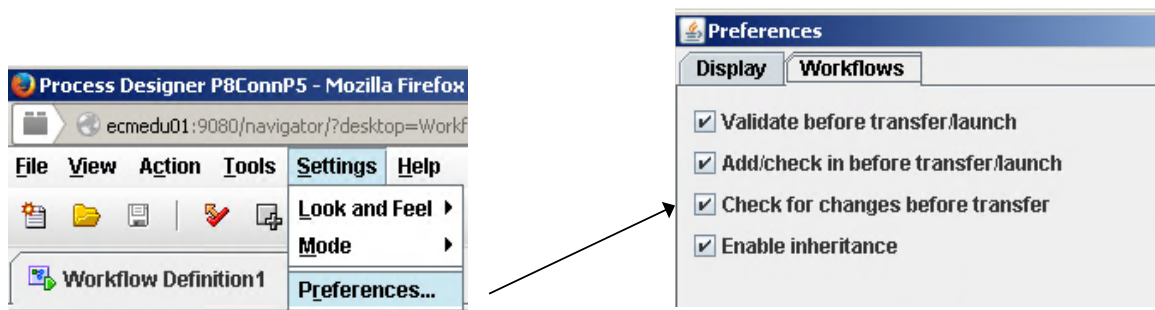
After selecting the base workflow, all of its submaps, data fields, attachments, workflow groups, and other elements are made available in the current workflow definition. Any inherited submap is designated as Read Only until you choose to override it. The default base class for all workflow definitions that are built in Process Designer is the system-provided WorkObjectEx. From WorkObjectEx, workflow definitions inherit system data fields, the Terminate submap, and the Malfunction submap. WorkObjectEx cannot be modified.

Default base workflow for IBM Case Manager solutions

For IBM Case Manager solutions, the default base workflow is called CaseWorkObject, which is derived from WorkObjectEx. The CaseWorkObject and WPSCaseWorkObject work classes are used only in workflows built with IBM Case Manager Builder. Information about IBM Case Manager solutions is beyond the scope of this course.

Workflow Preferences for inheritance

- To use inheritance in Process Designer, you must select the Enable Inheritance preferences option (enabled by default).
 - Use Settings > Preferences > Workflows tab
 - If the option is selected
 - Workflow inheritance functionality is fully available.
 - You can select a previously transferred base workflow from which to inherit.
 - If the option is cleared
 - Workflow inheritance functionality is disabled in Process Designer.
 - WorkObjectEx is the base workflow.



Workflow inheritance

© Copyright IBM Corporation 2016, 2011

Figure 6-6. Workflow Preferences for inheritance

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Validate, transfer, and launch a workflow definition or workflow collection>Preferences – workflow

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh035.htm

You use the Enable inheritance option in the Preferences > Workflows tab to show or hide the workflow inheritance features in Process Designer.

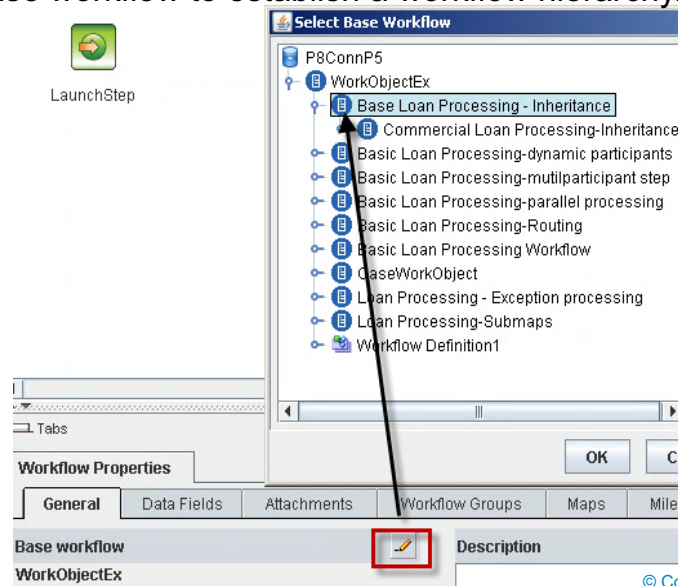
- When the Enable inheritance option is selected, you can choose an existing workflow definition (already transferred to the isolated region) as a base workflow from which to inherit maps, fields, milestones, and so forth. Inherited items are designated as Read Only in the current workflow unless you override them (enabled by default).
- When the Enable inheritance option is not selected, then the base workflow for any new workflow definitions that you create is the system-provided WorkObjectEx. You cannot select another base workflow.

Disabling workflow inheritance for a derived workflow

You can choose to disable workflow inheritance in a workflow that already inherits maps, fields, and other properties from another workflow. In this case, if you disable inheritance, the base map is reset to WorkObjectEx. All inherited items that were not overwritten from the previous base workflow are no longer available. If these inherited items are referenced, validation errors occur.

Create a workflow hierarchy

- You specify a base workflow in Workflow Properties > General > Base Workflow field.
 - Click the Modify icon.
 - The Select Base Workflow window lists only transferred workflows in the isolated region.
 - Select a base workflow to establish a workflow hierarchy.



Workflow inheritance

© Copyright IBM Corporation 2016, 2011

Figure 6-7. Create a workflow hierarchy

Help path

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Select base workflow

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh046.htm



To create a new workflow definition that inherits submaps and other items that are defined in another workflow definition, you select a previously defined workflow definition as the base workflow in the Workflow Properties General tab.

When you select a base workflow for a new workflow definition, by default the Workflow main map of the base workflow definition is not used in the inherited current workflow definition. The Workflow main map of the current workflow definition is retained. You can reactivate the inherited Workflow map by deleting the main map in the new workflow definition.

Submaps that are present in the base workflow definition are designated as Read Only in the current workflow definition. If you want to override this behavior, you can use the Override Inherited Map option.

The diagram on this slide shows that the Base workflow modify (pencil) icon is clicked. The Select Base Workflow window opens. The Base Loan Workflow is selected.

Override an inherited map

- Use the Create Map icon on map toolbar.
 - Select the Override Inherited Map option.
 - Select name of inherited submap that you want to override.
- Inherited submap is no longer Read Only.
 - You can modify the inherited submap.
 - You can modify inherited workflow properties, but **not** delete them.
- In Workflow Properties, symbols indicate the status of maps and workflow fields.
 - Inherited symbol indicates an inherited item 
 - Overridden symbol indicates an item that is inherited and overridden 

Workflow inheritance

© Copyright IBM Corporation 2016, 2011

Figure 6-8. Override an inherited map

Help paths

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Create a map

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh150.htm

FileNet P8 Platform 5.2.1>Integrating workflow into document management>Designing workflows>Define workflow properties>Workflow properties - data fields

http://www.ibm.com/support/knowledgecenter/SSNW2F_5.2.1/com.ibm.p8.pe.designerui.doc/bpfdh172.htm

You can override an inherited submap from a base workflow definition and redefine its steps, routes, and properties, as needed. Use the Override Inherited Map option on the Create Map tool.

After overriding an inherited map, the inherited map is no longer designated as Read Only. The steps on the inherited submap are preserved for you to modify or delete. You can make changes to the inherited submap by adding, deleting, or modifying its steps and routes.

Inherited fields are displayed in the Workflow Properties of a derived workflow and are designated by an Inherited symbol. Generally, you can modify some inherited workflow properties, such as data field descriptions and values. However, you cannot delete them or change the data type of an inherited data field.

In several of the Workflow Properties panes of a derived workflow definition, you can see whether an item is inherited or inherited and modified.

- The Inherited symbol indicates that the item is inherited.
- The Overridden symbol indicates that the item is inherited and modified in the current workflow.

The Inherited and Overridden symbols are displayed on items in the Data Fields, Attachments, Workflow Groups, Maps, and Milestones tabs in the Workflow Properties pane.

Unit summary

- Create a workflow hierarchy
- Override an inherited map

Workflow inheritance

© Copyright IBM Corporation 2011, 2016

Figure 6-9. Unit summary

Exercise: Use workflow inheritance

Workflow inheritance

© Copyright IBM Corporation 2011, 2016

Figure 6-10. Exercise: Use workflow inheritance

Exercise introduction



- Create a workflow hierarchy
- Override an inherited map
- Reactivate and modify the Workflow map
- Test the inherited workflow

Workflow inheritance

© Copyright IBM Corporation 2011, 2016

Figure 6-11. Exercise introduction



IBM Training

