

Senior System Architect

7.2
Exercise Guide

**© Copyright 2017
Pegasystems Inc., Cambridge, MA
All rights reserved.**

Trademarks

For Pegasystems Inc. trademarks and registered trademarks, all rights reserved. Other brand or product names are trademarks of their respective holders.

For information about the third-party software that is delivered with the product, refer to the third-party license file on your installation media that is specific to your release.

Notices

This publication describes and/or represents products and services of Pegasystems Inc. It may contain trade secrets and proprietary information that are protected by various federal, state, and international laws, and distributed under licenses restricting their use, copying, modification, distribution, or transmittal in any form without prior written authorization of Pegasystems Inc.

This publication is current as of the date of publication only. Changes to the publication may be made from time to time at the discretion of Pegasystems Inc. This publication remains the property of Pegasystems Inc. and must be returned to it upon request. This publication does not imply any commitment to offer or deliver the products or services described herein.

This publication may include references to Pegasystems Inc. product features that have not been licensed by you or your company. If you have questions about whether a particular capability is included in your installation, please consult your Pegasystems Inc. services consultant.

Although Pegasystems Inc. strives for accuracy in its publications, any publication may contain inaccuracies or typographical errors, as well as technical inaccuracies. Pegasystems Inc. may make improvements and/or changes to the publication at any time.

Any references in this publication to non-Pegasystems websites are provided for convenience only and do not serve as an endorsement of these websites. The materials at these websites are not part of the material for Pegasystems products, and use of those websites is at your own risk.

Information concerning non-Pegasystems products was obtained from the suppliers of those products, their publications, or other publicly available sources. Address questions about non-Pegasystems products to the suppliers of those products.

This publication may contain examples used in daily business operations that include the names of people, companies, products, and other third-party publications. Such examples are fictitious and any similarity to the names or other data used by an actual business enterprise or individual is coincidental.

This document is the property of:

Pegasystems Inc.
One Rogers Street
Cambridge, MA 02142-1209
USA
Phone: 617-374-9600
Fax: (617) 374-9620
www.pega.com

DOCUMENT: Senior System Architect Exercise Guide

SOFTWARE VERSION: Pega 7.2

UPDATED: 04 19 2017

CONTENTS

COURSE INTRODUCTION	1
Before you begin	2
Senior System Architect 7.2 overview	2
Downloading the exercise system	3
Completing the exercises	4
How to locate a record using the Records Explorer	4
Recreating the Pega 7 search index	5
APPLICATION DESIGN	8
Creating a new application version	9
Exercise: Creating a new application version	9
Configuring application rulesets	12
Exercise: Configuring application rulesets	12
Branching rulesets for parallel development	15
Exercise: Merging a development branch	15
Rule resolution	19
Exercise: Rule resolution	19
Parameterizing rules for reuse	23
Exercise: Parametrizing rules for reuse	23
CASE DESIGN	29
Creating temporary cases	30
Exercise: Working with temporary cases	30
Searching for duplicate cases	33
Exercise: Searching for duplicate cases	33
DATA MODEL DESIGN	39
Configuring a localizable list of values	40
Exercise: Creating work status values	40
Configuring data access patterns	44
Exercise: Configuring data access patterns	44
PROCESS DESIGN	50
Creating organization records	51
Exercise: Updating assignment routing through Dynamic Class Referencing	51
Configuring a cascading approval process	54
Exercise: Configuring a cascading approval process	54
Prioritizing user assignments	62
Exercise: Prioritizing user assignments	62
Delegating rules to business users	67
Exercise: Delegating a business rule	67
Configuring parallel processing	72
Exercise: Configuring parallel processing	72
Improving the user experience with screen flows	78

Exercise: Configuring a screen flow78
Adding attachments	83
Exercise: Adding attachment categories to cases83
Configuring flow action pre- and post-processing	90
Exercise: Configuring flow action pre- and post-processing90
Circumstancing rules on multiple variables	93
Exercise: Circumstancing rules on multiple values93
UI DESIGN	97
Customizing a user portal	98
Exercise: Customizing a harness98
Exercise: Customizing a portal102
Designing a mobile-ready application	106
Exercise: Designing a mobile-ready application106
Customizing the look and feel of an application	108
Exercise: Updating the look and feel of an application108
REPORT DESIGN	113
Creating reports that combine data from multiple tables	114
Exercise: Creating reports that combine data from multiple tables114
DATA MANAGEMENT	119
Exposing an application with a service	120
Exercise: Exposing an application with a SOAP service120
Exercise: Accessing a Pega application using the Pega API132
Reading and writing data to the database	134
Exercise: Reading and writing data to a database134
Simulating integration data	141
Exercise: Simulating integration data141
Addressing integration errors in connectors	146
Exercise: Configuring error handling for data pages146
Managing integration settings	151
Exercise: Managing integration settings151
APPLICATION DEBUGGING	156
Reviewing log files	157
Exercise: Reviewing log files157
Analyzing application performance	160
Exercise: Analyzing application performance160
APPLICATION ADMINISTRATION	162
Securing an application	163
Exercise: Securing an application163
Creating an agent for background processing	168
Exercise: Configuring batch processing168

Migrating an application	172
Exercise: Migrating an application	172
DECISION DESIGN	178
Inside Pega decision management	179
Exercise: Update the prioritization for a strategy	179
Enriching business applications with Next-Best-Action	187
Exercise: Changing the strategy for an interaction	187
The importance of propositions	192
Exercise: Measuring a reject rate for a proposition	192
How to design a decision strategy	200
Exercise: Creating a next-best-action decision strategy	200
COURSE SUMMARY	211
Next steps for Senior System Architects	212
Senior System Architect summary	212

COURSE INTRODUCTION

Before you begin

Senior System Architect 7.2 overview

The Senior System Architect course is an advanced course designed to help Pega 7 architects further their knowledge of Pega 7.

The lessons in this course focus on tasks a senior system architect performs to develop a Pega application.

This course is based on the Pega 7.2 platform. Pega recommends that students complete Pega Platform Fundamentals 7.2 and System Architect Essentials 7.2 prior to starting this course.

Objectives

After completing this course, you should be able to:

- Identify the tasks and responsibilities of the Senior System Architect on a Pega Implementation
- Create and extend a Pega application
- Manage rules and rulesets
- Leverage the Enterprise Class Structure (ECS) to promote rule reuse between case types and applications
- Configure roles, access groups and privileges
- Manage data access within an application
- Create wizards to configure a sequence of assignments
- Design applications for multiple device types and screen sizes, including mobile devices
- Manage integration settings
- Incorporate next-best-action decision-making into applications
- Test your application design to analyze rule behavior and identify configuration errors

Intended audience

This advanced course is designed to help System Architects further their knowledge of Pega 7 and improve their ability to implement Pega 7 solutions in an efficient manner. Upon completion of this course students should be ready to take the Certified Senior System Architect (CSSA) exam.

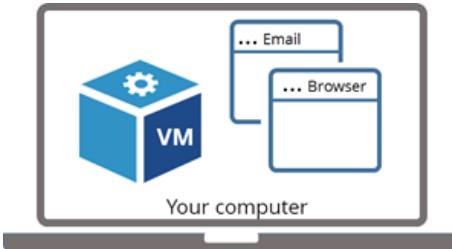
Prerequisites

To succeed in this course, students should have:

- Some familiarity with application development concepts
- Completed the System Architect Essentials 7.2 course

Downloading the exercise system

This course provides an opportunity for you to practice what you learn.



A virtual machine (VM) exercise system is provided. It has its own operating system and it runs software applications such as Pega 7, PostgreSQL (a database server) and Apache Tomcat (a simple web server). The exercise guide and the VM exercise system are made available for you to download from the related content section

Best of all, this VM runs on Windows or OS X.

Exercise VM archive

The exercise VM is made available as an OVA (Open Virtualization Archive) file that contains a compressed, "installable" version of a virtual machine. The Virtual Machine User Guide provides details on how to download and import the OVA file, and details on how to run the VM in one of the recommended VM Players.

Note: If you are using Microsoft Internet Explorer, Microsoft Edge, or Google Chrome, please read the [Virtual Machine User Guide](#) for important information about how to correctly download the exercise VM.

Use your virtualization software (VM Player) to open/import and extract an OVA file.

Important: Although Pega does not provide a VM Player for you to use, we do recommend that you use one of the following: VMWare (Windows), VMware Fusion (macOS), Virtual Box (Windows, macOS). You are responsible for downloading and installing your own VM player. You can use any VM player that is compatible with an OVA file.

To begin, download and install the VM following the instructions in the [Virtual Machine User Guide](#).

Completing the exercises

When learning new concepts or skills, there is no substitute for learning by doing.

This course includes exercises that provide practical, hands-on experience to help you apply your new skills immediately.

The exercises help reinforce the learning objectives of each lesson. They include a scenario that describes a business problem that needs to be solved and an overview of the tasks you need to perform to complete the exercise.

To help you complete the exercises, two levels of support are provided:

- Your Assignment — specifies the high level steps you need to perform to solve the business problem provided in the scenario.
- Detailed Steps — shows the series of steps needed to complete the exercise.

It is our goal that you can complete the exercises using the scenario and the tasks noted in your assignment. Use the detailed steps to check your work, or as a last resort to complete the exercise.

Exercise environment

This course provides a virtual exercise system that allows you to complete the exercises that accompany the course. Instructions for how to download and setup the virtual exercise system are provided in the previous topic.

Exercise guide reference

While individual exercises are included for each lesson online, you may find it useful to download and print out the complete exercise guide. This is available in the related content pane on the right side of this lesson.

How to locate a record using the Records Explorer

Use the Records Explorer to locate all instances of a specific type of record. The Records Explorer displays a list of all records available to you based on your ruleset list. Unlike search, the Records Explorer does not rely on a search index.

When you select a record type in the explorer area of Designer Studio, Pega displays a list of all records of that type. Use the filter control on each column to reduce the number of records displayed to help you locate the correct record to open.

Follow these steps to locate a record using the Records Explorer.

1. In the explorer area of Designer Studio, select **Records**.
2. In the Records Explorer, expand the record category and select the record type for the record you want to locate. Pega displays a list of records of that record type in the work area. For example,

expand the **Process** category and select **Flow action** to view a list of all flow actions on the exercise system, .

3. In the column header for the column that you want to use as a filter, click the **Filter** icon. For example, click the **Filter** icon for the **Applies to** column to filter the list by a class value.

Recreating the Pega 7 search index

To limit the size of virtual machines distributed with Pega Academy courses, the index of rule instances is removed from the system. To use the Search feature in Designer Studio, you must re-index the rules in your exercise environment.

Log into Pega 7 using an operator ID that has access to the *PegaRULES:SysAdmin4* access role.

Note: You can use the administrator account for the training application, or you can use the default Pega 7 administrator account **administrator@pega.com**.

Verify the active system node

1. From the Designer Studio menu, select **System > General > System, Nodes, Requestors**. The *Systems, Nodes & Requestors* landing page is displayed.

Nodes			
Node Name	JVM Temp Directory	Machine Name	Node ID
8b924ce08dde908c31ff6f76cc2509e9	file://runtime/PRGenClasses/ 8b924ce08dde908c31ff6f76cc2509e9		8b924ce08dde908c31ff6f76cc2509e9
c6adb3836479446153195258e8c5cabb	file://runtime/PRGenClasses/ c6adb3836479446153195258e8c5cabb	c6adb3836479446153195258e8c5cabb	

In the **Nodes** section, the active node is annotated with an arrow. Notice the **Node Name** and the **Node ID** are the same.

Tip: Click the **Node Name** to open the *System Node* record in a new tab.

Note: If more than one node is listed, you can safely ignore the inactive nodes.

2. Copy the value of the **Node Name** or **Node ID** to your clipboard.

Verify the search index host node settings

1. From the Designer Studio menu, select **System > Settings > Search**. The System Settings landing page is displayed opened on the Search tab.
2. Expand the **Search Index Host Node Setting** section.

▼ **Search Index Host Node Setting**

Search Index Host Node ID	Search Index File Directory
8b924ce08dde908c31ff6f76cc2509e9	/opt/tomcat/work/Catalina/localhost/prweb/F
Add Node	

Confirm the value for the **Search Index Host Node ID** is the same as the active *Node Name* or *Node ID* identified in step 2.

Add a search index host node setting

1. If a node id is not listed in the Search Index Host Node Setting, click **Add Node**.
2. In the **Search Index Host Node ID** field, enter the active Node ID identified in the *Verify the active system node* section above.
3. In the Search Index File Directory field, enter **/opt/tomcat/work/Catalina/localhost/prweb/PegaSearchIndex** as the value.
4. Scroll to the bottom of the page and click **Submit**.

Reindex rules and classes

The Pega 7 search index is comprised of three components:

- All rules - controls indexing for all Rule- classes. Use this index to search for all records by rule name.
- All data - controls indexing for all Data- classes. Use this index to search for records specific to Data-classes.
- All work - controls indexing for all Work- classes. Use this index to search for active cases by case ID.

Note: Each index must be created, or re-indexed separately.

2. Select an index to create, then click Re-index.

Index Administration		
Enable Indexing		
ENABLE/DISABLE	INDEX TYPE	INDEX FILE SIZE (MB)
<input checked="" type="checkbox"/>	All Rules	1026.636
<input type="checkbox"/>	All Data	0.535

3. In the Re-indexing dialog, select **All Classes** and click **OK**.

Re-Indexing

Note: Re-indexing all classes may take a long time. You should consider re-indexing one or a few classes at a time.

Select which classes to re-index:

- All Classes
- Only Classes listed below
- All Classes except those listed below

OK **Cancel**

A dialog message displays: *Reindexing request was submitted. Status will be refreshed every 10 seconds.* Click **OK** to close the dialog.

The status of the reindexing is displayed on the System: Settings landing page.

ENABLE/DISABLE	INDEX TYPE	INDEX FILE SIZE (MB)
All Rules	1307.235	Indexed 54300 of 163316

Check **Cancel Re-Index**

Caution: Rebuilding the search indexes may take 30 minutes or more to complete per index. It is recommended you rebuild each index separately so as not to affect system performance while you continue to work on exercises.

Important: Do not power off the VM before the re-indexing is complete. You may log out of Pega 7, but do not power off the VM.

APPLICATION DESIGN

Creating a new application version

Exercise: Creating a new application version

Scenario

Your manager wants to preserve the initial starting version of the HRApps application, before developers start configuring application functionality. To ensure that the initial version of the application is preserved unchanged, your manager requests that you create a new version of the application for this development cycle.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Increment the application rule for the HRApps application to create an HRApps 01.01.02 application. Increment the application rulesets to 01-01-02 versions. Update all of the access groups that reference the HRApps application to use the new version of the application.

Important: When you complete this exercise, you lock the 01-01-01 version of each application ruleset. You cannot modify rules contained in a locked ruleset version. To modify a rule after you lock the 01-01-01 ruleset version, open the rule and save a copy of it into the unlocked 01-01-02 version of the ruleset.

Detailed steps

1. Select **Designer Studio > Application > Structure > Ruleset Stack** to open the ruleset stack landing page.
2. Verify that there are no checked out rules for the rulesets. Check in any rules before proceeding.

3. Click **Lock & Roll** to open the Lock & Roll window to increment the application and the application rulesets.

Current Application		
Ruleset	Checked Out	All Rules
HRApps:01.01.01	0	646
HRAppsInt:01-01-01	0	3
TGB:01-01-01	0	12
TGBInt:01-01-01	0	3
UI-Kit-7:07-01-01	0	1507
CreditCheck:01-01-01	0	19

4. In the Lock & Roll window, select all application rulesets to be locked.
5. For each selected ruleset, in the **Password** field, enter **password** to lock the ruleset.
6. Select **Roll** for all application rulesets.
7. For each selected ruleset, in the **Roll to Version** field, enter **01-01-02**.
8. Update the **Description** to the new version.
9. Select **Create a new version of my Application**.
10. Under **Create a new version of my Application**, in the **Roll to Version** field, enter **01.01.02**.

11. Click **Run** to lock and roll the ruleset and create a new version of the application.

Lock	Ruleset version	Password	Roll	Roll to Version	Description
<input checked="" type="checkbox"/>	HRApps:01-01-01	password	<input checked="" type="checkbox"/>	01-01-02	HRApps:01-01-02
<input checked="" type="checkbox"/>	HRAppslnt:01-01-01	password	<input checked="" type="checkbox"/>	01-01-02	HRAppslnt 01-01-02
<input checked="" type="checkbox"/>	TGB:01-01-01 + Prerequisites	password	<input checked="" type="checkbox"/>	01-01-02	TGB 01-01-02
<input checked="" type="checkbox"/>	TGBInt:01-01-01 + Prerequisites	password	<input checked="" type="checkbox"/>	01-01-02	TGBInt 01-01-02
<input checked="" type="checkbox"/>	UI-Kit:7:07-01-01			07-01-02	This UI Kit is compatible with Pega 7.2.1
<input checked="" type="checkbox"/>	CreditCheck:01-01-01	password	<input checked="" type="checkbox"/>	01-01-02	Records for CreditCheck

Do not update my Application
 Update my Application to include the new RuleSet Versions
 Create a new version of my Application

10 Roll to Version: 01.01.01
 Update Description: HR Apps

11 Run

12. Update the access groups to use the new application version. The operators in the updated access groups are now using application version 01.01.02.

Tip: From the Records Explorer, select **Security > Access Group** to list all access groups. Filter on HRApps to find the access groups that require updating.

You can also access the list of all access groups from **Designer Studio > Org & Security > Groups & Roles**.

Edit Access Group: Administrators for the HRApps application

ID: HRApps:Administrators RS: HRApps [Edit]

Definition Advanced Operators History

Application

Name * HRApps Version * 01.01.02

13. Log out and log in to update your application to default to the new ruleset version when creating new records.

Note: Until you log out and log in, Pega continues to use the original version of the rolled ruleset — in this case, HRApps:01-01-01 — to save records. Since this ruleset version is now locked, Pega warns you that the record cannot be created. Logging out and logging in resets the ruleset version used when creating new records and resolves the issue.

Configuring application rulesets

Exercise: Configuring application rulesets

Scenario

The Human Resources (HR) department requests the ability to update rules that describe department and company policies in the HRApps application.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

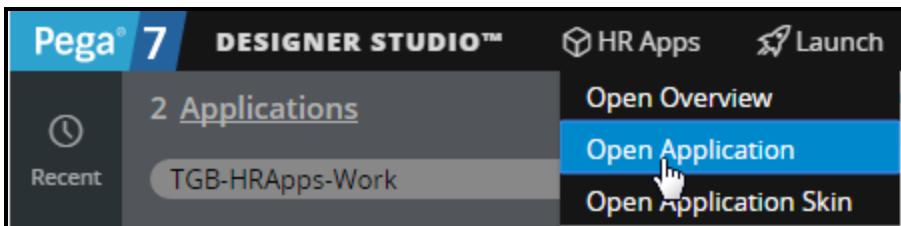
Your assignment

Add the *HR_DelegatedRules* ruleset to the HRApps application as a production ruleset to support the delegation of rules to the HR department for maintenance.

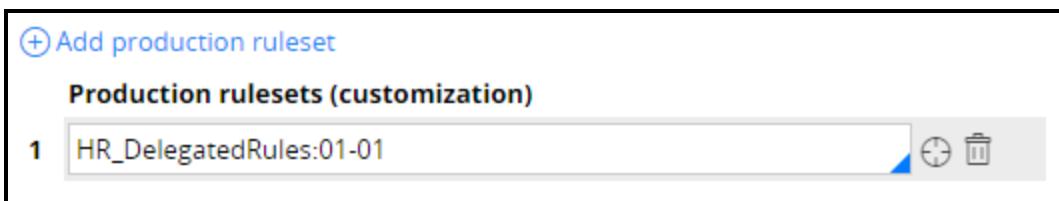
Detailed steps

Follow these steps to add rulesets to an application rule.

1. Open the *HRApps* application rule.

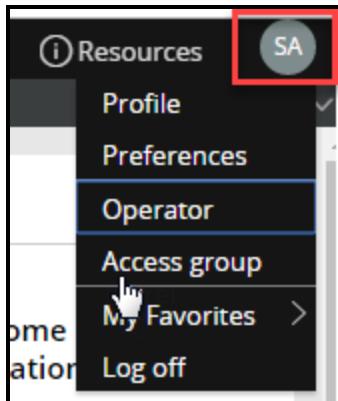


2. Expand the Advanced section and add the *HR_DelegatedRules* ruleset to the **Production rulesets** list.



Note: You must specify the ruleset version manually. Minimally, you must specify the major-minor version numbers.

3. Save your changes to the application rule record.
4. Edit the Access group to which SSA@TGB belongs.



Note: Rulesets added to the **Production rulesets** list require that specific access be granted through the access group.

5. On the **Advanced** tab, add the *HR_DelegatedRules* ruleset to the **Production Rulesets** list.

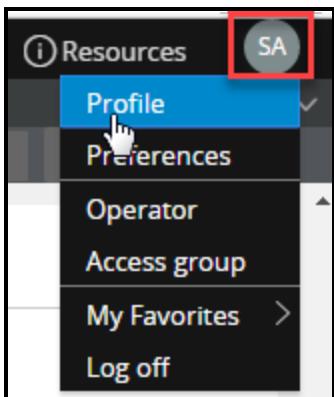
A screenshot of the 'Run time configuration' and 'Production Rulesets' sections of the Advanced tab. The 'Run time configuration' section contains two checkboxes: 'Enable accessibility add-on (Please ensure PegaWAI has first been installed)' (unchecked) and 'Enable offline support' (checked). Below these is a button labeled 'Force full sync for all users' and a note stating 'Full sync has not yet been forced.' The 'Production Rulesets' section shows a list with one item: '1 HR_DelegatedRules:01-01'. There are icons for adding (+) and deleting (-) records.

6. Save your changes to the access group record.
7. Add the *HR_DelegatedRules* ruleset as a **Production Ruleset** to the *HRApps:Managers* access group.

Tip: Use the Records Explorer to locate the access group. Access groups are in the Security category.

Verify your work

1. Open the operator profile for SSA@TGB.



2. The *HR_DelegatedRules* ruleset is available in the Rulesets list.

	Name Position ID Phone	Senior System Architect Senior System Architect SSA@TGB	Organization Division Unit Work Group	TGB Div Unit default
Skills		Access		
Name	Rating	Application Access group Portal Layout Work Pool	HRApps 01.01.01 HRApps:Administrators Developer TGB-HRApps-Work	
Time		Server		
Calendar	USDefault	Name pega	Directory	/webwb
Time Zone	America/New_York	Node 10.60.214.102	URI	/prweb/
Locale Settings				
Current Locale				
Roles		Rulesets		
HRAcc:Administrator PegaRULES:SecurityAdministrator PegaRULES:PegaAPI		SSA@TGB: HR_DelegatedRules:01-01 HRApps:01-01 HRAppsInt:01-01 TGB:01-01-01 TGBInt:01-01-01		

Branching rulesets for parallel development

Exercise: Merging a development branch

Scenario

The Payroll Setup case type allows users to automatically deposit their paychecks into a checking or savings account. On the *Provide bank information* step, employees enter the name of their financial institution, the routing number for the financial institution, and their account number.

Provide bank information

Administrator

Bank name

Routing number* Attach voided check

Account number*

Cancel Save Submit

This manual data entry leads to errors that cause the first deposit to fail. To reduce the number of errors, the Human Resources (HR) department requested that users select their financial institution from a drop-down list, so the application populates the routing number on the form.

Another development team redesigned the form. To avoid conflicts, this team configured the changes in a development branch. You are responsible for merging the changes into the application.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Merge the contents of the development branch labeled *Branching* into the HRApps and TGB rulesets. The branch rulesets are locked with the password **rules**.

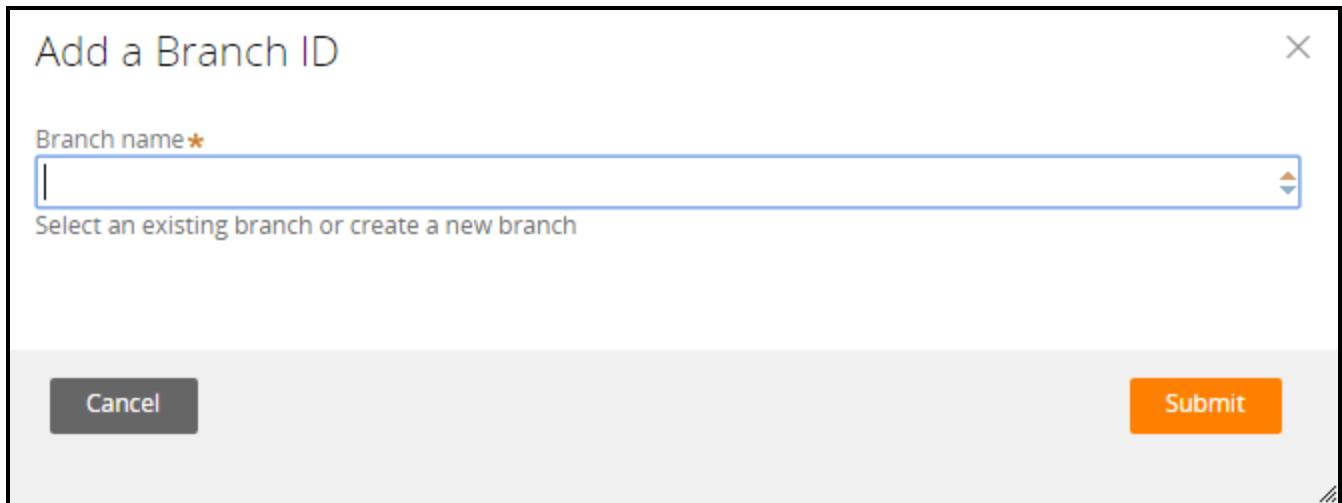
Detailed steps

Add the branch to the application rule

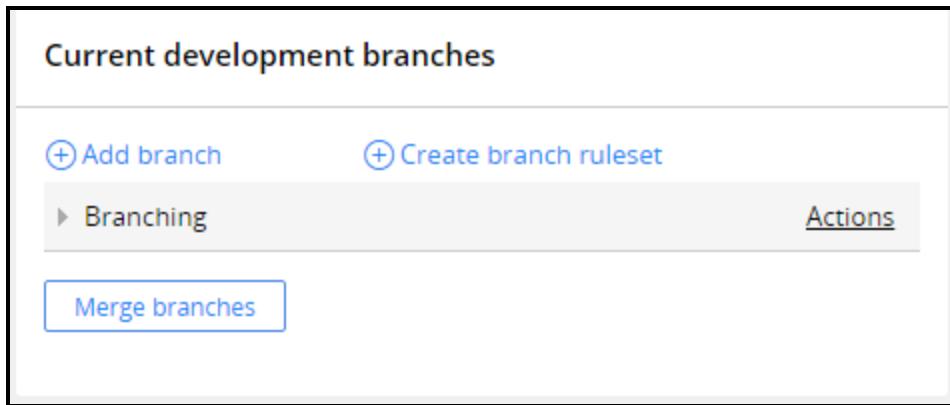
1. Open the *HRApps* application record.

Tip: You can access the application record from the Application menu. Open the Application menu and select **Open Application**.

2. Under *Current development branches*, click **Add branch**. The Add a Branch ID dialog is displayed to allow you to specify the name of the branch to add.



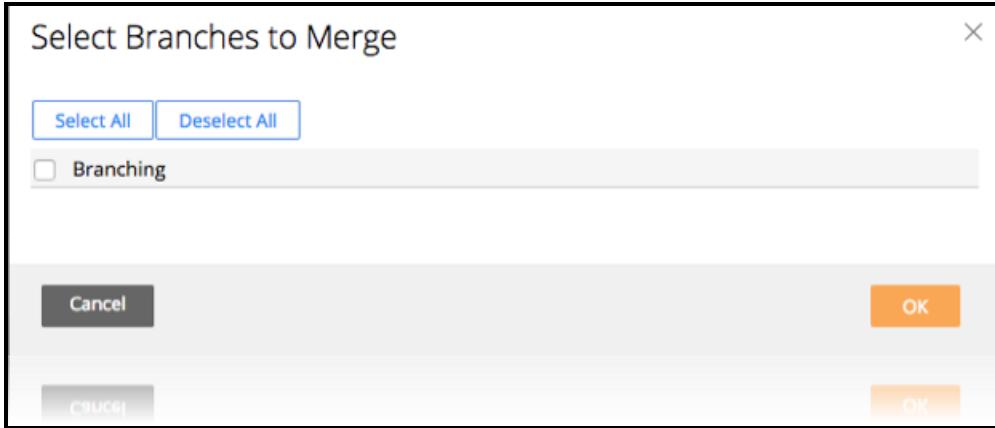
3. In the **Branch name** field, enter **Branching**.
4. Click **Submit**. The dialog is dismissed and the branch ruleset is added to the application rule.



5. Click **Save** to save the change to the application record and add the branch rulesets to the ruleset stack.

Merge the branch contents into the HRApps and TGB rulesets

1. In the *Current development branches* section of the application record, click **Merge branches**.



The *Select Branches to Merge* dialog displays a list of development branches currently associated with the application.

2. Select the **Branching** branch and click **OK** to dismiss the dialog and open the Merge Branches form.
3. If the *Target checked out* count for a target ruleset is not zero, click the number to identify the rule or rules that are checked out in the application ruleset. Then, click the *Total candidates* number for the branch ruleset and confirm that the rules in the branch are not the same rules that are checked out.

A screenshot of the "Branching" tab of the Merge Branches form. It shows two sections: one for HRApps and one for TGB. Each section has a "Source ruleset" and a "Target ruleset".
HRApps Section:
Source ruleset: HRApps
Target ruleset: HRApps
Total candidates: 2 (highlighted)
Source checked out: 0
Target checked out: 0
No conflicts or warnings found.
Below this is a "2 Passwords" section with fields for "Source version" (01-01-01) and "Lock target after merge" (checkbox).
TGB Section:
Source ruleset: TGB
Target ruleset: TGB
Total candidates: 8 (highlighted)
Source checked out: 0
Target checked out: 0
No conflicts or warnings found.
Below this is a "2 Passwords" section with fields for "Source version" (01-01-01) and "Lock target after merge" (checkbox).

Note: The counts for *Target checked out* and *Warnings* should be zero (0), but may vary according to how you have developed your application.

4. For each target ruleset, select the highest available ruleset version.

Note: If you performed the *Creating a new application version* exercise, the highest available version for each version is 01-01-02. Otherwise, the highest available version for each ruleset is 01-01-01.

5. For each ruleset, in the **Source version** field, enter **rules** as the password.
6. Optional: Click the **Keep all source rules and rulesets after merge** check box to prevent Pega from deleting the branch ruleset, and allow you to repeat the exercise.

7. Click **Merge**. The Merge Branches wizard displays a confirmation page to indicate that the merge was successful.
8. Return to the application record.
9. In the *Current development branches* section, click **Actions** and select **Remove from application** to remove the branch from the application.
10. Click **Save** to save the change to the application record.

Verify your work

1. Create a new *Payroll Setup* case.
2. Advance the case to the *Provide bank information* step. Verify that the bank name field is now a drop-down list, and that selecting a bank from the list automatically populates the routing number field.

The screenshot shows a user interface for entering banking details. At the top left is the title "Provide bank information". To the right is a user profile icon labeled "Administrator". Below the title are three input fields: "Bank name" containing "1199 SEIU FEDERAL CREDIT UNION", "Routing number*" containing "226077862", and "Account number*" which is empty. To the right of the "Bank name" field is a blue button labeled "Attach voided check". At the bottom right are three buttons: "Cancel", "Save", and "Submit" (highlighted in orange).

Rule resolution

Exercise: Rule resolution

Scenario

Human Resources (HR) business partners create benefits enrollment and employee evaluation cases on behalf of TGB employees. To ensure that each case is created for the appropriate employee, HR business partners have asked that the employee information header area for each case type match the header used for Onboarding cases.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Save a copy of the *EmployeeInformation* section from the Onboarding case type to the class group (TGB-HRApps-Work) and add it to the HRApps:01-01-02 ruleset.

Note: Before you begin this exercise, complete the exercise *Creating a new application version* to lock the HRApps:01-01-01 ruleset and create the HRApps:01-01-02 ruleset.

Withdraw the *EmployeeInformation* section in the Benefits Enrollment, Employee Evaluation, and Onboarding case types.

Detailed steps

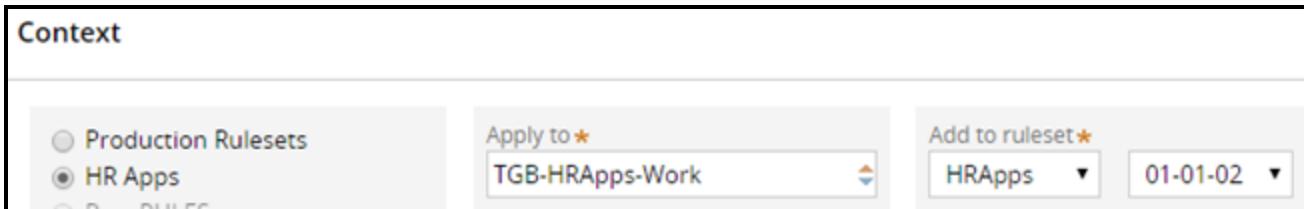
Copy the *EmployeeInformation* section record to the new context

Create a copy of the *EmployeeInformation* section in the TGB-HRApps-Work class to share between all the case types in the HRApps application.

1. In the Onboarding case type, open the *EmployeeInformation* section.

Tip: Section records are in the User Interface category. From the App Explorer, expand **Onboarding > User Interface > Section**.

- Save a copy of the *EmployeeInformation* section, and set the context of the record to the **TGB-HRAApps-Work** class and the **HRAApps:01-01-02** ruleset.

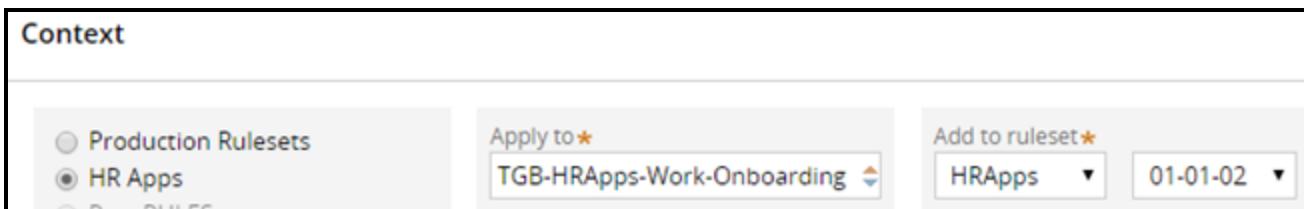


Withdraw the *EmployeeInformation* section record in the Onboarding case type

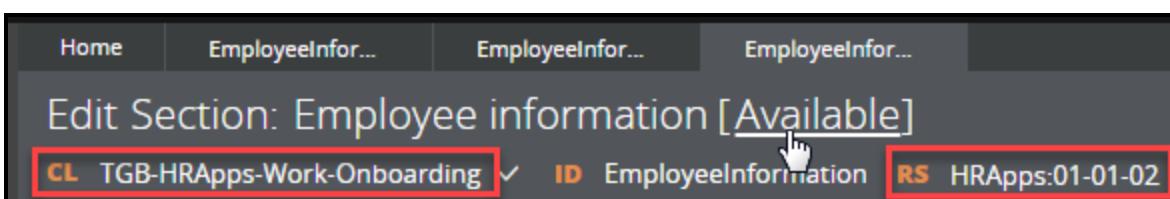
Since the original version of the rule is in a locked ruleset, create a new version of the *EmployeeInformation* section in the Onboarding class. Then, withdraw the new version of the rule to prevent Pega from using the section in the Onboarding class. Withdrawing the new version prevents Pega from using the version of the rule in the locked ruleset, so Pega uses the version in the TGB-HRAApps-Work class instead.

- Save another copy of the *EmployeeInformation* section record, and set the context of the record to the **TGB-HRAApps-Work-Onboarding** class and the **HRAApps:01-01-02** ruleset.

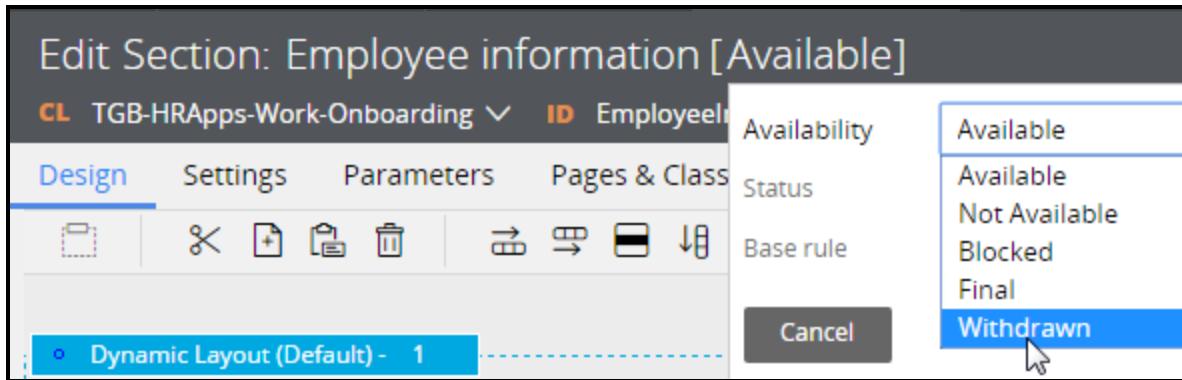
Tip: Because HRAApps:01-01-01 is a locked ruleset, you need to save a copy of the *EmployeeInformation* section in the Onboarding case type and the HRAApps:01-01-02 ruleset so you can change the availability to **Withdrawn**.



- Edit the availability of the *EmployeeInformation* record.



- Set the availability of the record to **Withdrawn**.



- Click **OK** to update the availability of the section.
- Click **Save**. The *EmployeeInformation* section record in TGB-HRApps-Work-Onboarding is now withdrawn.

Important: Remember to check in any records that you check out to configure.

Withdraw the *EmployeeInformation* section record in the Employee Evaluation and Benefits Enrollment case types

Create a new version of the *EmployeeInformation* section in the Employee Evaluation and Benefits Enrollment class. Then, withdraw the rules to prevent Pega from using the version of the section created for each case type and force Pega to use the version in the TGB-HRApps-Work class.

- Open the *EmployeeInformation* section in each of the case types.
- Save a copy of the record, set the context to the appropriate class, and add the record to the **HRAapps:01-01-02** ruleset.
- Set the availability of the section records to **Withdrawn**.
- Save your changes to withdraw the section.

Verify your work

- Create a new Onboarding case.
- Complete the required fields in the *Collect Employee Info* step.
- Click **Submit** to advance the case.

4. Use **Live UI** to select the *EmployeeInformation* section include record, and click the **Crosshair** icon to open the record in Designer Studio.

The screenshot shows a 'Live UI' interface for a 'Identify Home Office' form. On the right, there's a circular badge with 'SA' and the text 'Senior System Architect'. Below it is a table with two rows. The first row contains 'First Name' (Ray) and 'Last Name' (d'Ator). The second row contains 'Manager' (Someone) and 'Department' (---). A green rectangular selection box surrounds the entire table. In the top right corner of the table area, there's a toolbar with icons: a plus sign, a gear, an info circle, and a list. A hand cursor is hovering over the plus sign icon. A tooltip window titled 'Open rule in Designer Studio' is displayed at the bottom right of the toolbar.

5. Confirm the context of the **EmployeeInformation** section record is in the **TGB-HRApps-Work** class and the **HRApps:01-01-02** ruleset.

The screenshot shows an 'Edit Section' dialog titled 'Edit Section: Employee Information [Available]'. At the top, there are three tabs: 'CL TGB-HRApps-Work' (selected), 'ID EmployeeInformation', and 'RS HRApps:01-01-02'. All three tabs are highlighted with a red border.

6. Create a new Benefits Enrollment case and follow the steps above to confirm the context of the *EmployeeInformation* section record is in the **TGB-HRApps-Work** class and the **HRApps:01-01-02** ruleset.
7. Create a new Employee Evaluation case and confirm the context of the *EmployeeInformation* section record is in the **TGB-HRApps-Work** class and the **HRApps:01-01-02** ruleset.

Parameterizing rules for reuse

Exercise: Parametrizing rules for reuse

Scenario

TGB provides employees with three types of insurance coverage: medical, dental, and vision. The full set of coverage choices are located in a single database table. The Human Resources (HR) department wants users to select one of the insurance plans offered for each type of coverage. The HR department also wants to present a form listing the available plans for each type of coverage.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

A system architect created a form for each insurance coverage option (medical, dental, and vision) in the implementation application, and added each form to the benefits enrollment case type. Complete the configuration to present a filtered list of coverage options that match the plan type for each form.

- Create a Benefits Enrollment case. Advance the case to the Select Medical Coverage form. Observe the current behavior of the form.
- Configure the *GetHRPlansList* report to accept a text parameter named *Type*. Add a filter to the report to only display insurance plans where the *.Type* column matches the value of the parameter.
- Configure the *D_HRPlanList* data page to accept a string parameter named *Type*. Configure the data page to pass the parameter as the value of the parameter for the *GetHRPlansList* report definition that sources the content of the data page.
- Configure the *HealthcareBenefit* section to accept a parameter named *Type*. Configure the drop-down list to pass the parameter to the *D_HRPlanList* data page.
- Update the sections for each healthcare plan selection to pass the corresponding plan type as a parameter value.

Note: The data type for insurance plans is defined as part of the built-on application, not the implementation application. When completing this exercise, copy the rules you need to modify to the HRApps ruleset.

Tip: Use Live UI to identify and open the sections used to generate the form.

Detailed steps

Observe the current behavior of the Select Medical Coverage form

Create a new benefits enrollment case to observe the current behavior of the Select Medical Coverage form. Keep the case open so you can use Live UI to open the sections used by the form to complete your configuration changes.

1. Open the Benefits Enrollment case in the Case Designer.
2. Click **Run** to create a new benefits enrollment case.
3. Advance the case to the Select Medical Coverage step. Note that the **Plan name** drop-down list displays all of the insurance coverage options offered by TGB, including options for dental and vision coverage.

Configure the report definition with a parameterized filter

Add a parameter to the *GetHRPlansList* report definition to filter results by the type of insurance plan. This allows Pega to filter the results when the report is run.

1. Open the *GetHRPlansList* report definition in PegaHR-FW-HRFW-Data-HRPlans.

Tip: The report definitions are listed in the **Reports** category in the Records Explorer. Click the **Report Definition** rule type to list all of the report definition rules, then use the filter control to find the *GetHRPlansList* report definition.

2. On the **Parameters** tab, add a parameter using the information provided in the following table.

Name	Description	Data Type	Default value
Type	Healthcare Benefit Option	String	

Important: Leave the **Default Value** field empty. The *Type* parameter is used to determine the type of healthcare benefit on three UI forms. Each form presents a different type of benefit: medical, dental, or vision. As a result, a default value would be incorrect on at least two of the forms.

3. On the **Query** tab, add a filter to the report to only display insurance plans where the *.Type* column matches the value of the parameter.

Condition	Caption	Column source	Relationship	Value
A	Healthcare Benefit Option	.Type	Is equal	Param.Type

4. Save your changes to complete the configuration of the report definition.

Configure the data page to pass the parameter to the report definition

Update the *D_HRPlanList* data page to pass a parameter value to the report definition. This populates the data page with a filtered list of insurance plans, using the parameterized report definition.

1. Open the *D_HRPlanList* data page.

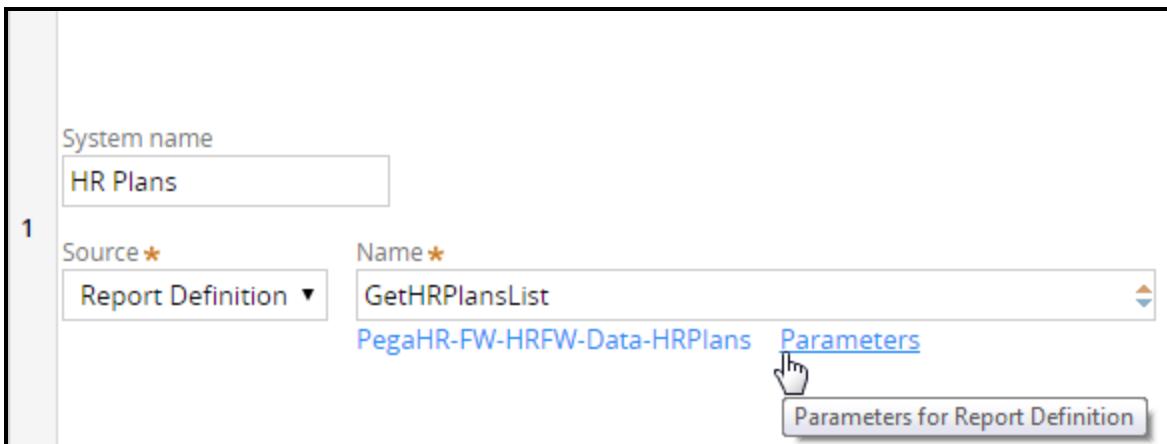
Tip: The *D_HRPlanList* data page is listed under the HR Plans data type in the Data Explorer.

2. On the **Parameters** tab, add a parameter using the information provided in this table.

Name	Description	Data Type	Required	Smartprompt type	Validate as
Type	Healthcare Benefit Option	String	No		

Important: Leave the **Smartprompt type** and **Validate as** fields empty. The parameter value is passed by another rule, rather than specified by user input. As a result, there is no need to enter a Smartprompt type to provide users with a list of values. Neither is there a need to validate user input.

3. On the **Definition** tab, in the Data Sources section, click the **Parameters** link.



- In the **Value** field, select **Param.Type** from the list of available options.

Parameters for GetHRPlansList

<input type="checkbox"/> Skip Summary Processing	
<input type="checkbox"/> Pass current parameter page	
PARAMETER	VALUE
Type*	<input type="text"/> <div style="border: 1px solid #ccc; padding: 2px; width: 150px;"> Parameters Param Param.Type Healthcare Benefit Option </div>
Cancel	

Tip: When using parameters across multiple rule types, the name of the parameter should be consistent.

- Click **Submit**.
- Save your changes to complete the configuration of the data page.

Tip: Now is a good time to test the data page. From the **Actions** menu, select **Run**. In the **Value** field, enter **Medical**, **Dental**, or **Vision**, and click **Run**. Check the contents of pxResults to confirm the plans are listed.

Configure the HealthcareBenefit section to pass a parameter value to the data page

Update the *HealthcareBenefit* section to pass a parameter to the data page. This populates the drop-down list with the filtered list of plans from the data page.

- Open the *HealthcareBenefit* section in PegaHR-FW-HRFW-Data-HRPlans.
- Tip:** Use Live UI to identify and open the section from the Select Medical Coverage form.
- On the **Parameters** tab, click **None** and select **Standard** from the drop-down list to create a standard parameter for the section. Pega displays a warning dialog to inform you that the parameter type cannot be changed after you save the section.
- Note:** Once you save the section, the drop-down control is no longer available on the Parameters tab.
- Click **Submit** to dismiss the dialog and return to the Parameters tab.
 - On the **Parameters** tab, add a parameter using the information provided in the following table.

Name	Description	Data Type	Required	Smartprompt type	Validate as
Type	Healthcare Benefit Option	String	No		

Important: Leave the **Smartprompt type** and **Validate as** fields empty.

- On the **Design** tab, edit the properties of the **Plan name** drop-down control.
- In the **List source** section, delete the name of the data page, then add it back to force a refresh of the properties panel.

Note: Pega only updates the parameters for the data page when you change the name of the page referenced as the source of the list. Performing this step of the exercise forces Pega to refresh the contents of the properties panel to display a field for the parameter you added to the data page.

- In the **Value** field, enter **Param.Type**.

List source	
Type	Data page
Data page*	D_HRPlanList
PARAMETER	
Type	Param.Type

- In the **Property for value** field, enter **.ID** as the value.

Note: When you reselect the data page to refresh the properties panel, Pega clears the entry in the **Property for value** field. You must reenter this value to complete the configuration of the drop-down control.

- Click **Submit** to close the Properties dialogue for the drop-down control.
- Save your changes to complete the configuration of the section.

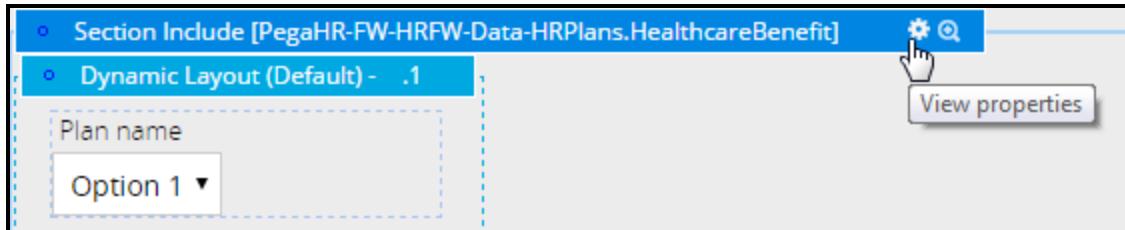
Configure the section include with an applicable parameter value

Update each section include to pass a parameter to the section in the data class that displays the insurance plan information. This allows you to customize the data presented for each of the three steps in the case life cycle.

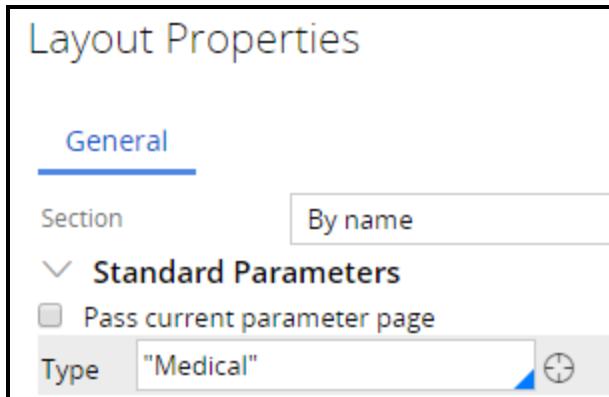
- Open the *SelectMedicalCoverage_0* section in TGB-HRApps-Work-BenefitsEnrollment.

Tip: Sections are in the User Interface category.

- Edit the *Layout properties* of the section include.



3. In the **Type** field, enter "Medical".



Note: When entering a literal string, such as **Medical**, encapsulate the string in double quotes. This prevents Pega from interpreting the string incorrectly.

4. Save your changes to complete the configuration of the user form.

Verify your work

Verify your configuration of the *SelectMedicalCoverage_0* section. Once you confirm that your configuration works, repeat the procedure in *Configure the section includes with an applicable parameter value* to add a parameter value of "Dental" to the *SelectDentalCoverage_0* section, and a parameter value of "Vision" to the *SelectVisionCoverage_0* section.

1. Create a new Benefits Enrollment case.
2. Advance the case to the Insurance Selection stage.
3. On the **Select Medical Coverage** screen, confirm only *Medical plans* are displayed in the drop-down list.

CASE DESIGN

Creating temporary cases

Exercise: Working with temporary cases

Scenario

The Human Resources (HR) department at TGB has determined that more than 50 percent of new applications for an open position are discarded because the candidate is not able to provide the information necessary to complete the application and qualify for further evaluation. The required information is entered as the user collects the candidate's details. To avoid accumulating a large number of abandoned candidate cases in the system, HR wants only cases in which the applicant has provided the required information to be entered into the system and tracked throughout the case life cycle.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

The requirement states that cases will not be persisted in the database when they are created. The requirement also states that after the work sample is collected in the Collect Candidate Details process, the case will be persisted. Persisting the case allows the case to advance through the case life cycle and be processed by other users.

To support these requirements, you must do the following things:

- Configure the starting flow for the candidate case (`pyStartCase`) so that a candidate case is created as a temporary case.
- Add a Persist Case Smart Shape after the final step in the Collect Candidate Details process. This persists the case to the database after the user submits the candidate's work sample.

Note: The user interface (UI) should include validation logic such as validation rules or required fields to verify that the case contains all of the information required for processing. Otherwise, cases that are not qualified for further processing may be persisted. This does not conform to the company's business policies.

Detailed steps

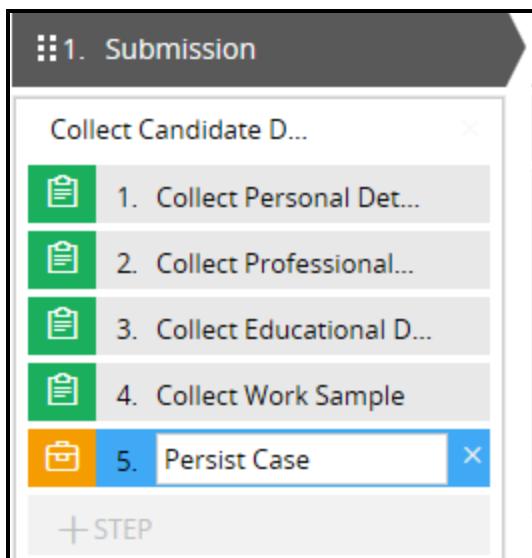
Configure the starting flow

- From the Cases Explorer, right-click **Candidate** and select **Open** from the menu to open the Candidate case type rule.

- On the **Processes** tab, in the **Starting processes** section next to the **pyStartCase** field, click the **Crosshair** icon to open the *pyStartCase* flow rule.
- Note:** If you completed the exercise for *Creating a new application version*, save the *pyStartCase* record into the highest available unlocked ruleset version to be able to edit the record.
- On the **Process** tab, locate the Case Creation setting section and select the **Temporary object** check box.
 - Click **Save**. A new case will not be persisted to the database.

Add a Persist Case utility to the process

- From the Cases Explorer, open the Candidate case type.
- On the Submission stage click **+Step** beneath the Collect Work Sample step and select **More**.
- In the dialog displayed beneath the new step, expand the Utilities section.
- Select **Persist Case** to display the Persist Case panel.
- In the Persist panel, click **Select**. The Submission stage now displays the Persist Case step.



- On the case life cycle header, click **Save**. The process now includes the Persist Case shape.

Verify your work

- Create a Candidate case.
- Complete the Collect Personal Details form and click **Submit**.

Note: The required fields on the form are examples of information that is required before a case should be persisted.

3. Open the Clipboard and select *pyWorkPage*. Note that *pyTemporaryObject* is set to **true** and there is no *pzInsKey* value. No instance is persisted to the database.

pyActionInfo (Embed-UI-RunTime-Element)	pyTemplateDisplayText
pyDocuments (classless)	pyTemplateInputBox
pyFilterSettings (classless)	pyTemporaryObject true
pyPortal (Data-Portal)	pyTransientAssignmentStatus New
pyReportParamPage (Embed-QueryInputs)	pyWindowTitle
pyWorkPage (TGB-HRApps-Work-Candidate)	pyWorkIDPrefix C-
Candidate (TGB-HRApps-Data-Candidate)	pyWorkPartiesRule pyCaseManagementDefault
Education	pzInsKey
Experience	pyStageStartInProgress false

4. Advance the case through the *Collect Work Sample* step. Click **Submit** to see what happens when a case is persisted.
 5. Open the Clipboard and select *pyWorkPage*.

Note: The system created a case instance (in this example, *pzInsKey TGB-HRAPPSC-17*). This is persisted to the database.

pyActionInfo (Embed-UI-RunTime-Element)	pyTemplateDisplayText
pyDocuments (classless)	pyTemplateInputBox
pyWorkPage (TGB-HRApps-Work-Candidate)	pyTemporaryObject true
pyFilterSettings (classless)	pyTransientAssignmentStatus New
pyPortal (Data-Portal)	pyWindowTitle
pyReportParamPage (Embed-QueryInputs)	pyWorkIDPrefix C-
ReportParameters (Embed-QueryInputs)	pyWorkPartiesRule pyCaseManagementDefault
ShapeList (Code-Pega-List)	pzDontResolveCoverCase false
pyWorkPage (TGB-HRApps-Work-Candidate)	pzIndexCount 1
Candidate (TGB-HRApps-Data-Candidate)	pzInsKey TGB-HRAPPSC-17
Education	pyStageStartInProgress false
Experience	

Searching for duplicate cases

Exercise: Searching for duplicate cases

Scenario

Human Resources (HR) policy requires each manager at TGB to evaluate their direct reports annually. HR uses the evaluations to determine which of the following actions to take:

- Retain the employee in their current position
- Qualify the employee for a promotion
- Place the employee on a development plan
- Terminate the employee

HR wants to ensure that a manager only submits one evaluation for each employee. If a manager submits a second evaluation for an employee within the same evaluation period, the HR Apps application should identify the existing evaluation and identify the new evaluation as a duplicate case.

All duplicate cases are identified by the employee's name. However, in specific situations, some conditions may indicate that the cases are not duplicates. Some of these specific conditions are included in the following list:

- The employee changed positions during the evaluation period.
- The employee had more than one manager because the employee changed positions.
- More than one evaluation period start date may exist because the employee changed positions.
- More than one evaluation period end date may exist because the employee changed positions.

HR has determined that when two or more of the specific conditions are true, the cases are duplicates.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Use the **Track duplicates** feature for the Employee Evaluation case type to identify duplicate cases. Use the following information to define the evaluation:

- Identify duplicate cases based on the following weighted match conditions:
 - The manager's last name (*.Employee.Manager*)
 - The employee's position (*.Employee.pyPosition*)
 - The evaluation start date (*.StartDate*)

- The evaluation end date (*.EndDate*)
Each condition carries a weight of 25, and a case is considered a duplicate case if the weighted sum is at least 50.
- Configure *.Employee.pyFullName* as an exact match condition.
- In the Create assessment process, add a Duplicate Search shape after the Identify assessment period step. The duplicate case evaluation occurs after the user submits the evaluation start and end dates.

Detailed steps

Define the weighted match conditions

1. In the Cases Explorer, select **Employee Evaluation**.
2. On the **Settings** tab, click **Track duplicates**. This displays a panel in which you configure the conditions for tracking duplicate cases.

The screenshot shows the 'Settings' tab selected in the Cases Explorer. On the left, there's a sidebar with various configuration sections like Data model, Life cycle, Views, and Settings. Under Settings, there are several tabs: General, Actions, Attachment categories, Case-wide supporting processes, Email instantiation, and Track duplicates. The 'Track duplicates' tab is active. It contains a section for 'Weighted conditions' with a field 'Case is duplicate when sum >= *' containing the value '50'. Below this is a checkbox for 'Configure must match conditions'. At the bottom of the sidebar, there's a section titled 'Track duplicates' with the sub-instruction 'Logic to match existing cases with new ones'. This entire section is highlighted with a red box.

3. In the **Case is duplicate when sum >=** field, enter a threshold of **50**.

4. Above the field, click **+ Add condition**. The panel is displayed in which you add the condition.

+ Add condition

Provide the values based on the function selected

Use 'Primary' when referencing the current case (Example: Primary.FirstName)

Weight Condition

=

OK

5. In the **Weight** field, enter **25**.
6. In the first **Condition** field, enter **Primary.StartDate**. Do not change the = (equals) boolean.

Note: Use the *Primary* keyword in these fields to distinguish between property values in the current case and existing cases. For example, enter *Primary.FirstName* and *.FirstName* in the fields for the [first string] equals [second string] function.

7. In the second **Condition** field, enter **.StartDate**.
8. Click **OK** to add the condition to the Track duplicates configuration.

+ Add condition

Provide the values based on the function selected

Use 'Primary' when referencing the current case (Example: Primary.FirstName)

Weight Condition

25 Primary.StartDate = .StartDate

OK

The **Weighted conditions** list displays the condition.

Track duplicates
Automatically track duplicate cases using the following conditions

Weighted conditions

25	Primary.StartDate = .StartDate
+ Add condition	
Case is duplicate when sum >=*	
<input type="text" value="50"/>	
<input type="checkbox"/> Configure must match conditions	

9. Repeat steps 4 through 8 to add conditions for the remaining properties. When you are done, your condition list will look like the following image:

Track duplicates
Automatically track duplicate cases using the following conditions

Weighted conditions

25	Primary.Start.Date = .StartDate
25	Primary.EndDate = .EndDate
25	Primary.Employee.Manager = .Employee.Manager
25	Primary.Employee.pyPosition = .Employee.pyPosition
+ Add condition	
Case is duplicate when sum >=*	
<input type="text" value="50"/>	
<input type="checkbox"/> Configure must match conditions	

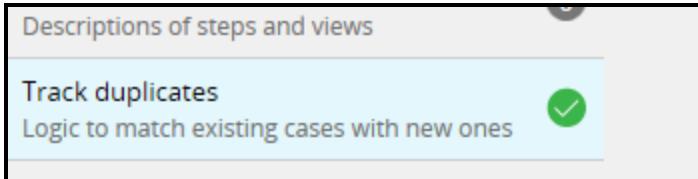
10. Click **Save** to store your settings in the pyDefaultCaseMatch case match rule.

Add an exact match condition to the duplicate logic

1. Add an exact match condition for the candidate name as defined in your assignment.

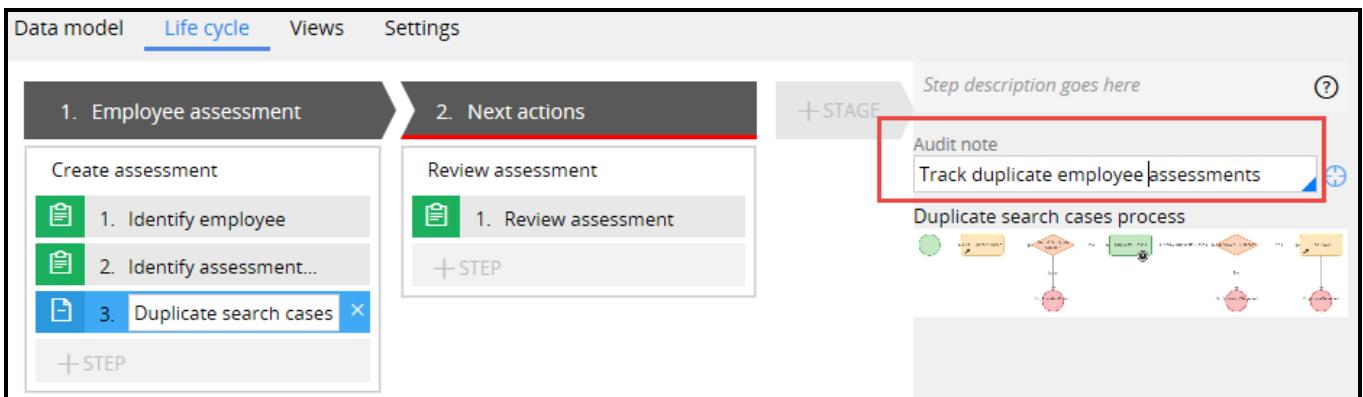
Tip: Use the procedure described in the help topic [Adding an exact match condition to duplicate logic](#). In the **Potential duplicates** field, enter `.Employee.pyFullName`. Do not change the **is same** Operation value.

- When you have completed your updates, click **Save** on the case life cycle header. This stores your settings in the `pyDefaultCaseMatch` case match rule. A check mark is displayed next to the **Track duplicates** option on the Settings tab.



Add the Duplicate search cases Smart Shape

- In the Employee Evaluation case, select the **Life cycle** tab.
- In the Employee assessment stage, add a Duplicate Search step from the Utilities section and click **Select**. This adds the Duplicate search cases step to the stage and displays the duplicate search cases process flow diagram in the right panel.
- In the **Audit note** field above the flow diagram, add a meaningful name to describe the purpose of the duplicate search.



- Click **Save** on the case life cycle header to add the Duplicate search cases Smart Shape to the Create assessment process.

Verify your work

- Create an Employee Evaluation cases for an employee. Note the manager, position, and start and end dates you entered, which are the potential match criteria.
- Advance the case to the Assess employee step.
- Create another Employee Evaluation case for the employee. Enter different manager and position values but identical start and end dates (two matching criteria).

4. In the Identify assessment step, click **Submit**. The following form is displayed. Note that the Match Score equals 50 — this meets the potential match threshold.

The screenshot shows a user interface for an 'Employee assessment' process. At the top, there's a blue header bar with the text 'Employee assessment' and a 'Next actions' button. Below this is a green header bar with the text '2 days from now'. The main content area has a title 'Display Duplicate Cases' and includes a user profile icon ('SA Senior System Architect') and a due date indicator ('Due in 7 days from now'). A message below the title says 'Please select duplicate case below. If none are duplicates of the current case, please continue to return to process.' A table follows, listing one duplicate case: Case ID EE-5, Created on Sep 20, 2016 1:19:23 PM, and Case Status Open. There are two buttons at the bottom of the table: 'Continue' and 'Resolve as Duplicate'. To the right of the table, a red box highlights the 'Match Score' column, which contains the value '50'.

	Case ID	Created on	Case Status	Match Score
1	<input type="radio"/> EE-5	Sep 20, 2016 1:19:23 PM	Open	50

[Continue](#) [Resolve as Duplicate](#)

DATA MODEL DESIGN

Configuring a localizable list of values

Exercise: Creating work status values

Scenario

Recruiters require a way to quickly differentiate candidate cases during the Interview stage. To clarify reporting on candidate cases, recruiters have asked that a unique, custom status be applied to candidate cases during each step of the Interview stage.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Create three new work status values for use in candidate cases.

Status value	Class	Purpose
Open-Scheduling	TGB-HRApps	For candidate cases that have not been scheduled for an interview.
Open-Interview	TGB-HRApps	For candidate cases that have been scheduled for an interview.
Open-Assessment	TGB-HRApps	For candidate cases that have completed the interview step and are under review to determine whether to extend an offer.

In the candidate case life cycle, apply the work status values to the appropriate steps in the Interview stage.

Status value	Step
Open-Scheduling	Schedule interview
Open-Interview	Interview candidate (Conduct interview)
Open-Assessment	Assess candidate

Tip: Apply the *Open-Interview* status to the *Interview Candidate* assignment within the *Conduct interview* process. This allows you to reuse the Conduct Interview process with the updated status value.

Detailed steps

Create the new work status values

1. Create the three custom status values: Open-Scheduling, Open-Interview, and Open-Assessment.

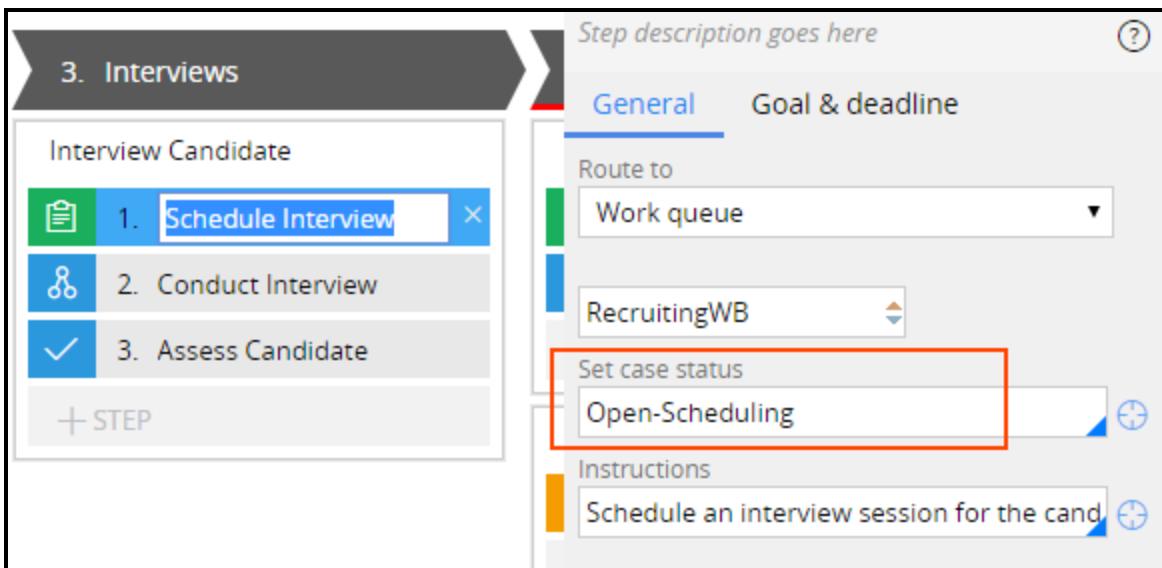
Tip: Use the procedure described in the help topic [Defining a custom status for a case](#). Enter the purpose of each field value In the **Description** field located on the record's History tab, enter the purpose of each field value. The descriptions are useful when deciding whether to reuse an existing work status or create a new one.

2. Click **Designer Studio > Process & Rules > Processes > Status Values** to open the Status Values landing page. The array displays the status values and descriptions.

Open-Assessment	TGB-HRApps	For candidate cases that have completed the interview step and are under review to determine whether to extend an offer.
Open-Interview	TGB-HRApps	For candidate cases that have been scheduled for an interview.
Open-Scheduling	TGB-HRApps	For candidate cases that have not been scheduled for an interview.

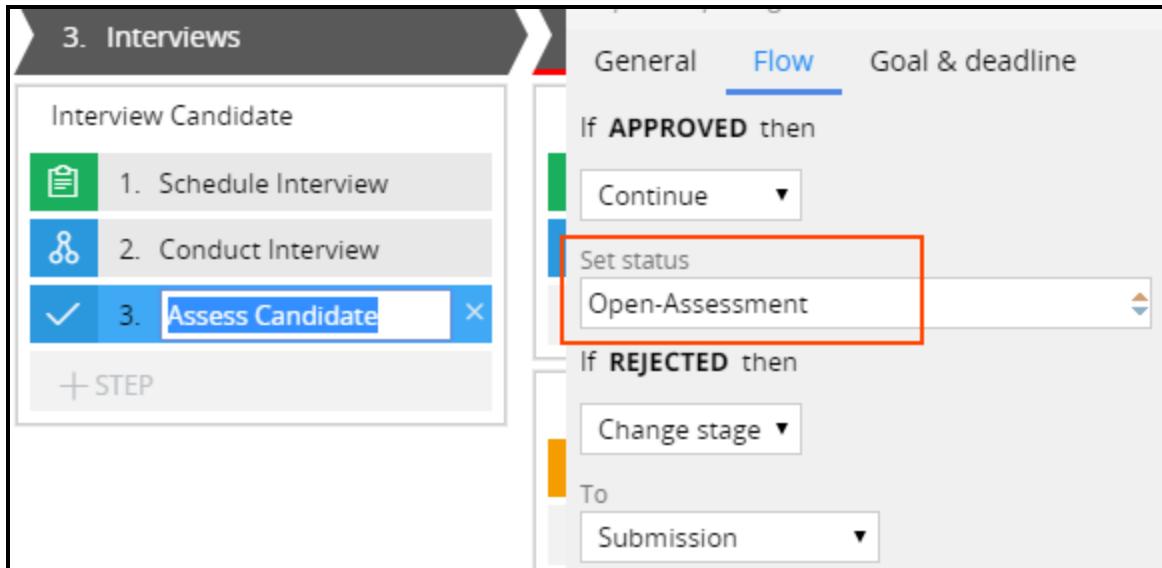
Apply the statuses to the candidate case type

1. Open the Candidate case life cycle.
2. In the Interviews stage, select the **Schedule Interview** step.
3. On the **General** tab, select **Open-Scheduling** in the **Set case status** field.



4. Select the **Assess Candidate** step.

- On the **Flow** tab, in the **Set status** field, select **Open-Assessment**.



- On the case life cycle header, click **Save** to update the Interview Candidate process.
- Select the Interview Candidate process.
- On the **General** tab, click **Open process** to open the Interview Candidate process.
- Right-click the **Conduct Interview** subprocess shape and select **Open flow** to open the Conduct Interview process.
- In the Conduct Interview process, right-click the **Interview Candidate** assignment and select **View properties**.
- In the Assignment properties dialog, in the Advanced section, expand **Assignment details**.
- In the **Work status** field, enter or select **Open-Interview**.

The screenshot shows the 'Assignment details' section of the Assignment properties dialog. It includes fields for 'Harness name' (Perform), 'Instructions' (Conduct interview with the candidate), 'Confirmation note', and a checkbox for 'Do not perform'. Below these is a 'Work status' field containing 'Open-Interview', which is highlighted with a red box. There is also an 'Effort cost' field at the bottom.

13. Click **Submit** to close the Assignment properties dialog.
14. Click **Save** to commit your changes to the flow.

Verify your work

1. Create a candidate case and advance it to the Schedule Interview step. The case status is Open-Scheduling work status.

Submission ✓ Screening ✓ Interviews Offer

3 days from now

Schedule Interview
Schedule an interview session for the candidate

Due in 6 days from now

Assigned interviewer

Interviewer	Location	Start time
-------------	----------	------------

Information **Audit**

Case details

Case ID C-81	Urgency 10	Status Open-Scheduling
-----------------	---------------	---------------------------

2. Advance the case to the Conduct Interview step. The case has an Open-Interview work status.
3. Advance the case to the Assess Candidate step. The case has an Open-Assessment status.

Configuring data access patterns

Exercise: Configuring data access patterns

Scenario

During the Submission stage of the Candidate case type, applicants for an open position at TGB:

1. Review a list of open job positions
2. Select a position to apply for

To help applicants select the correct position during the Submission stage, recruiters ask that the description of an open position be displayed when selected.

Note: During the Screen stage, the recruiter may change the selected position after speaking with the applicant.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

In this exercise, you use the keyed access pattern to display the selected position. Configure the *D_PositionList* data page for keyed access and use it to display the job description.

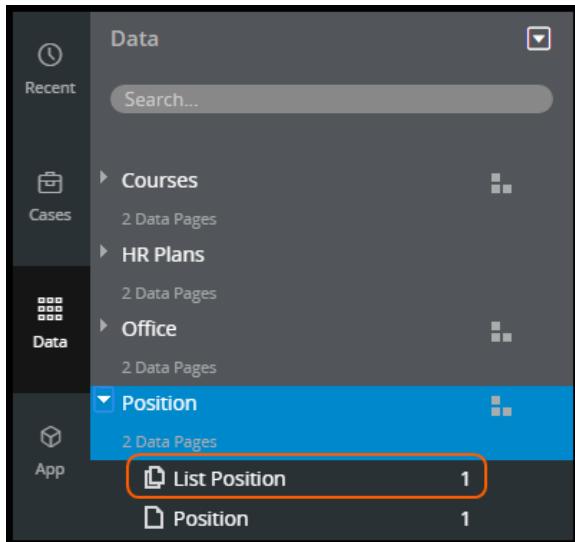
The recommended tasks for completing this assignment are:

- Updating the configuration of the *D_PositionList* data page to access individual position information with the position identifier
- Configuring the *Position* property to refer to the *D_PositionList* data page using the *PositionAppliedFor* property as the key value
- Adding the *JobDescription* field to the *CollectPersonalDetails* section
- Ensuring that the job description is refreshed when the user selects a different position

Detailed steps

Configure keyed data access for the data page

1. In the Data Explorer, open the *D_PositionList* data page.



2. Select **Save As** to save the rule into an unlocked ruleset version so that the rule can be edited.
3. Select **Access pages with user defined keys** to enable keyed page access.
4. Select **.ID** as the **Page list keys**.

The screenshot shows the 'Definition' tab of the Data page configuration interface. On the left, the 'Data page definition' panel includes fields for Structure (List), Object type (TGB-HRAApps-Data-Position), Edit mode (Read Only), and Scope (Thread). On the right, the 'Keyed page access' panel contains two checkboxes: 'Access pages with user defined keys' (which is checked and highlighted with an orange border) and 'Allow multiple pages per key'. Below these checkboxes is a 'PAGE LIST KEYS' input field containing '.ID', which is also highlighted with an orange border.

Refer to the keyed data page

1. In the App Explorer, open the Position property for the Candidate case type.

Tip: Property records are in the Data Model category. From the App Explorer, expand Candidate > Data Model > Property.

Note: If you performed the *Configuring a new application version* exercise, you must save the property to an unlocked version of the HRApps ruleset, which results in a warning that creating a duplicate property definition can negatively impact system performance. You must create a duplicate property definition to update the page source to complete this exercise. To do so, click **OK** to dismiss the warning and create a new version of the Position property.

2. Change the **Data Page** field to point to *D_PositionList* and specify the *PositionAppliedFor* property as the key.

Data access

Manual When data page parameters change and a reference to a non-parameter value is made, the system loads a new data page. Reads from and writes to the property occur on the new data page.
 Refer to a data page
 Copy data from a data page

Data Page: *

TGB-HRApps-Data-Position

Parameters
No parameters to set.

Keys
.ID

Save parameters with this property for access on reopen

Add the position job description as a read-only field

1. Create a new candidate case, then open the Live UI tool to identify the section where the job description will be added.
2. Click the **Crosshair** icon to open the *CollectPersonalDetails_0* section.

Collect Personal Details

Section - CollectPersonalDetails_0

Position applying for*

Select...

Referred by employee?
Referring employee

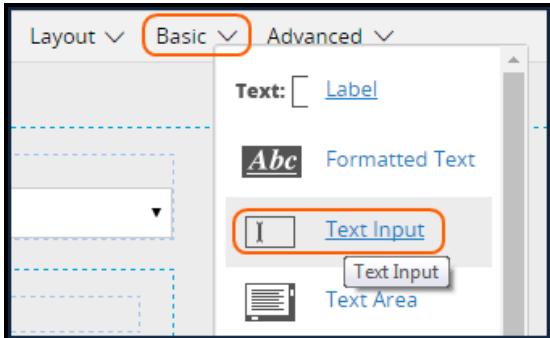
Candidate information

Skin - HRApps

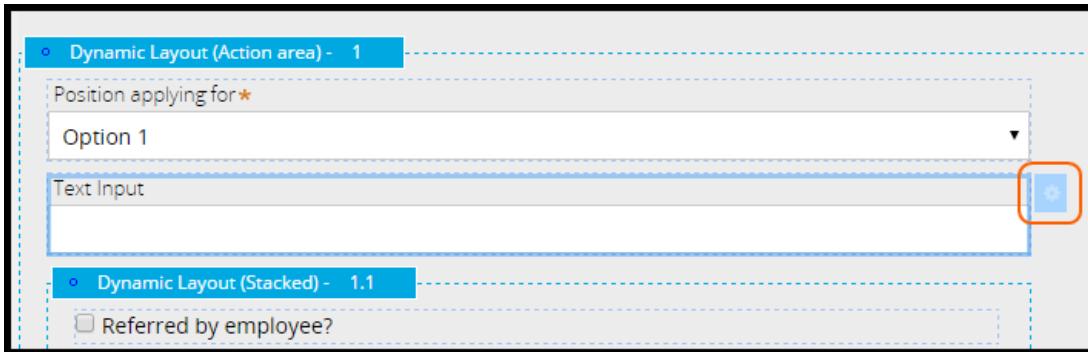
- Dynamic Layout
- Dynamic Layout
- Section in cell - non-auto generated
- Flow Action - CollectPersonalDetails_0
- Section - CollectPersonalDetails_0
 - Dynamic Layout
 - Dropdown - .PositionAppliedFor
 - Dynamic Layout
 - Section include - CollectPersonalDetails_0

3. Click **Save As** to save the rule into an unlocked ruleset version so that it can be edited.

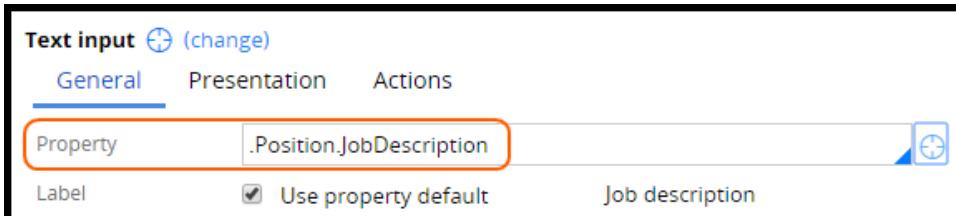
4. Select **Basic > Text input**, and drag the control into the dynamic layout.



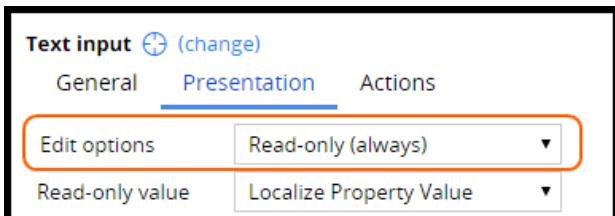
5. Select the control and click the **Gear** icon to open the settings.



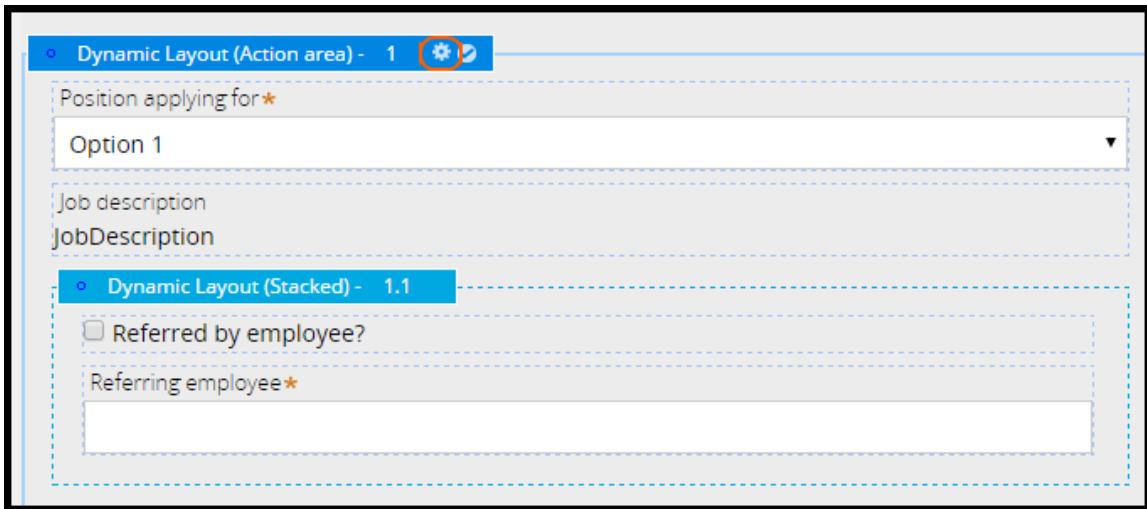
6. On the **General** tab, in the property field, enter **.Position.JobDescription**.



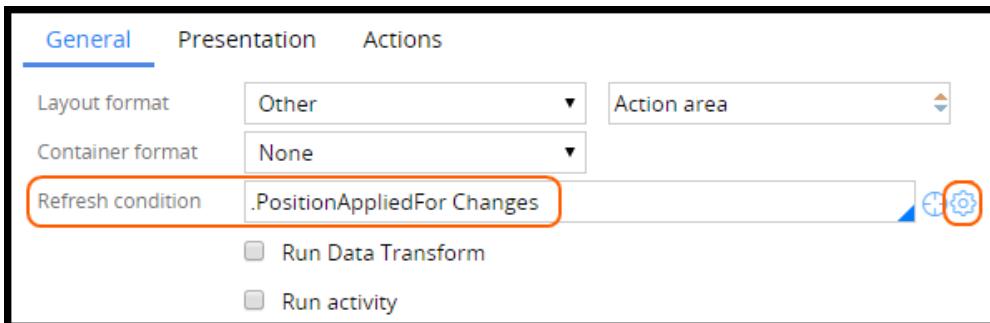
7. On the **Presentation** tab, in the **Edit options** field, select **Read-only (always)**.



8. Select the dynamic layout with the **Position applying for** and **Job description** fields, and click the **Gear** icon to open the settings.

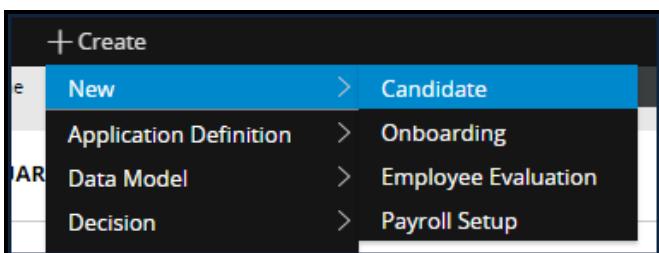


9. Select the **Gear** icon to the right of the **Refresh condition** field to configure a refresh condition to ensure that the description is refreshed when the drop-down is updated.



Verify your work

1. Select **Create > New > Candidate** to create a new candidate case.



2. Under **Position applying for**, select a position.

3. Verify that the **Job description** is displayed.

Submission Screening Interviews Offer

Collect Personal Details

Position applying for *

Marketing Manager

Job description

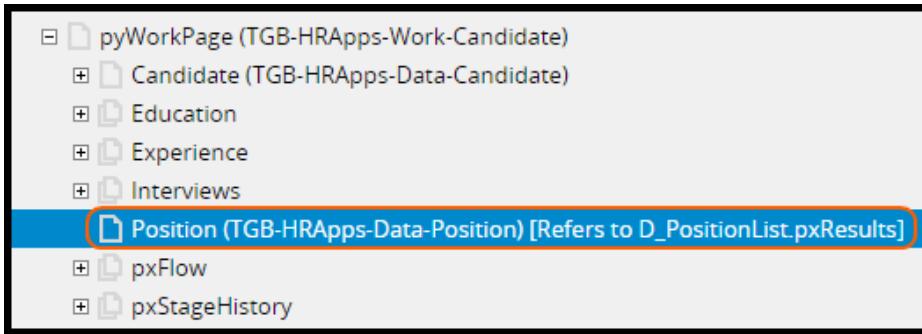
As a marketing manager, implement promotional plans and coordinate the sales managers to improve sales.

Referred by employee?

4. Open the clipboard tool.



5. Open the *Position* property in the *pyWorkPage* page to verify that the property refers to the *D_PositionList* data page. When the user selects a job position, the description of the open position is displayed.



PROCESS DESIGN

Creating organization records

Exercise: Updating assignment routing through Dynamic Class Referencing

Scenario

The Human Resources (HR) department wants the ability to add a new hire to the company payroll. The PegaHR application includes a case type to perform this work. A business architect has determined that the business process is sufficient for TGB's needs at this time. However, the process uses a workbasket that does not match the organization structure at TGB. The Human Resources (HR) department has asked that you create a new workbasket to match the organizational structure at TGB.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Override the default workbasket (*Payroll/ReviewWB*) used in the Review Selection process in the PegaHR application. To do this:

1. Create a new work group for processing Payroll Setup cases with the following details.

Field	Value
Short description	TGB Payroll
Work group name	Payroll@TGB
Manager	Manager@TGB
Default workbasket	default@TGB

2. Create a new workbasket for routing assignments to review Payroll Setup cases with the following details.

Field	Value
Short description	TGB Payroll Workbasket
Workbasket	TGBPayrollWB
Name	TGB

Division	Finance
Unit	Accounting
Work group	Payroll@TGB

3. Update the **AppExtension** data transform in the implementation layer to use the **TGBPayrollWB**.

Detailed steps

Create the work group and workbasket

1. Create a work group with the following details.

Field	Value
Short description	TGB Payroll
Work group name	Payroll@TGB
Manager	Manager@TGB
Default workbasket	default@TGB

Tip: If you do not remember how to create a work group, use the procedure outlined in the help topic [Work Groups](#).

2. Create a payroll workbasket with the following details.

Field	Value
Short description	TGB Payroll Workbasket
Workbasket	TGBPayrollWB
Name	TGB
Division	Finance
Unit	Accounting
Work group	Payroll@TGB

Tip: If you do not remember how to create a work basket, use the procedure outlined in the help topic [Workbaskets](#).

Create a customized version of the AppExtension data transform

1. In Designer Studio, search for AppExtension.
2. Select the AppExtension data transform and copy it into the HRApps ruleset.

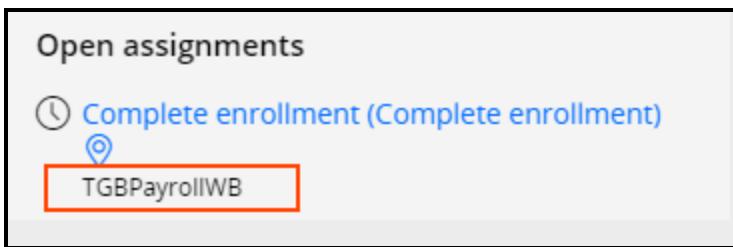
The screenshot shows a search interface with a search bar containing 'AppExtension'. Below the search bar is a table with columns: Type, Name, and Applies to. The table lists several items:

Type	Name	Applies to
Class	History-PegaHR-FW-HRFW-Data-AppExtension	
	PegaHR-FW-HRFW-Data-AppExtension	
Data Page	D_AppExtension	
Data Transform	AppExtension	PegaHR-FW-HRFW-Data-AppExtension
Database Table	History-Pegahr-Fw-Hrfw-Data-Appextension	History-PegaHR-FW-HRFW-Data-AppExtension
Property	AppExtension	PegaHR-FW-HRFW-Work

3. Set the value of the **.PayrollReviewWorkbasket** property to "TGBPayrollWB".
4. Click **Save** to update the data transform configuration.

Verify your work

1. Open the Case Manager portal.
2. Create and complete a **Payroll Setup** case.
3. After completing the case, the workbasket name appears as part of the assignment.



Configuring a cascading approval process

Exercise: Configuring a cascading approval process

Scenario

Before extending an offer to a candidate, the offer must be approved. At TGB, the number of approvals required in the approval process depends on the salary offered to the candidate. All candidate offers must be approved by the hiring manager. Offers where the salary level is greater than or equal to USD80,000 must also be approved by a director, the manager's immediate supervisor. Offers where the salary level is greater than or equal to USD120,000 must be approved by the hiring manager, director, and vice president (VP).

Recruiters have asked that the approval process automatically route candidate cases for approval to the required users.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Reconfigure the approval process for candidate cases.

- For testing purposes, update the reporting structure:
 - Configure the Manager@TGB operator to report to the Director@TGB operator.
 - Configure the Director@TGB operator to report to the VP@TGB operator.
- Create when conditions that test for salary levels so that approvals route to the appropriate users in the reporting structure. The salary levels and users are as listed in the following table:

Salary level (USD)	Approved by
Greater than or equal to 80,000	Manager and director
Greater than or equal to 120,000	Manager, director, and vice president

Note: The manager must approve all candidate offers, regardless of salary. Therefore, you do not need to create a when condition for the manager.

- Add a cascading approval process for director and vice-president approvals.

Detailed steps

Update the reporting structure

The cascading approvals are defined by the reporting structure of the operator initially assigned to the approval. In this section, you configure the reporting structure on the required operator IDs to reflect the chain of approvals required for the application.

Note: In a production environment, the reporting structure would be configured when creating the individual operator ID records. In a development system that uses model operators, you often need to configure the operators manually.

Follow these steps to update the reporting structure:

1. Select **Org & Security > Organization > Operators** to open the Operators landing page for the Designer Studio menu.
2. Under the Operator ID column, click **Manager@TGB** to open the Operator ID record.
3. On the **Work** tab, in the **Reports to** field, select **Director@TGB**.

The screenshot shows the 'Edit Operator ID: Manager' interface. At the top, the operator ID is 'Manager@TGB' and the reporting structure is 'HRApps_Branch_Casca'. The 'Work' tab is active. In the 'Routing' section, there's a 'Reports to' field containing 'Director@TGB', which is highlighted with a red box. Below it, there's a 'Work group' section with a single entry 'default@TGB'.

4. Click **Save**. The manager is now defined as reporting to the director.
5. Update the Director@TGB record so that the operator reports to VP@TGB.

You have configured the operator ID reporting definitions to reflect the TGB reporting structure.

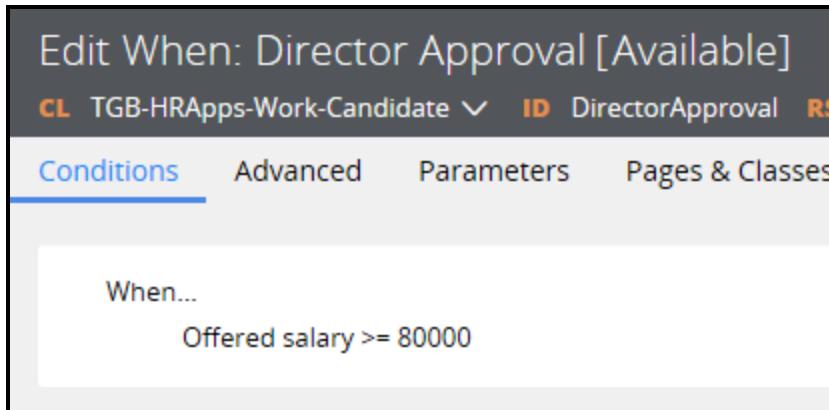
Create the when rules and conditions

The cascading approval consists of two thresholds: USD80,000 and USD120,000. Create two when rules to test when each threshold is met or exceeded.

Note: The thresholds could also be defined using a decision table and configuring the approval by an authority matrix. However, using an authority matrix requires additional effort to identify and configure the additional approvals required. The reporting structure already reflects the approvals required by the application.

Follow these steps to create when rules and conditions:

1. Create a when rule named Director Approval and use a when condition of **Offered salary >= 80000**.



2. Click **Save**. If the amount is equals or exceeds USD80,000, the case is routed to the director.
3. Create a when rule named VP Approval and use a when condition of **Offered salary >= 120000**.

You have created when conditions for testing salary levels.

Configure the cascading approval process

All salary offers for candidates must be approved by the hiring manager. Additional approvals may be required, and are based on the offered salary. Add a cascading approval to the flow for these additional approvals, and add a decision shape to determine when the additional approvals are needed.

Follow these steps to configure the cascading approval process:

1. On the Candidate case life cycle, add an Approve/Reject step under the Manager Approval step.
2. Name the step **Additional Approvals**.
3. On the **General** tab, select **Cascading** in the **Approval flow type** field.
4. In the **Approval based on** field, select **Reporting structure**.
5. In the **Approval to be completed by** field, select **Reporting manager**.
6. Expand the **Approval level** section and select **Custom**.

7. Click the **Update custom levels** link located at the bottom of the form.

The screenshot shows the 'Offer' step configuration. On the left, there's a list of steps: 'Prepare Offer' (with '1. Determine Compens...', '2. Manager Approval', and '3. Additional Approvals'), 'Extend Offer' (with '1. Send Offer Letter'), and 'Accept Offer' (with '1. Candidate Approval'). The 'Additional Approvals' step is currently selected. On the right, under 'General' settings, the 'Approval flow type' is set to 'Cascading'. Below that, 'Approval based on' is set to 'Reporting structure'. Under 'Approval level', the 'Custom' radio button is selected. At the bottom right of this panel, the link '**Update custom levels**' is highlighted with a red box.

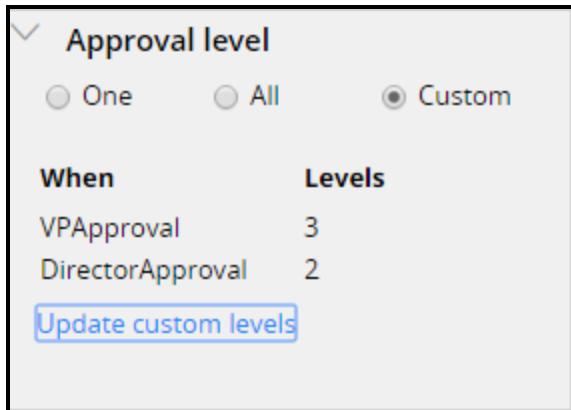
The system displays the **Levels of approval** pop-up dialog.

8. In the **Levels of approval** dialog, click the **+Add custom approval** link.
9. In the **When** field, select **VPAapproval**.
10. In the **Levels of approval** field, enter 3. The form will look like the following image.

The screenshot shows the 'Levels of approval' dialog. It has two main sections: 'When' and 'Levels of approval'. The 'When' section contains a dropdown menu with 'VPAapproval' selected, a '+' button, a text input field containing '3', and a trash can icon. The 'Levels of approval' section is empty. At the bottom, there are 'Cancel' and 'Submit' buttons.

11. Click **Add custom approval** to add a second approval condition.
12. In the **When** field, select **DirectorApproval**.
13. In the **Levels of approval** field, enter 2.

14. Click **Submit**. The Approval level section on the General tab displays the when conditions and levels.



Note: Any case that reaches this approval shape is routed to the manager for review. The case cascades to the director if the USD80,000 threshold is reached or exceeded, and to the VP (the director's manager) if the USD120,000 threshold is reached or exceeded.

15. Click **Save** on the case life cycle header to update the Prepare Offer flow, which now includes the Additional Approvals process.

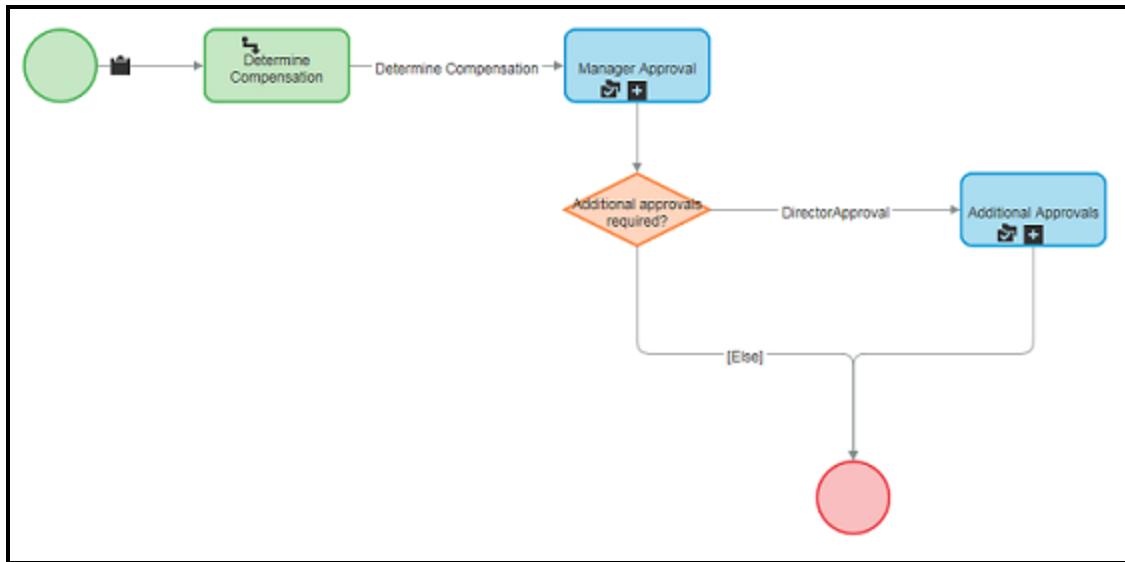
Update the cascading approval process

Update the Prepare Offer flow so that a case where the amount is greater than USD80,000, the case advances to the Additional Approvals process and is routed to the director. If the amount is equal to or greater than USD120,000, the case cascades to the VP.

Follow these steps to update the cascading approval process:

1. Open the Prepare Offer flow and add a decision shape.
2. Open the decision shape properties panel.
3. Enter the shape name **Additional approvals required?**.
4. Set the decision type to **Fork**.
5. Click **Submit** to save your updates.
6. From the decision shape, draw a connector to the Additional Approvals process shape.
7. Open the connector properties panel.
8. Enter the shape name **Director Approval**.
9. Set the condition type to **When** and select **DirectorApproval** in the **When** field.
10. In the **Likelihood** field, select **70**.
11. Click **Submit** to save your updates.
12. Draw a connector from the decision shape to the end shape.
13. Open the connector properties panel and select an **Else** condition type.
14. Click **Submit** to save your update.

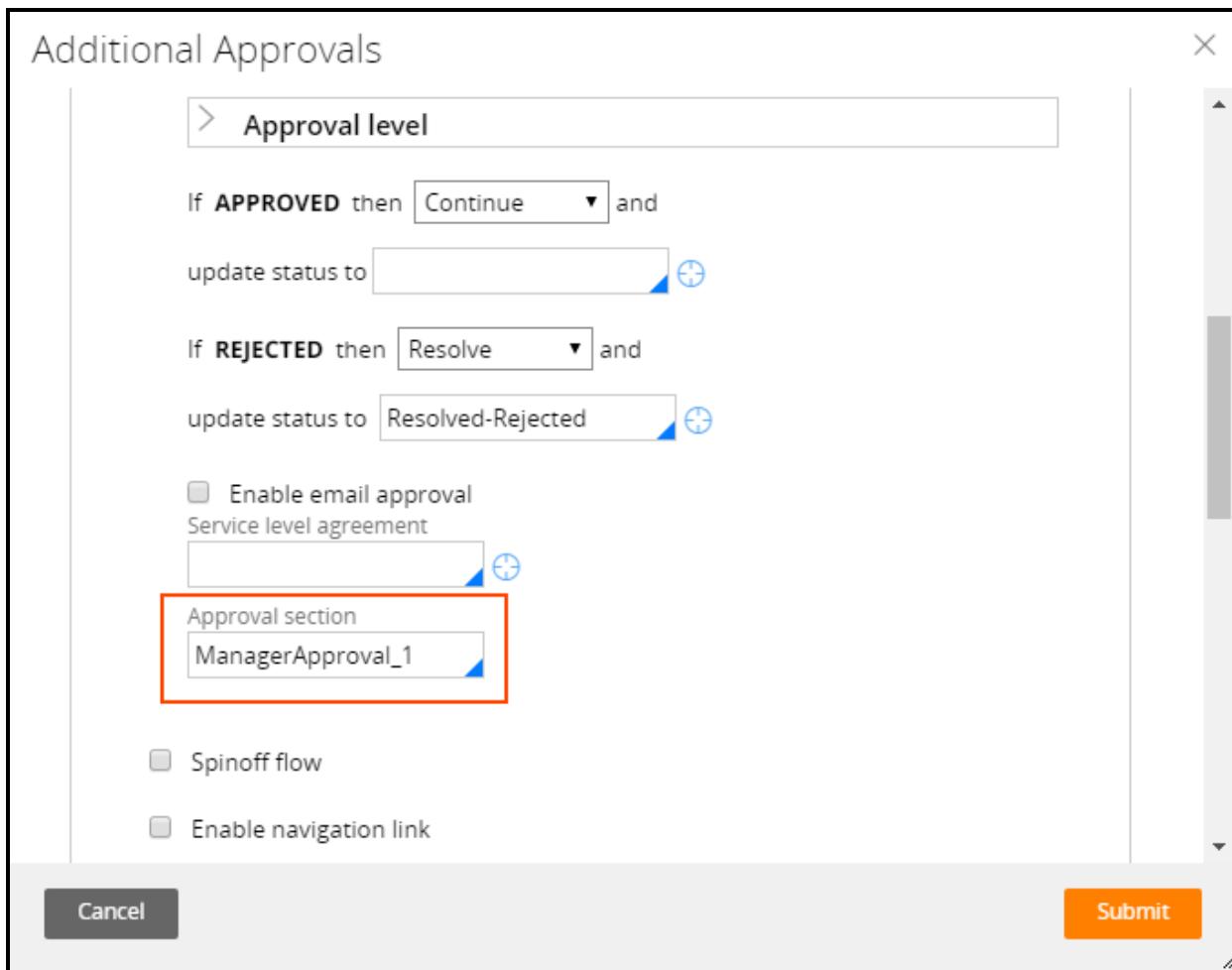
15. When you have completed the previous steps, the process looks like the following diagram:



Add a section for the cascading approval

Add a section to the cascading approval to provide a form for users to complete. For this exercise, reuse the existing ManagerApproval_1 section used for the single approval shape.

1. Open the properties panel for the Additional Approvals shape.
2. In the **Approval section** field, enter or select **ManagerApproval_1** to reuse the Manager approval section record.

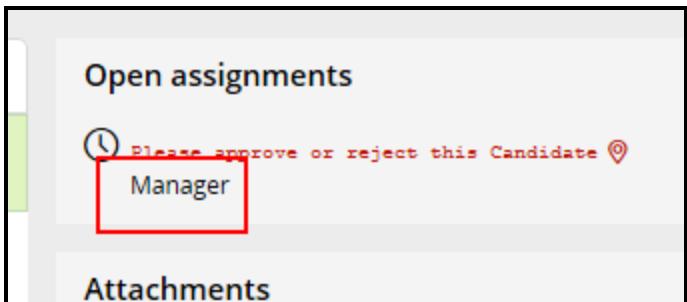


3. Click **Submit** to save your update.
4. Click **Save** to save the flow.

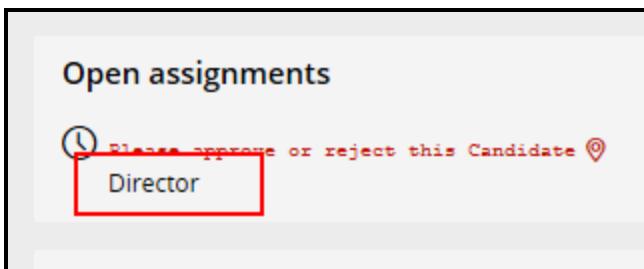
Verify your work

Follow these steps to test the approval configuration:

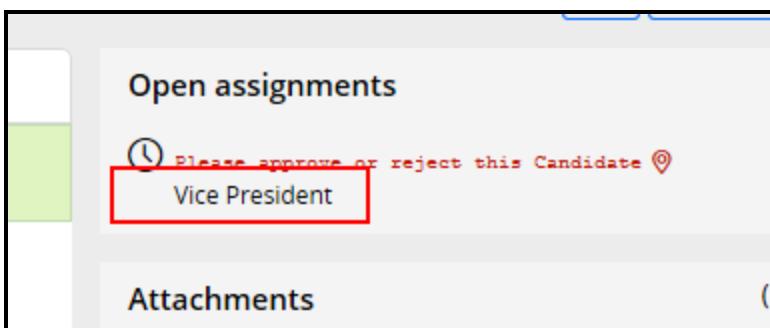
1. Create a Candidate case. You can specify any position.
2. Advance the case until you reach the Determine Compensation form in the Offer stage.
3. In the **Offered Salary** field, enter a value greater than 120,000.
4. Click **Submit**. The next assignment is the original single approval to the manager.
5. Click **Submit**. The flow directs the case to the cascading approval and the Manager is assigned to approve or reject the candidate.



- As Manager, you approve the candidate. The case then appears as an approval assignment for the director.



- As Director, you approve the candidate. The case then appears as an approval assignment for the Vice President.



- Test the Director when condition by creating two new cases. Enter a USD50,000 salary, requiring only manager approval. Enter a USD100,000 salary, requiring both manager and director approvals.

Prioritizing user assignments

Exercise: Prioritizing user assignments

Scenario

Candidate cases are automatically directed to a workbasket for recruiters to review and qualify for an open position at TGB. To improve the quality of applicants for open positions, the Human Resources (HR) department wants employees to refer candidates for open positions whenever possible. To encourage employee referrals, HR wants to prioritize candidates referred by employees.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Create a new workbasket for Candidate cases in which the candidate was referred to TGB by an employee.

Configure the Candidate case type to route a case to the referrals workbasket if the candidate has selected **Referred by employee?**. Replace the workbasket router on the *Conduct phone screen* assignment with a decision table router and configure a decision table to test the value of *.ReferredByEmployee*.

Configure the Recruiter@TGB operator to open cases from the referrals workbasket before opening cases from the RecruitingWB workbasket.

Detailed steps

Create the EmployeeReferralsWB workbasket

1. In Designer Studio, click **Create > Organization > Workbasket** to open the Create Workbasket form.
2. On the Create Workbasket form, in the **Short description** field, enter **Referrals workbasket**.
3. In the **Workbasket** field, enter **EmployeeReferralsWB**.
4. Click **Create and open** to open the workbasket record.
5. On the EmployeeReferralsWB record, in the **Name** field enter or select **TGB**.
6. In the **Division** field enter or select **Administration**.

7. In the **Unit** field enter or select HR.
8. In the **Work group** field enter or select Recruiting@TGB.

Edit Workbasket: Referrals workbasket

ID EmployeeReferralsWB RS HRApps [Edit]

Organization

Name	TGB	<input type="button" value="+"/>
Division	Administration	<input type="button" value="+"/>
Unit	HR	<input type="button" value="+"/>
Work group	Recruiting@TGB	<input type="button" value="+"/>

9. Click **Save** to complete the configuration of the EmployeeReferralsWB workbasket.

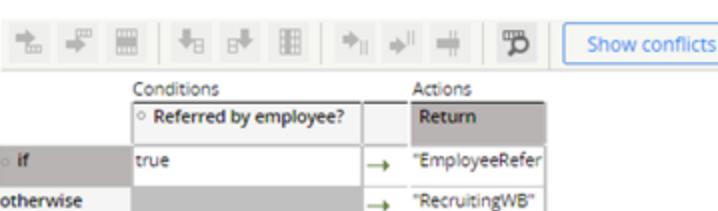
Configure the decision table used to determine routing

1. Create a decision table named ToEmployeeReferralWB in the Candidate case type.

Decision Table Record Configuration

Label *	ToEmployeeReferralWB
A short description or title for this record	
Apply to *	TGB-HRApps-Work-Candidate
View all	

2. Add a condition to test the value of `.ReferredByEmployee`. If the value of `.ReferredByEmployee` is true, return "EmployeeReferralsWB" as the result. Otherwise, return "RecruitingWB" as the result.

Decision Table: ToEmployeeReferralWB [Available]													
CL	TGB-HRApps-Work-Candidate	ID	ToEmployeeReferralWB	RS	HRA								
Table	Results	Parameters	Pages & Classes	Specifications									
 <table border="1"> <thead> <tr> <th>Conditions</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>Referred by employee?</td> <td>Return</td> </tr> <tr> <td>if true</td> <td>→ "EmployeeRefer"</td> </tr> <tr> <td>otherwise</td> <td>→ "RecruitingWB"</td> </tr> </tbody> </table>						Conditions	Actions	Referred by employee?	Return	if true	→ "EmployeeRefer"	otherwise	→ "RecruitingWB"
Conditions	Actions												
Referred by employee?	Return												
if true	→ "EmployeeRefer"												
otherwise	→ "RecruitingWB"												
Show conflicts													

Modify the *Conduct phone screen* step to use the decision table to determine routing

1. Open the *Candidate* case type in the Case Designer and select the **Conduct phone screen** step.
2. Configure the *Conduct phone screen* step to use the *ToEmployeeReferralWB* decision table to determine which work basket to route to.

General [Goal & deadline](#)

Route to
Custom

Assignment type
Work queue

Router
ToDecisionTable

Parameters

DecisionTable*

ToEmployeeReferralWB

Add the *EmployeeReferralsWB* workbasket to the *Recruiter@TGB* operator ID

1. Open the *Recruiter@TGB* operator ID record.
2. Add the *EmployeeReferralsWB* as the first workbasket, and the *RecruitingWB* as the second workbasket.

Workbasket	Urgency Threshold
EmployeeReferralsWB	
RecruitingWB	

Verify your work

1. Create a Candidate case, and select the **Referred by employee?** check box.

Candidate (C-91)

Submission Screening Interviews Offer

Collect Personal Details
Collect information about the candidate

Selected position

Position applied for*
Director

Referred by employee?
Referring employee*
Designer@TGB

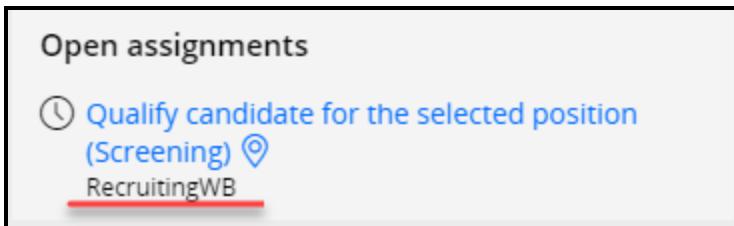
Note: Remember to complete all required fields on this form before submitting.

2. Advance the case through the next three steps in the Submission stage. Completing the fields in any of these screens is optional for testing purposes.
3. When the case advances to the *Screening* stage, confirm the routing of the assignment is set to the **EmployeeReferralsWB**.

Open assignments

Qualify candidate for the selected position
(Screening)
EmployeeReferralsWB

4. Create a new candidate case, but do not select the check box for *Referred by employee?* this time.
5. Advance the case to the *Screening* stage and confirm the routing of the assignment is set to **RecruitingWB**.



Test your changes from the Case Worker portal (optional)

The following table provides the credentials you need to complete this portion of the exercise.

Role	Operator ID	Password
HR Recruiter	Recruiter@TGB	rules

1. Using the credentials from the table above, log in to the Case Worker portal.
2. Create several Candidate cases where the candidate is referred by an employee, and several Candidate cases where the candidate is not referred by an employee.

Note: As you create each case, make a note of the case ID numbers, and remember to advance each case to the Submission stage.

3. From the Case Worker portal, click **Next assignment** to verify that assignments are pulled from the referral workbasket first.

Note: Use the case ID numbers to confirm that the employee-referred cases are displayed before the non-referral cases.

Tip: If you have at least two browsers on your computer, use a different browser to log in to the Case Worker portal.

Delegating rules to business users

Exercise: Delegating a business rule

Scenario

As part of the candidate recruiting process, recruiters should schedule the candidate for an initial interview within two business days of qualifying a candidate. However, the candidate must be scheduled for an initial interview no later than five business days after qualifying.

Recruiters have stated concerns regarding these performance expectations. The vice president (VP) of Human Resources (HR) has decided that these expectations should be monitored and adjusted as needed.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules
Business user	HRManager@TGB	rules

Your assignment

Delegate the service level agreement (SLA) for the Schedule Interview assignment to an access group for HR employees.

- Rules delegated to the HR department are organized in a ruleset named *HR_DelegatedRules*.
- The access group responsible for managing delegated rules is *HRApps:DelegatedUsers*. The short description reads *HR users managing delegated rules*.

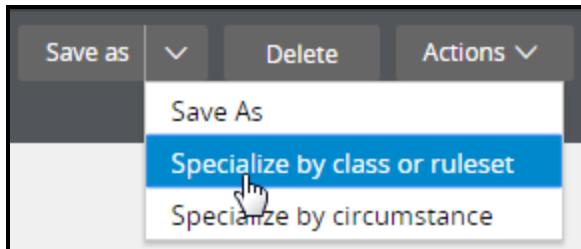
Detailed steps

Follow these steps to delegate a rule.

- Open the **ScheduleInterview** SLA record in TGB-HRApps-Work-Candidate.

Tip: SLA records are in the Process category. From the App Explorer, expand Candidate > Process > Service Level Agreement.

- Click **Save as > Specialize by class or ruleset** to create a copy of the *ScheduleInterview* SLA record.



- In the Context section of the new record, select **Production Rulesets** and then select the *HR_DelegatedRules* ruleset from the **Add to ruleset** drop-down list.

Note: Depending on the customer, creating the rulesets used for delegating rules may fall to a system administrator, an LSA, or others with detailed knowledge of the application structure. For this exercise, the ruleset was prepared in advance as noted in the Assignment details.

- Click **Create and open** to open the new SLA record.
- Click **Save** to save the new record in the *HR_DelegatedRules* ruleset.
- From the **Actions** menu, select **Delegate**.
- In the **Delegate to access group** field, select **HR users managing delegated rules** from the list of available options.
- In the **Title** field, enter **SLA for candidate interviews**.
- In the **Detailed description** text area, enter **This SLA monitors the milestones (goal, deadline, and passed deadline) for scheduling candidate interviews. Changes to any of the milestones will affect in-flight transactions if the case has not reached a milestone.**

The dialog box is titled "Delegate Service Level Agreement". It contains the following fields:

- What will the end user be able to edit?**: A note saying "Select one:" followed by a radio button next to "Manage SLA".
- Delegate to access group***: A dropdown menu set to "HR users managing delegated rules".
- Title ***: An input field containing "SLA for candidate interviews".
- Detailed description**: A text area containing the following text: "This SLA monitors the milestones (goal, deadline, and passed deadline) for scheduling candidate interviews. Changes to any of the milestones will affect in-flight transactions if the case has not reached a milestone."

At the bottom are two buttons: "Cancel" on the left and "Delegate" on the right.

- Click **Delegate** to complete the rule delegation process.

11. Click **Save**. The *ScheduleInterview* SLA record is now delegated.

Verify your work

Verify the delegated SLA record can be edited from the Case Manager portal

The following table provides the credentials you need to complete this portion of the exercise.

Role	Operator ID	Password
Business user	HRManager@TGB	rules

1. Using the credentials from the table above, log in to the Case Manager portal.
2. From the left-side navigation panel, click **Configuration**. Rules delegated to the user are displayed in the workspace.

The screenshot shows the 'Application Configuration' page with a header 'Application Configuration' and a search bar 'Filter by keyword'. Below the header, it says 'Displaying 2 of 2 items'. There are two entries listed:

- New hire welcome email**: This email is sent to all new hires. Changes to the email message will not be resent. An 'Edit' button is available.
- SLA for candidate interviews**: This SLA monitors the milestones (goal, deadline, and passed deadline) for scheduling candidate interviews. Changes to any of the milestones will affect in-flight transactions if the case has not reached a milestone. An 'Edit' button is available.

Note: If you have completed the Customizing a portal exercise, this menu item is now named **Manage Delegated Rules**.

3. Click **Edit** for the *SLA for candidate interviews* delegated rule.

4. Add an escalation action to notify the assignee if the goal is missed.

Goal

Days hh:mm
2 0:0 Calculate using business days

Increase urgency by
0

Perform actions

Notify assignee (+) Add escalation action

5. Save your changes and log out of the Case Manager portal.

Verify changes made from the Case Manager portal are reflected in the SLA record

The following table provides the credentials you need to complete this portion of the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

1. Log in to Designer Studio using the credentials in the table above.
2. Using the App Explorer, open the *ScheduleInterview* SLA.
3. Confirm the SLA record opened is in the *HR_DelegatedRules:01-01-01* ruleset.

Edit Service Level Agreement: Schedule interview [Available]

CL TGB-HRApps-Work-Candidate ▾ ID ScheduleInterview RS HR_DelegatedRules:01-01-01

The changes made to the SLA record are displayed in the form.

Service level definitions

Calculate service levels
Interval from when assignment is ready ▾

Goal

Days	Hrs	Mins	Secs
2	0	0	0

Time interval starts when the associated assignment (or work item) is created

Only calculate using business days

Amount to increase urgency
0

Actions	When
Notify Assignee	

[Edit](#) [Delete](#)

[+](#)

Configuring parallel processing

Exercise: Configuring parallel processing

Scenario

At TGB, candidates for an open position are interviewed by between three to six current employees. The interview process for candidates provided in the HRApps application only allows for one interviewer. The Human Resources (HR) department has requested that this process be updated to support more than one interviewer for a candidate.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

The following list includes recommended tasks for completing this assignment:

- Creating a page list property to hold the list of interviewers, and adding an empty entry to the list
- Updating the schedule interview form to allow the user to enter a list of interviewers, rather than a single interviewer
- Replacing the *ConductInterview* subprocess with a Split For Each shape, and configuring the Split For Each shape to call the *ConductInterview* flow for each interviewer in the list
- Updating the Assess Candidate form to allow the user to view the feedback from all interviewers

Note: Because the *ConductInterview* flow is located in the *Interview* data class, changing from page to page list has no impact on the flow rule.

Detailed steps

1. In the Candidate case type, create a page list property of data type **TGB-HRApps-Data-Interview** named **Interviews**.

Tip: To add a property, right-click the **Candidate** case type in the App Explorer and select **Create properties** to open the Data model for the case type.

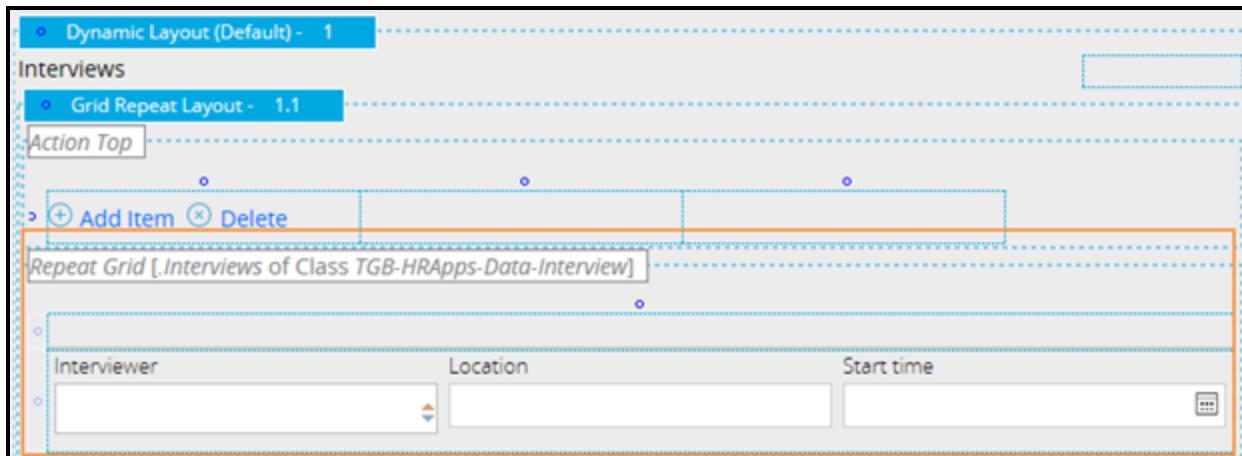
2. In the Candidate case type, add a default entry to the list in the *pySetFieldDefaults* data transform. This ensures that the list contains an empty entry when the screen is displayed.

Tip: Initialize an element in the first item in the list (for example, **.Interviews[1].Location = ""**) to add a default entry.

3. Update the *ScheduleInterview_0* section to use a repeat grid sourced by the *Interviews* property, allowing the user to add and remove interviews. In the repeat grid, include the existing *InterviewSchedule* section to display the interview information in the grid.

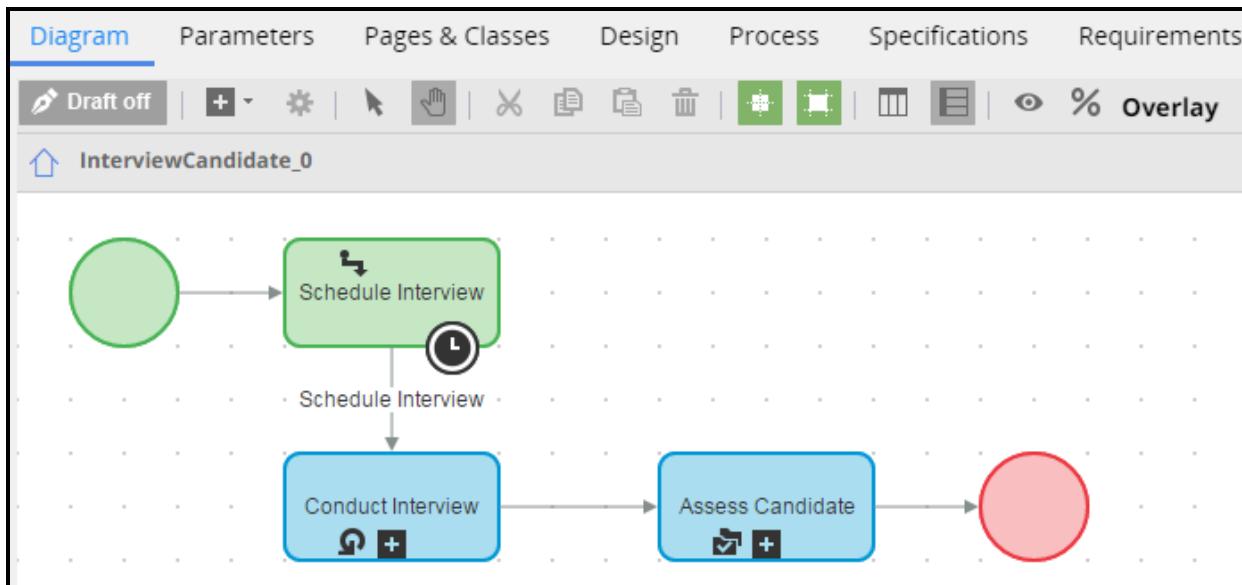
Tip: In the Case Explorer, select the **Schedule Interview** step and click **Configure view** to open the *ScheduleInterview_0* section.

Important: Remember to set the edit mode on the repeat grid so users can update the contents of the page list.



4. In the *InterviewCandidate_0* flow, replace the Conduct Interview process with a Split For Each shape to create a subprocess for each interviewer.

Tip: In the Case Explorer, select **Interview Candidate** and click **Open process** to open the flow.



5. Specify **Interviews** as the page property and use the existing **ConductInterview** subflow to configure the Split For Each shape.

Split For Each properties X

Split For Each: **ConductInterview**

Start a set of the same subprocess to be worked on in parallel for each item in the given list.

Split For Each details
What will this split for each shape be doing primarily?

Join

Page property*
 +

Class*
 +

Filter flow by

Flow name*
 +

When

6. In the *AssessCandidate_1* section add a repeating grid sourced by the *Interviews* property, allowing the user to see the feedback from all interviews. Include the existing *InterviewFeedback* section to display the feedback in the grid.

7. Update the *AssessmentRating* declare expression to calculate the average of all interview ratings.

Tip: Use the **Average of** computation type to calculate the average value for a property in a page list.

Verify your work

1. Create a new candidate case and move to the Interview stage. Verify that an empty entry in the interviewer list exists.

Candidate (c-124)

Submission ✓ Screening ✓ Interviews Offer

2 days 3 hours from now

Schedule Interview Schedule an interview session for the candidate	Due in 7 days from now
---	------------------------

Interviews

[+ Add Item](#) [Delete](#)

Interviewer	Location	Start time
Director@TGB	Cambridge	9/29/2016 10:00 AM

[Cancel](#) [Save](#) [Submit](#)

2. Add two interviewers to the list, and click **Submit**.

Interviews

[+ Add Item](#) [Delete](#)

Interviewer	Location	Start time
Director@TGB	Cambridge	9/29/2016 10:00 AM
Manager@TGB	Cambridge	9/29/2016 11:00 AM

3. Verify that two assignments were created.

Candidate (c-124)

Submission ✓ Screening ✓ Interviews Offer

Thank you! The next step in this case has been routed appropriately.

Information	Audit	Open assignments
		Interview candidate Director
		Interview candidate Manager

4. Complete the two Interview candidate assignments.

5. Verify that the assessment rating was computed correctly in the approval screen.

The screenshot shows the 'Approval' step in a recruitment process. At the top, there are four tabs: 'Submission ✓' (green), 'Screening ✓' (green), 'Interviews' (blue, currently selected), and 'Offer'. Below the tabs, the section title is 'Approval' with the sub-instruction 'Please approve or reject this Candidate'. There are two sections, each representing an interviewer's rating:

Interviewer	Interpersonal skills rating	Average rating
Director@TGB	Below average	2.00
Interviewer rating	Below average	
Below average		

Feedback: _____

Interviewer	Interpersonal skills rating	Average rating
Manager@TGB	Above average	4.00
Interviewer rating	Above average	
Above average		

Feedback: _____

Improving the user experience with screen flows

Exercise: Configuring a screen flow

Scenario

During the benefits enrollment process, users must select a medical coverage plan, dental coverage plan, and vision coverage plan. Business partners in the Human Resources (HR) department report that new employees often rethink their choices. As a result, HR business partners have introduced a requirement to allow users to step back and forth when selecting an option for each type of coverage.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Reimplement the Insurance Selection process as a screen flow to allow a user to step back and forth in the process.

Detailed steps

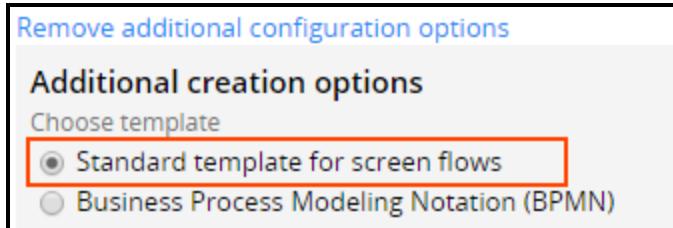
Configure the screen flow

1. Create a new flow record named **Select Health Insurance Coverage**.

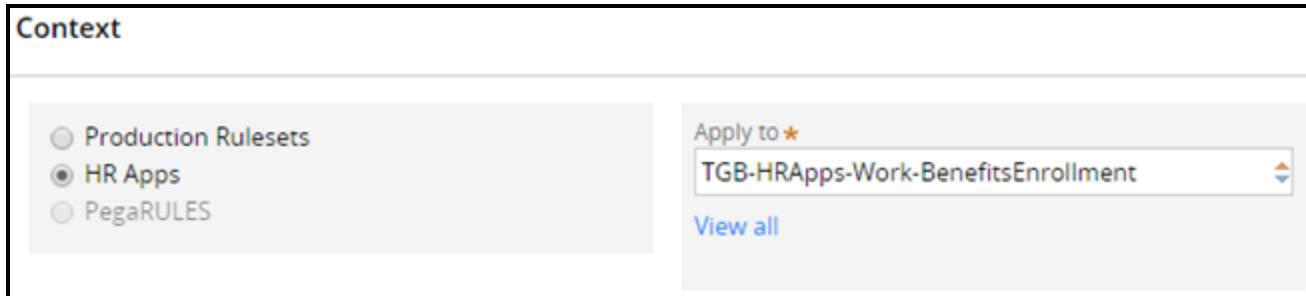
Tip: Flow records are in the Process category. From the App Explorer, expand **Benefits Enrollment > Process > Flow**. Right-click **Flow** and click **Create**.

2. On the new *Create Flow* form, click **View additional configuration options**.

3. In the *Additional creation options*, select the **Standard template for screen flows** option.



4. Confirm that the **Apply to** context is set to **TGB-HRApps-Work-BenefitsEnrollment**.

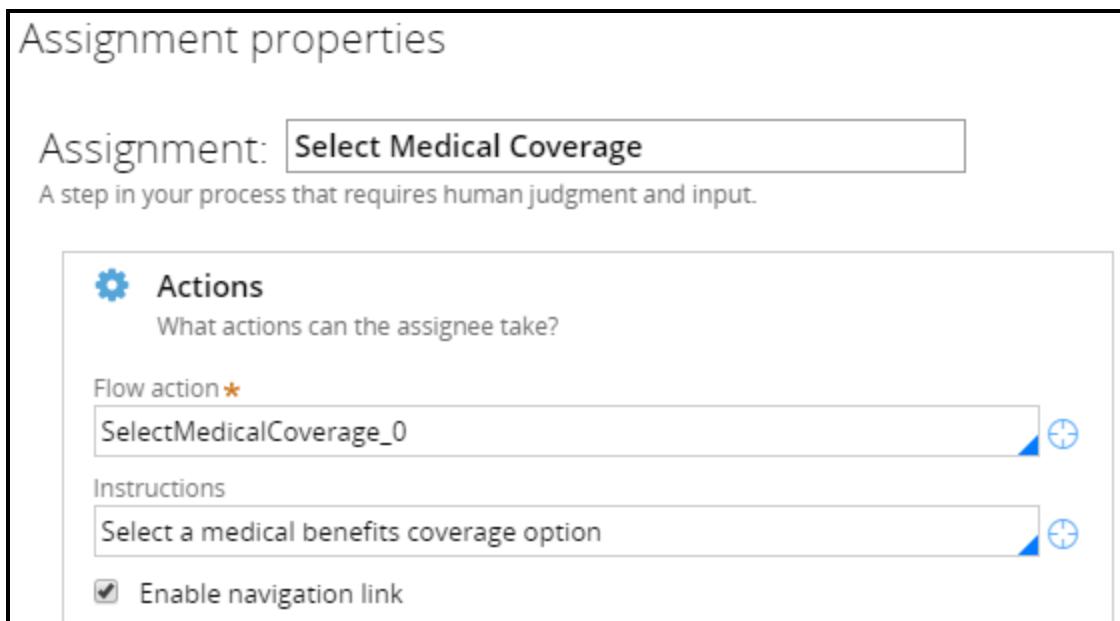


5. Click **Create and open**. The flow record is opened in the work area with a single assignment named *Perform Task*.

6. Open the properties panel for the *Perform Task* assignment shape.

Tip: To open the properties panel of any shape in a flow, you can double-click the shape, or right-click on the shape and select **View Properties**.

7. Change the name of the assignment to **Select Medical Coverage**.
8. In the **Flow action** field, from the drop-down list, select **SelectMedicalCoverage_0**.
9. In the **Instructions** field, enter **Select a medical benefits coverage option**.

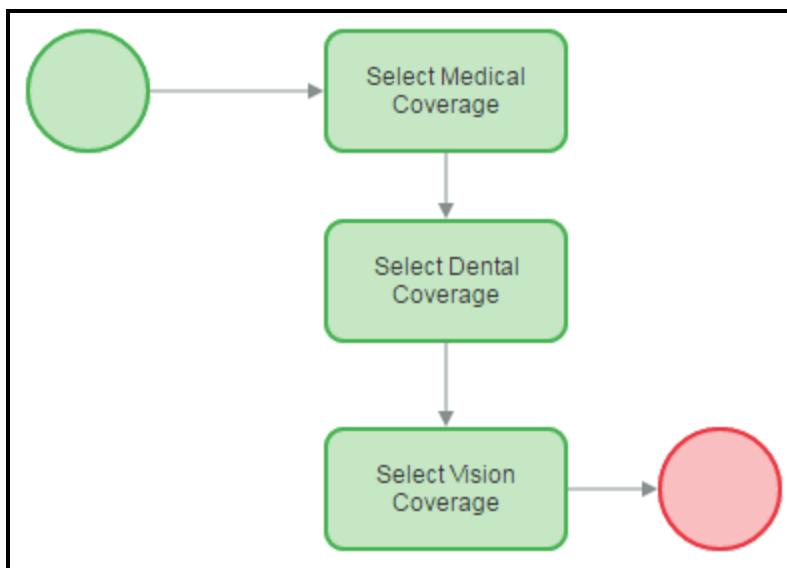


Note: Do not type a period (".") in the **Instructions** field. In Pega, the period is a reserved character, used to indicate a property. If you enter a period in the **Instructions** field, you must encapsulate the contents of the field with quotation marks. Otherwise, Pega returns an error.

10. Click **Submit** to save your changes to the assignment shape and close the properties panel.
11. Add two additional assignment shapes to the flow, and define the properties for each shape according to the table below.

Tip: Remember to click **Submit** after editing the properties on an assignment shape.

Assignment Name	Flow Action	Instructions
Select Dental Coverage	SelectDentalCoverage_0	Select a dental benefits coverage option
Select Vision Coverage	SelectVisionCoverage_0	Select a vision benefits coverage option



12. Save your changes.

Add the screen flow to the case type

1. Open the *pyDefault* case type record for the Benefits Enrollment case type.

Tip: Case type records are in the Process/Case Type category. From the App Explorer, expand **Benefits Enrollment > Process > Case Type**.

2. On the **Stages** tab, select the **Insurance Selection** stage.

The screenshot shows the 'Stages' tab of a configuration interface. The 'Primary Stages' section contains three stages listed vertically: 'Covered Parties', 'Insurance Selection', and 'Review Selections'. The 'Insurance Selection' stage is highlighted with a blue background, indicating it is selected. Below the stages is a button labeled '+ add primary stage'.

3. In the **Automatically launched processes** section, enter **SelectHealthInsuranceCoverage** in the **Process** field.

The screenshot shows the 'Automatically launched processes' section. It displays a single process configuration with the following settings:

- Process:** SelectHealthInsuranceCoverage
- Start when:** Always
- Start Step:** Start upon stage entry
- Service level:** (empty field)

4. Click **Save** to update the case type record.
5. Open the **Benefits Enrollment** case type in the Case Designer. The screen flow is displayed as a process in the Insurance Selection stage.

The screenshot shows the 'Edit case type: Benefits Enrollment' screen with the 'Life cycle' tab selected. The process flow is visualized as a sequence of steps:

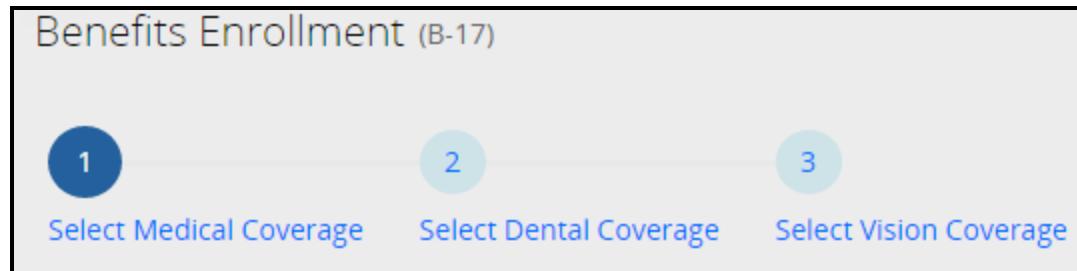
- 1. Covered Parties**: Contains 'Select Participants' with sub-steps '1. Confirm Employee D...' and '2. Identify Dependents', and a '+ STEP' button.
- 2. Insurance Selection**: Contains 'Select Health Insura...' with sub-steps '1. Select Medical Cover...', '2. Select Dental Covera...', and '3. Select Vision Coverage'.
- 3. Review Selections**: Contains 'Review Selections' with sub-step '1. Review Selections', and a '+ STEP' button.

Verify your work

Run the Benefits Enrollment case type to test your changes.

1. On the *Confirm Employee Details* screen, optionally select a *Marital Status* and then click **Submit**.
2. On the *Identify Dependents* screen, optionally enter *Dependents* and then click **Submit**.

The insurance selection process is displayed as a screen flow.



Adding attachments

Exercise: Adding attachment categories to cases

Scenario

Government regulations require that an employee verify eligibility for employment before starting a job. To provide this verification, an employee must submit two types of documents. The first type verifies an employee's identity, and the second type verifies that the employee is eligible for employment. Business partners in the Human Resources (HR) department must certify that the employee has provided both types of documents before completing the onboarding process.

To improve compliance with government regulations, HR requests that a copy of both documents be attached to an employee's onboarding case. The case audit trail must indicate the document category (employee identity and employee eligibility). The documents are provided as either PDF documents or scanned (TIFF) image files. The form should prompt the user to attach the document when the user selects the document type. Users are allowed to create and edit attachments, but are not allowed to delete them. To ensure compliance, the user should not be able to submit the form unless the documents are attached.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

You have the following four requirements:

- Identify the document categories for the two document types by creating the following two attachment categories:
 - Identity verification document
 - Employment eligibility document
- Ensure that users attach documents in the correct file format. Define the allowed formats — files and scanned TIFF images — in the attachment categories.
- Use two standard When rules to configure access to the Create, Edit, Delete Own and Delete Any operations:
 - *Always*: Create, Edit
 - *Never*: Delete Own, Delete Any

- Update the user interface so that users can attach the documents when verifying the documents in the Review verification documents form. To meet this requirement, you must do the following things:
 - Update the Review verification documents section to add a button for each attachment category. Configure each button to open the AddAttachments flow action in an overlay. The user adds the document file in the overlay.
 - To prompt the user, the form displays the button when a user selects the check box that indicates the document is provided.
- Ensure that users have attached both documents. Add a validate rule to the Review verification documents flow action. Configure the validate rule to use the condition *A [attachment category] is [attached/not attached] to the current case* to verify that an attachment in each category has been added to the case.

Detailed steps

Create two attachment categories and attachment types

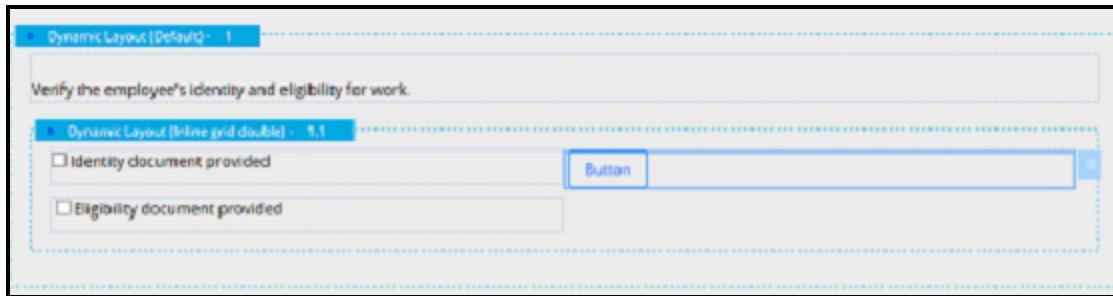
1. In the Application Explorer, right-click **Onboarding > +Create > Process > Attachment Category**.
2. Name the new attachment category **Employment eligibility document** and create the attachment category.
3. On the **Security** tab, in the Access control list by When Rule section add the when rules *Always* and *Never*.
4. For the *Always* when rule, click the **Create** and **Edit** checkboxes.
5. For the *Never* when rule, click the **Delete Own** and **Delete Any** checkboxes.
6. On the **Availability** tab, click the **File** and **Scanned document** checkboxes.
7. Click **Save**. You have created an eligibility document attachment category named Employee eligibility document, which can be used for attaching text files or scanned image files.
8. Repeat steps 1 through 7 to create the **Identity verification document** attachment category.

Add attachment buttons to the user interface

Note: The visible when conditions reference the *.IdentityDocument* and *.EligibilityDocument* properties associated with the check boxes on the form.

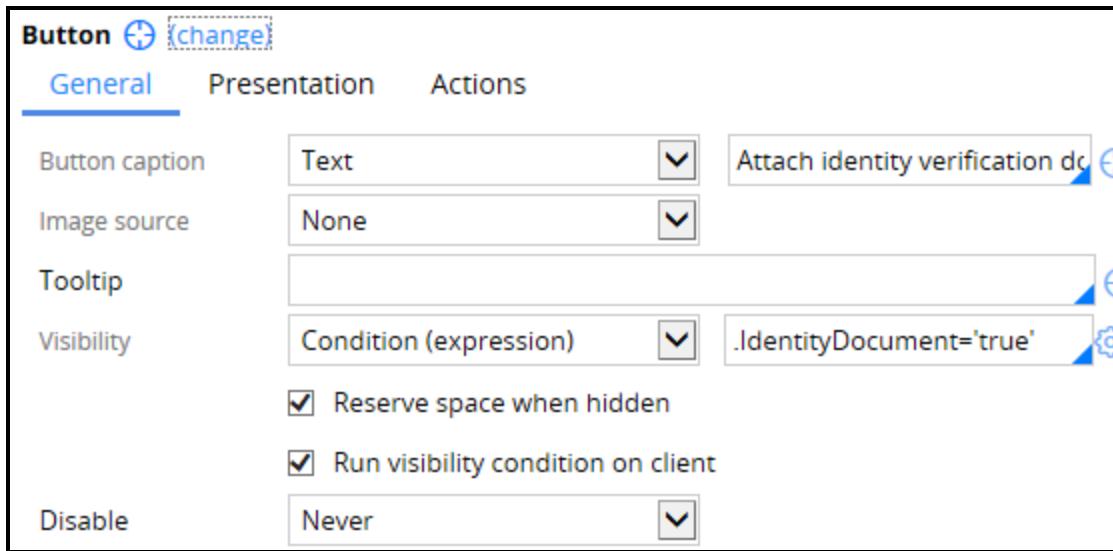
1. In the Applications Explorer, select **Onboarding > User Interface > Section > ReviewVerificationDocuments_0**.
2. On the **Design** tab, on the Dynamic Layout (Default) 1.1 header, select the **Gear** icon to open the Dynamic layout properties panel.
3. In the **Layout format** field, select **Inline grid double**.
4. Click **Submit** to save your update and close the panel.

- Drag a button control from the Basic palette to the dynamic layout and place the button to the right of the **Identity document provided** check box. The button is displayed in a new cell next to the check box.



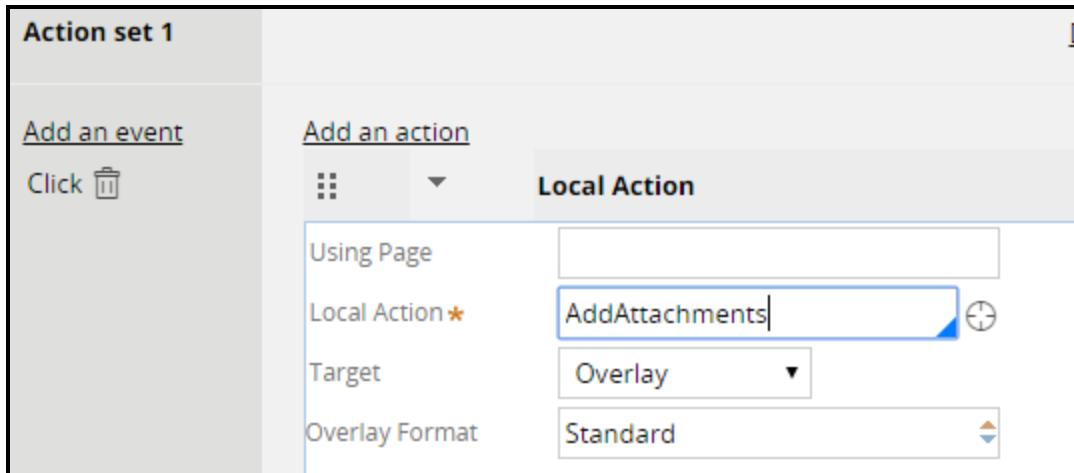
- Click the **Gear** icon next to the button to open button properties panel.
 - On the properties panel, on the **General** tab, select **Text** in the **Button caption** field and enter **Attach identity verification document** in the empty field.
 - In the **Visibility** field, select **Condition (expression)** and enter the condition **.IdentityDocument='true'**.
- Note:** The condition references the **.IdentityDocument** property used associated with the **Identity document provided** check box on the form.
- Click the **Reserve space when hidden** check box so that the **Eligibility document provided** check box on the form does not shift position when the button is concealed.

The completed properties panel looks like the following diagram.



- Open the **Actions** tab.
- Add a **Click** event.
- Add a **Local Action** action.
- In the **Local Action** field, select **AddAttachments**.
- In the **Target** field, select **Overlay**.

15. In **Overlay Format** field, select **Standard**. The completed action set looks like the following diagram.



16. Click **Submit** to save your updates. When a user selects the **Attach identity verification** check box on the form, the form displays the **Attach identity verification document** button for making the attachment. When the user clicks the button, the form displays an overlay in which the user attaches a file.
17. Repeat steps 5 through 16 in order to configure a button named **Attach employee eligibility document** next to the **Eligibility document required** check box. Configure a visible when condition to display the button only when the value of *.EligibilityDocument* is true.

Add a validate rule to the Review verification documents flow action

- In the Cases Explorer, open the Onboarding case life cycle.
- In the Verification stage, open the Verify Employee Documentation process.
- Right-click the Review Verification Documents connector and select **Open Flow Action** to open the Review verification documents flow action.
- On the rule form, click the **Validation** tab.
- Enter a validate rule named **VerifyDocumentsAttached** to create a new validate rule.
- Use the following information to create the validation logic. Note that the conditions include the attachment categories *IdentityVerificationDocuments* and *EmploymentEligibilityDocument*. These are evaluated when validation is performed.

Property	Condition	Message
.IdentityDocument	IF A IdentityVerificationDocument is not attached to the current case THEN display message:	Please attach a document verifying identity
.EligibilityDocument	IF A EmploymentEligibilityDocument is not attached to the current case THEN display message:	Please attach a document verifying eligibility for employment

Tip: Use the procedure described in the help topic [Validate form — Completing the Validate tab](#).

The completed validate rule looks like the following diagram.

PROPERTY	*Req Conditions	
.IdentityDocument	IF A IdentityVerificationDocument is not attached to the current case THEN display message: Please attach a document verifying identity	Edit
.EligibilityDocument	IF A EmploymentEligibilityDocument is not attached to the current case THEN display message: Please attach a document verifying eligibility for employment	Edit
(+)		

7. Click **Save**. The validation condition ensures that both documents are attached. The form displays an error message if either of the documents are not attached.
8. On the Review verification documents flow action, click **Save**. The validation rule is invoked when a user submits the Review verification documents form.

Verify your work

1. Create an onboarding case and advance it to the Review verification documents form in the Verification stage.
2. In the form, select the **Identity document provided** check box. The form displays the **Attach identity verification document** button.
3. Click the **Attach identity verification document** button to open the attachment overlay.

4. Add a sample PDF from your desktop and select the Employment eligibility document in the Category field. The following example shows a PDF that was dropped into the overlay.

Attach identity verification document

Attachments

Drag and drop files

Drag and drop attachments here

Select a file

Description

Category

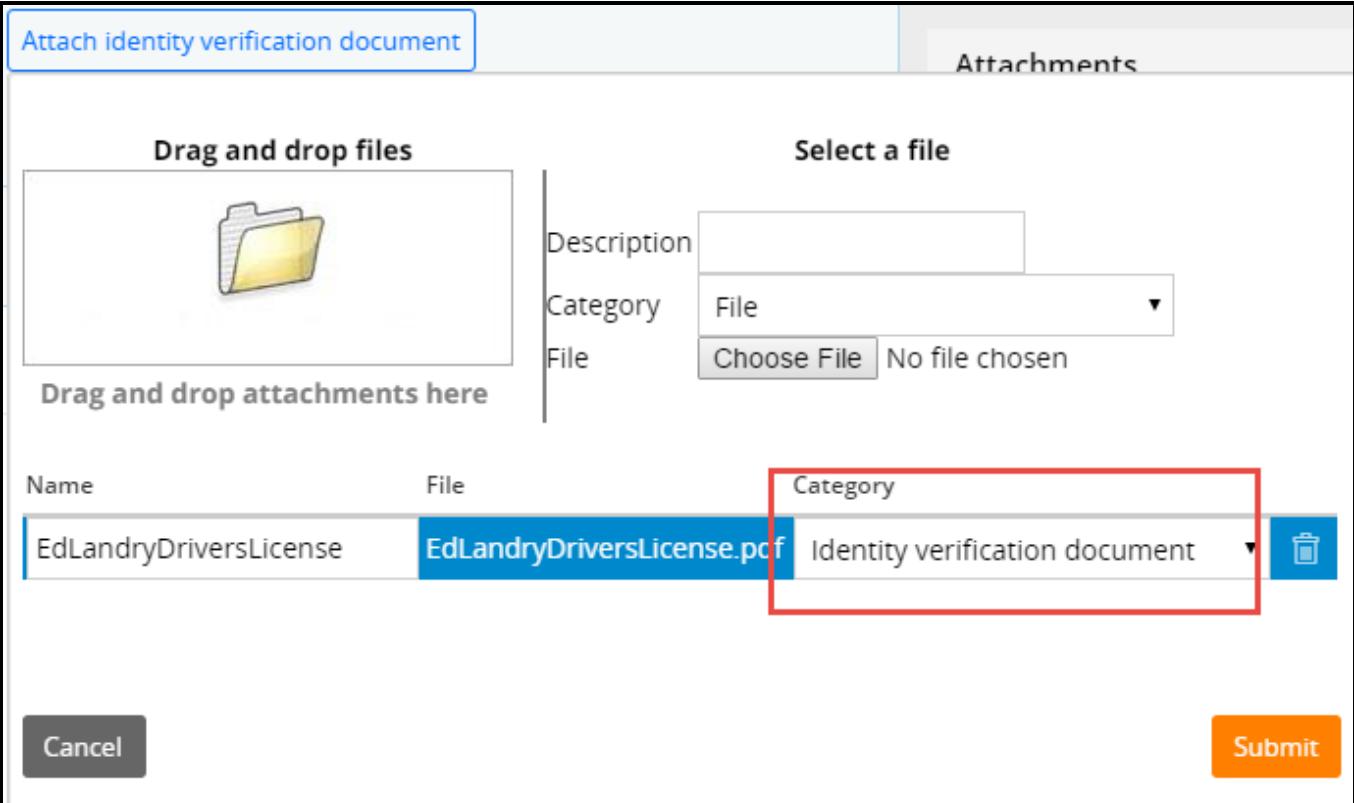
File

Choose File No file chosen

Name	File	Category
EdLandryDriversLicense	EdLandryDriversLicense.pdf	Identity verification document

Cancel

Submit



5. Click **Submit**. Because the Eligibility document is not attached, the form displays the following error message.

Review Verification Documents

SA Senior System Architect

Verify the employee's identity and eligibility for work.

Identity document provided

Eligibility document provided

⚠ ** Please attach a document verifying eligibility for employment

Attach identity verification document

Cancel

Save

Submit

6. Click the **Eligibility document provided** check box and attach another file from your desktop as described in step 4. The harness displays links to the attachments in the Attachments area on the harness.

Attachments

 EdLandryPassport	
O-58 EmploymentEligibilityDocument Senior System Architect	
 EdLandryDriversLicense	
O-58 IdentityVerificationDocument Senior System Architect	

The case audit trail records the date and time the documents were attached and provides links to the attachments.

History

Time▼	Description
9/27/16 12:42 PM	Assigned to Senior System Architect to ' complete task '.
9/27/16 12:42 PM	Assignment to ' complete task ' completed by performing a 'Review verification doc'.
9/27/16 12:42 PM	A file has been attached: EdLandryPassport.
9/27/16 12:42 PM	A file has been attached: EdLandryDriversLicense.

Case narrative

Showing [newest](#) on top

 SA	Senior System Architect attached Onboarding (O-58)  EdLandryPassport Delete 1 hour 6 mins ago
 SA	Senior System Architect attached Onboarding (O-58)  EdLandryDriversLicense Delete 1 hour 6 mins ago

Configuring flow action pre- and post-processing

Exercise: Configuring flow action pre- and post-processing

Scenario

Currently, the Candidate case type creates a candidate party to represent a job applicant, and adds the candidate party to the case at the beginning of the Screening stage. Candidates are not parties in the case throughout the Submission stage. As a result, users cannot contact candidates by email while the case is in the Submission stage.

In a future release of the HRApps application, users are allowed to save cases after a candidate has provided personal information at the start of the Submission stage. This allows users to contact by email candidates who have not completed the Submission stage.

To support this change, the vice president of Human Resources has asked you to configure the application to support emailing candidates whose cases have not completed the Submission stage.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Reconfigure the Collect Candidate Details process so that a candidate is added as a participant to the case when the user submits the Collect Personal Details form. After you complete this change, the Create Candidate Party process in the Screening stage is no longer needed.

- Update the Collect Personal Details flow action so that it runs the *addWorkPartyObject* activity as a post-processing action. Use the following activity parameter values.

Parameter	Value
PartyRole	Candidate
PartyClass	TGB-HRApps-Data-Candidate
PartyModel	NewParty

- Remove the Create Candidate Party process from the Screening stage.

Detailed steps

Update the Collect Personal Details flow action

1. Open the Candidate case life cycle.
2. Open the Collect Candidate Details process.
3. Right-click the Collect Personal Details connector and select **Open Flow Action**. This opens the *CollectPersonalDetails_0* flow action.
4. On the rule form, open the **Action** tab.
5. In the Post-Processing section, in the **Run activity** field, select or enter **AddWorkObjectParty**.
6. Update the parameter values as defined in your assignment.

Parameter	Value
PartyRole	Candidate
PartyClass	TGB-HRApps-Data-Candidate
PartyModel	NewParty

When you are done, the section will look like the following image.

The screenshot shows a configuration interface for a flow action. At the top, it says "Run activity" followed by "AddWorkObjectParty". Below this, there's a section titled "Parameters" containing three entries: "PartyRole" with value "Candidate", "PartyClass" with value "TGB-HRApps-Data-Candidate", and "PartyModel" with value "NewParty".

7. Click **Save**. When the user submits the Collect Personal Details form, the system adds the candidate to the case.

Remove the Create candidate process from the Screening stage of the Candidate case life cycle

1. On the case life cycle, in the Screening state, select the **Create candidate** process.
2. Click **X** to delete the process.
3. On the case life cycle header, click **Save**. The Create Candidate Party process is removed from the life cycle of the Candidate case type.

Note: When you remove a rule from use in a case type, do not forget to determine whether the rule is used elsewhere in your application. If the rule is unused, consider removing it from the application. If the rule exists in a previous ruleset version, you must withdraw the rule to remove it from your application.

Verify your work

Create a case and advance it to the Collect Professional Details form. The form displays the candidate in the **Participants** area on the harness.

Candidate (C-120) Actions ▾

Submission Screening Interviews Offer

Collect Professional Details
Collect the candidate's work history

Experience

+ Add Item
✖ Delete

Job title	Start date	End date
1		

Cancel Save Submit

Information Audit

Case details

Case ID C-120	Urgency 10	Status New
Created about a minute ago	Goal —	Last Update less than a minute ago
Created by Senior System Architect	Deadline —	Last Updated By Senior System Architect

Open assignments

⌚ **Collect the candidate's work history**
(Submission) ⌚
(Current)
Senior System Architect

Attachments (0)

Attach new

Tools

Follow Tags
Review Related
Print Where am I

Participants

Howard Glover
Candidate

Senior System Architect
Owner

Circumstancing rules on multiple variables

Exercise: Circumstancing rules on multiple values

Scenario

As part of the onboarding process, TGB sends an email to each new employee to inform the employee of their start date. This email also reminds the employee to bring specific paperwork on their first day at TGB. Business partners in the Human Resources (HR) department want to customize the emails by country and employee status to direct the employee to bring the appropriate paperwork on their first day.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Circumstance the Welcome email on two properties: `.pyCountryCode` and `.Contractor`. You need to create three circumstanced versions of the Welcome email as outlined in the following table.

Contractor	Country Code	Documents
true	USA	1099
false	USA	W4
false	GBR	P45

To complete this exercise, you need to:

1. Create a circumstance template using the country code and contractor properties
2. Create a circumstance definition for each of the scenarios
3. Circumstance the Welcome email for each set of conditions and reference the appropriate tax document in each circumstanced version

Detailed steps

Create a circumstance template

1. In Developer Studio, click **Create > Technical > Circumstance Template**.
2. Use the following information to complete the form.

Property	Value
Label	ContractorStatusAndCountry
Apply to	TGB-HRApps-Work-Onboarding

3. Add the following properties:
 - a. .Employee.Contractor
 - b. .Employee.pyCountryCode
4. Click **Save** to update your circumstance template.

Create circumstance definitions

1. In Developer Studio, click **Create > Technical > Circumstance Definition**.
2. Use the following information to complete the form.

Property	Value
Label	USContractor
Template Name	ContractorStatusAndCountry
Apply to	TGB-HRApps-Work-Onboarding

3. Click **Create and open** to configure the remainder of the record.
4. Set the Contractor value to **true** and the Country Code to "USA".



5. Click **Save** to update the circumstance definition.

6. Create another circumstance definition and use the following information.

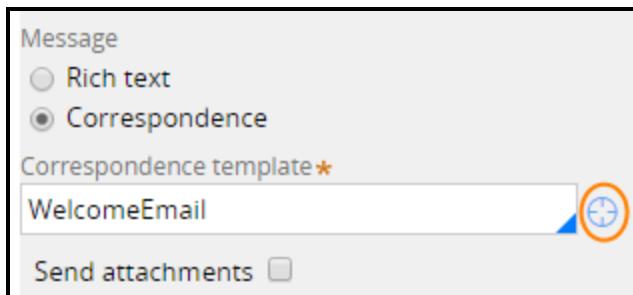
Property	Value
Label	USEmployee
Template Name	ContractorStatusAndCountry
Apply to	TGB-HRApps-Work-Onboarding
Definition	Contractor = false Country code = "USA"

7. Create a third circumstance definition and use the following information.

Property	Value
Label	GBEmployee
Template Name	ContractorStatusAndCountry
Apply to	TGB-HRApps-Work-Onboarding
Definition	Contractor = false Country code = "GBR"

Circumstance the Welcome email

- From the Cases Explorer, open the **Onboarding** case.
- Select the **Send Welcome Packet** step to determine the correspondence used to send the welcome email.
- Click the **Crosshair** icon to open the WelcomeEmail.



- Click **Save as > Specialize by circumstance** to create a circumstanced version of the record.
- Note:** The original record is used if none of the circumstanced versions are appropriate.
- Enter or select **ContractorStatusAndCountry** as the Template and **USEmployee** as the Definition.

- Click **Create and open** to create the new record.

7. Update the text of the email to add W-4 to the list of things to bring.

Please bring the following:

- Passport
- W-4

8. Click **Save** to update the correspondence.

Repeat these steps to specialize the WelcomeEmail two more times using the USContractor and GBEmployee circumstance definitions. In the USContractor circumstance, specify that the contractor should bring a 1099. In the GBEmployee circumstance, specify that the employee should bring a P45.

Verify your work

1. From the Case Manager portal, start an Onboarding case.
2. Complete the Employee Information screen. For this case, select **Contractor** and enter **USA** for the country code.

Tip: You want to create additional cases to test the employee/USA and employee/GBR conditions.

3. Complete the **Identify Home Office** and **Select Orientation Plan** steps.

Note: You can specify any values to complete the steps. They do not impact this exercise.

4. After completing the Select Orientation Plan step, the Welcome Email is sent. A copy of the email is added as an attachment to the case. Open the email and confirm the correct email was sent.

UI DESIGN

Customizing a user portal

Exercise: Customizing a harness

Scenario

Based on feedback from end users, the vice president of Human Resources requests a change to how information is displayed in the Case Worker and Case Manager portals. The approved requirement is to position the list of case participants so it displays below the list of open assignments for each case. Stakeholders throughout the TGB organization have agreed to the change, and request that their applications also reflect this change.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Reorganize the right-side column of the Perform harness so that the *Participants* section is displayed below the *Open assignments* section.

To do this, copy the rule that contains these two layouts into the TGB ruleset, and then reorder the layouts as requested.

Tip: By choosing the TGB ruleset, changes to the Perform harness apply to any application using the TGB ruleset.

Detailed steps

Follow these steps to customize a harness.

1. Create a case of your choice.

Note: The Perform harness is the standard harness used for all cases in an application so you can edit the Perform harness from any case in the application (Benefits Enrollment, Employee Evaluation, or Onboarding).

2. Use **Live UI** to find and open the *pyCaseAssets* section record in Designer Studio.

The screenshot shows the 'Section in cell - pyCaseAssets' view. At the top, there are several icons: a gear, a person, a magnifying glass, and a list. A tooltip 'Open rule in Designer Studio' points to the gear icon. Below the icons, the section record details are listed:

- Open assignments**:
 - Collect information about the candidate (Submission) (Current)
Senior System Architect
- Attachments**: (0)
An upward arrow icon and a '+ Attach new' button.
- Tools**:
 - Follow
 - Review
 - Print
 - Tags
 - Related
 - Where am I
- Participants**:
 - SA Senior System Architect Owner
- Manage**: An icon with a gear and a person.

3. Save a copy of the *pyCaseAssets* section record in the highest unlocked version of the TGB ruleset.

Important: Do not change the **Apply to:** context.

Tip: Make a note of the ruleset version to which you saved the *pyCaseAssets* section record. You will use this version number to verify your changes.

4. Move the layout named *Dynamic Layout (Simple List)* 2 directly under the layout named *Dynamic Layout (Default)* 1.

The screenshot shows the Pega Platform's layout editor. At the top, there are tabs: Design (which is selected), Settings, Parameters, Pages & Classes, HTML, and Specifics. Below the tabs is a toolbar with various icons. The main area displays a layout tree. The first item is 'Dynamic Layout (Default) - 1', followed by 'Dynamic Layout (Simple list) - 2'. A red box highlights the handle of 'Dynamic Layout (Simple list) - 2'. Underneath 'Dynamic Layout (Simple list) - 2' is a sub-item 'Repeating Dynamic Layout (Default) [.pyWorkParty of Class Data-Party] - 2.1'.

Tip: Click and hold the handle of any layout to move it.

5. Save your changes.

Verify your work

1. Create a case of your choice, and confirm the **Participants** list is displayed below the **Open assignments** list.

The screenshot shows a case detail view. In the 'Open assignments' section, there is one item: 'Collect information about the candidate (Submission)' with a location pin icon. In the 'Participants' section, it lists 'Senior System Architect' as the owner, accompanied by a phone and email icon. At the bottom is a 'Manage' button.

2. Use **Live UI** to open the *pyCaseAssets* section record in Designer Studio and confirm the section record in use is from the TGB ruleset — and not from the Pega-provided UI-Kit-7:07-01-01 ruleset.



Exercise: Customizing a portal

Scenario

The vice-president (VP) of the Human Resources (HR) department has requested that you customize the manager portal to meet the needs of HR managers.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

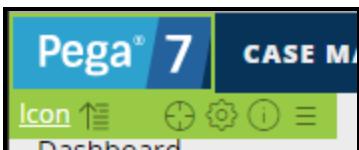
Customize the Case Manager portal to reflect the following changes:

- In the portal header, replace the Pega logo (webwb/pega-7-logo.svg) in the portal with the TGB logo (webwb/tgb-logo.svg).
- In the navigation area, replace the **Configuration** entry in the menu with an item labeled **Delegated rules**.
- In the content area, add a widget to display a chart on the dashboard. Configure the widget to display the Timeliness of Resolved Items By Operator report.

Detailed steps

Update the logo in the portal header

- In Designer Studio, click **Launch > Manager** to open the Case Manager portal in a new browser window.
- Click **Live UI** to determine what to edit.
- Select the **Pega 7 logo** and click the **Crosshair** icon to open the Case Manager portal header in Designer Studio.



- In Designer Studio, `pyPortalHeader` should now be open. Click **Save As** and save the rule to your ruleset.

5. The logo is displayed in a field next to the Case Manager. Click the field and then the **Gear** icon to edit the image.



6. Update the image field to **webwb/tgb-logo.svg**.

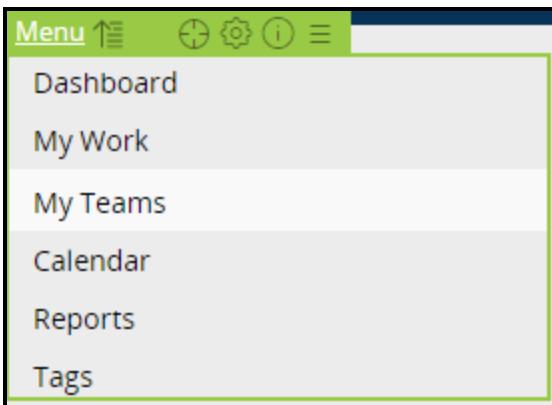
Note: For this exercise, the image named in this step has been uploaded to your Pega server as a binary file rule. If time allows, use the Help topic "About Binary File rules" to upload an image of your choosing (103 px by 40 px).

7. Click **Save** to update the configuration.

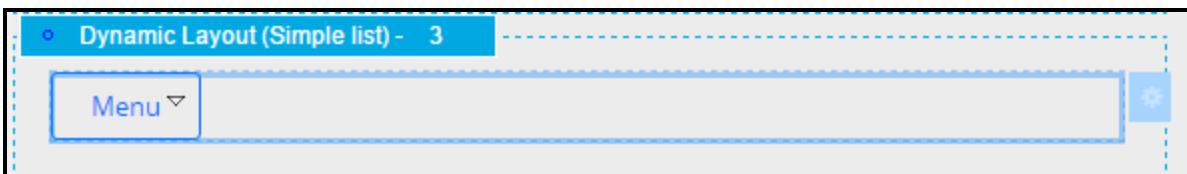
Open the Case Manager portal and confirm that the TGB logo now displays in the upper left corner.

Update the navigation area

1. In Designer Studio, click **Launch > Manager** to open the Case Manager portal in a new browser window.
2. Click **Live UI** to determine what to edit.
3. Select the left navigation menu and click the **Crosshair** icon. This opens the Case Manager Portal Navigation section in Designer Studio.



4. In Designer Studio, Case Manager Portal Navigation should now be open. Click the **Gear** icon to edit the properties of the menu.



5. Click the **Crosshair** icon to access the Navigation options for the menu.

The screenshot shows the 'Cell Properties' dialog with the 'General' tab selected. Under the 'Navigation' section, there is a field labeled 'pyCaseManagerLinks' with a small blue crosshair icon to its right. This icon is highlighted with a red circle. Below the field is a checkbox labeled 'Defer load sub menu'.

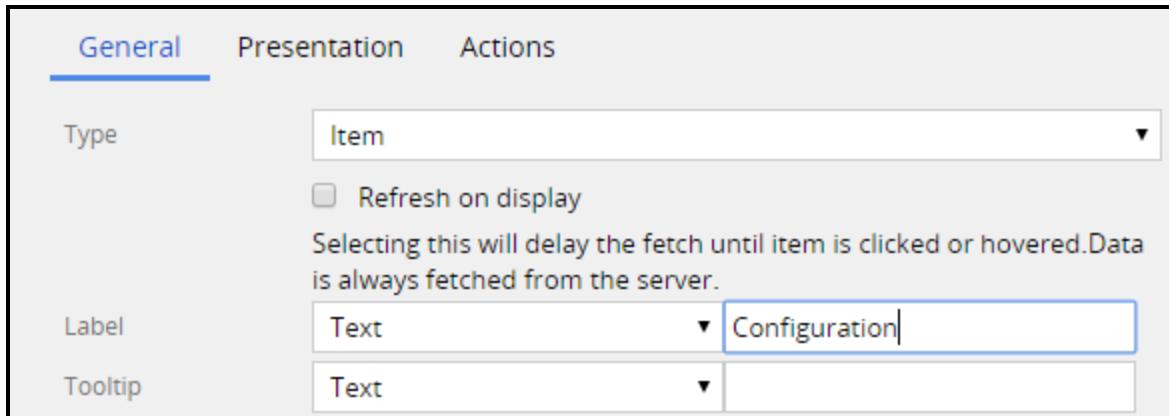
6. Click **Save As** and save the record to your ruleset. This allows you to edit the Manager Portal Navigation record.
7. Double click the **Configuration** row.

The screenshot shows a table titled 'Manager Portal Navigation'. At the top, there are buttons for 'Add above', 'Add below', 'Add child node', 'Delete', and 'Refresh'. The table has three columns: 'Label', 'Type', and 'Action(s)'. The rows are as follows:

Label	Type	Action(s)
Dashboard	Item	Harness, Run Script
My Work	Item	Harness, Run Script
My Teams	Item	Harness, Run Script
Calendar	Item	Harness, Run Script
Reports	Item	Harness, Run Script
Configuration	Item	Run Script, Run Script
pyMobileSettings	Reference	None
Tags	Item	Harness, Run Script
Log off	Item	Log Off

The 'Configuration' row is highlighted with a light blue background. The entire table is enclosed in a black border.

8. Change the **Label** to Manage Delegated Rules.



9. Click **Save** to update the configuration.

In the navigation menu on the left, you should see Manage Delegated Rules link.

Add and configure a widget on the dashboard

1. Open the Case Manager portal.
2. Click the **Gear** icon to configure the dashboard.

Tip: The gear icon is located in the upper right corner of the Case Manager portal.

3. In the left column, click **Add widget(s)**.
4. Select the **Report** widget.
5. Click **Add selected** to add the widget to the dashboard.
6. Select the report widget you added.

Tip: When selected, a green rectangle highlights the widget.

7. Configure the widget to use the **Timeliness of Resolved Items By Operator** report.

Tip: This report is in the Monitor Assignments category.

8. Click **Save** to update the configuration settings for the reports widget.
9. Click **Publish** to update the dashboard.

You should now see the Timeliness of Resolved Items By Operator report displayed on the dashboard.

Designing a mobile-ready application

Exercise: Designing a mobile-ready application

Scenario

During the onboarding process, employees must provide documents to verify their identity and eligibility to work. After submitting these documents, the employee must attest that the documents are genuine. The current version of the application provides the employee with a check box for attestation. By clicking the check box, the employee affirms the attestation.

The Human Resources (HR) department wants to use tablets to reduce the opportunity for fraud when confirming an employee's documentation. Rather than attesting to the validity of these documents by clicking a check box, HR plans to use tablets to capture the employee's signature. The signature is then added to the onboarding case as an attachment to verify compliance with government regulations.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

To complete this task, you need to:

- Update the Certify document authenticity section to replace the check box with a signature control
- Configure the control to use the property `.Employee.pyLastName` as the file name

Detailed steps

Update the section to use the Signature Control

1. In the Case Designer, open the Onboarding case type.
2. Select the **Certify Verification Documents** step.
3. Click **Configure view** to modify the corresponding section.
4. Drag a **Signature Capture** control onto the section.
5. Open the properties panel for the control.
6. In the **Signature name** field, enter or select `.Employee.pyLastName`.

Note: The **Signature name** field is used to provide a name for the image generated by the signature capture control. In this case, the property used is the last name of the employee.

7. Click **Submit** to complete the configuration of the control.
8. Remove the check box control from the section.

Verify your work

1. From the Case Manager portal, start an Onboarding case.
2. Complete all the steps until the Verification stage.
3. The final step of the Verification stage displays your updated section. Use the mouse to add a signature and click **Accept**.
4. Click **Submit** to complete the case. An attachment added to the case with a file name that uses the last name of the employee is displayed.

Customizing the look and feel of an application

Exercise: Updating the look and feel of an application

Scenario

To ensure consistency across all of TGB's systems, the Human Resources (HR) department wants the HRApps application to incorporate elements of TGB's corporate branding. As part of this request, the vice president of HR has requested that links, table rows, and button backgrounds use the shade of green used in the TGB logo.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

In order to complete this assignment, you need to:

- Create a topography mixin to display green text (#007000)
- Apply the topography mixin to the Link format
- Create a mixin with the background color green (#007000) and text white (#FFFFFF)
- Apply the mixin to the Buttons format
- Apply the mixin to the Even rows for reports using the tree and grid format

Detailed steps

Create a topography mixin to display green text

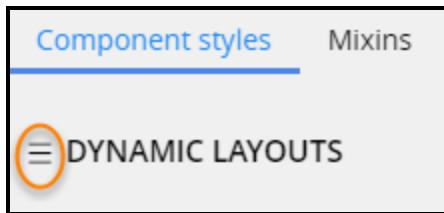
1. In Designer Studio, on the Application menu click **Open Application Skin**.
2. Save the *HRApps* skin into the *HRApps:01-01-02* ruleset version so that it can be updated.
3. Click **Mixins**.
4. Under **Typography**, click **Create new mixin**.

5. Enter **CorporateText** as the name.
6. Click **Submit**.
7. Change the **Font color** to #007000.
8. Change the **Font size** to 12.
9. Click **Save**. This updates the mixin configuration.

Apply the topography mixin to the Links format

1. Click **Component Styles**.

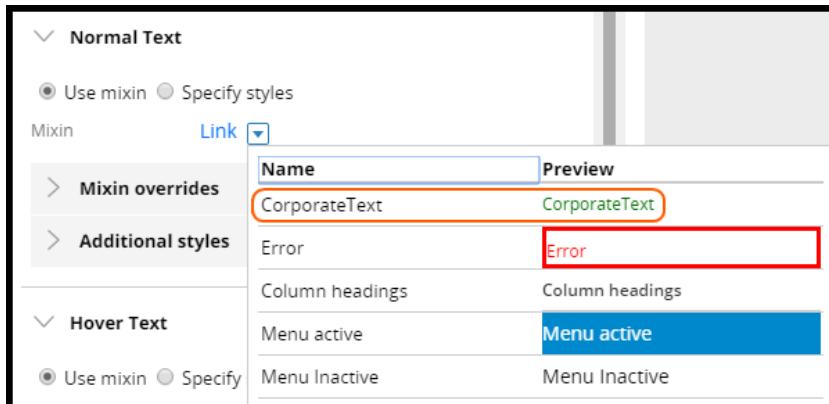
2. Click the slide out menu.



3. Click **Links**.
4. Click **Inherited** to update an existing format in the skin.
5. Click **Standard** to modify the format for field labels.
6. Click **override it** to change the style.



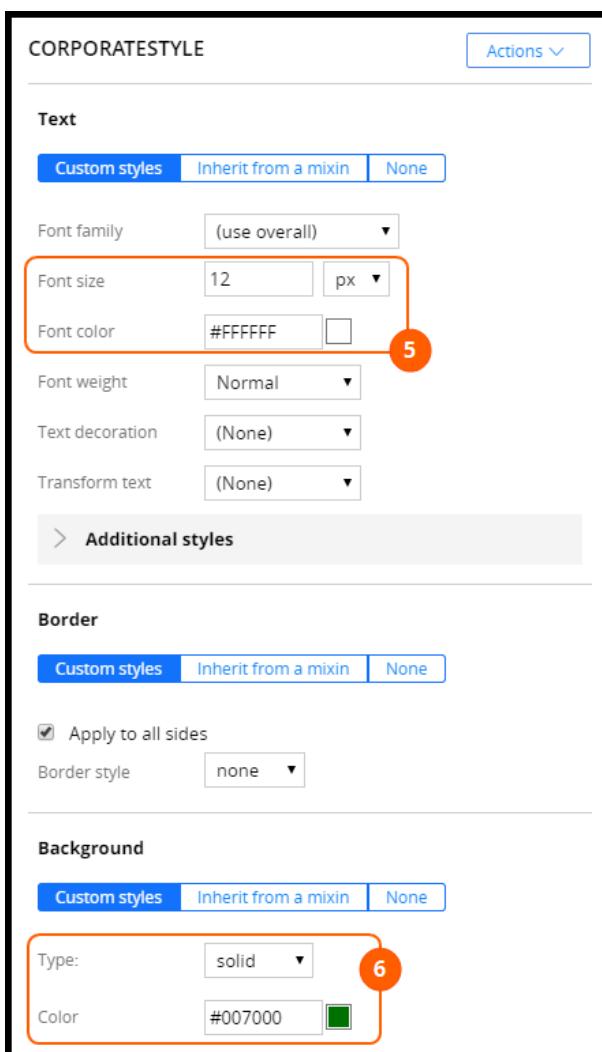
7. Expand **Normal Text** and change the mixin to **CorporateText**.



8. Click **Save** to update the format.

Create a mixin with the background color green and text white

1. Click **Mixins**.
2. Under **Combinations**, click **Create new mixin**.
3. Enter **CorporateStyle** as the name.
4. Click **Submit**.
5. Under **Text**:
 - a. Change the **Font color** to **#FFFFFF**.
 - b. Change the **Font size** to **12**.
6. Under **Background**:
 - a. Change the **Type** to **solid**.
 - b. Change the **Color** to **#007000**.



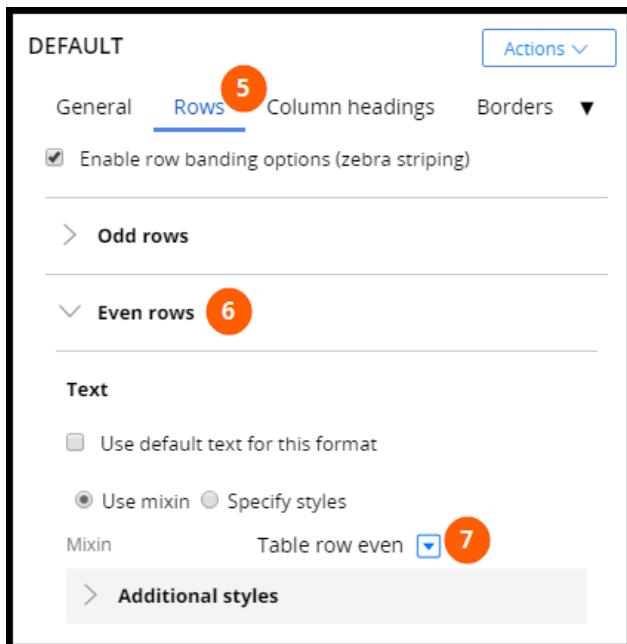
7. Click **Save** to update the mixin.

Apply the mixin to the Buttons format

1. Click **Component Styles**.
2. Click the slide out menu.
3. Under **Controls**, click **Buttons**.
4. Click **Inherited** to update an existing format in the skin.
5. Click **Standard** to modify the format for buttons.
6. Click **override it** to change the style.
7. Under **Text**, change the specified mixin to **CorporateStyle**.
8. Under **Background**, specify the **CorporateStyle** mixin.
9. Click **Save** to update the format.

Apply the mixin to the Even rows for reports

1. Under **Layouts**, click **Trees & grids**.
2. Click **Inherited** to update an existing format in the skin.
3. Click **Default** to modify the format.
4. Click **override it** to change the style.
5. Open the **Rows** tab to change the format of the rows.
6. Expand **Even rows**.
7. Under **Text**, open the Table row even menu.



8. Change the specified mixin to **CorporateStyle**.

9. Under **Background**, specify the **CorporateStyle** mixin.
10. Click **Save** to update the format.

Verify your work

1. Click **Launch > Manager** to open the Manager portal.
2. Click **Reports** to observe the following changes:
 - a. Links to each report should be green.
 - b. Even rows should have a green background.
3. Click **Calendar** to observe the buttons are now green.

REPORT DESIGN

Creating reports that combine data from multiple tables

Exercise: Creating reports that combine data from multiple tables

Scenario

The business partners in the Human Resources (HR) department have requested two reports.

The first report is to track the progress of candidate cases generated in the HRApps application. This report should return a list of open cases, displaying the status, urgency, assigned workbasket, and current assigned task for each case.

The second report is to comply with government regulations that require employees to enroll in a health insurance plan. For each onboarding case, this report should identify the medical insurance plan selected by the employee.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Create two reports for the HR business partners.

The first report lists every candidate case. For each case, list the current status of the case, the workbasket to which the case is currently routed, the current assignment for the case, and the assignment urgency.

The second report lists every onboarding case. For each case, list the last name of the associated employee, the ID of the benefits enrollment child case, the current status of the benefits enrollment case, and the selected medical plan.

Detailed steps

Create a report that joins classes referencing an association rule

Use an association rule to join the Assign-Workbasket class with the TGB-HRApps-Work-Candidate class.

1. In the Candidate case type, create a report definition named **Candidates by workBasket**.

Tip: To create the report definition, in the Application Explorer, right-click the case type and select **+Create > Reports > Report Definition**.

2. In the **Edit columns** section, in the **Column source** and **Column name** fields, enter the following values. Use the pxWorkbasketAssignment prefix to identify the properties in the *Assign-Workbasket* class you want to use in the report.

Column source	Column name
.pyID	Case ID
.pyStatusWork	Work Status
pxWorkbasketAssignments.pxAssignedOperatorID	Workbasket ID
pxWorkbasketAssignments.pxUrgencyAssign	Urgency of Assignment
pxWorkbasketAssignments.pxTaskLabel	Task Label

Note: Selecting one of the three properties with the pxWorkbasketAssignments prefix adds an association rule to the report. The association rule allows the report to return data from classes that are mapped to two database tables.

When you are done, the columns look like the following image.

Edit columns

Column source	Column name	Sub
.pyID	Case ID	<
.pyStatusWork	Work Status	<
pxWorkbasketAssignments.pxAssignedO	Workbasket ID	<
pxWorkbasketAssignments.pxUrgencyAs	Urgency of Assignment	<
pxWorkbasketAssignments.pxTaskLabel	Task Label	<

[+ Add column](#)

3. Click **Save** to save your report definition.
4. To verify your configuration, on the rule form header, click **Actions > Run** to generate the report. The report lists the cases, their work status, the workbaskets with which they are associated, and the current assignments and their urgencies.

Case ID	Work status	Workbasket ID	Urgency of Assignment	Task Label
C-81	Open-Scheduling	RecruitingWB	90	Schedule Interview
C-133	Open	RecruitingWB	100	Conduct phone screen
C-23	Open	RecruitingWB	100	Conduct phone screen
C-41	Open	RecruitingWB	10	Get Approval
C-147	Open	RecruitingWB	10	Background Check
C-134	Open	RecruitingWB	100	Conduct phone screen
C-71	Open	RecruitingWB	100	Conduct phone screen
C-106	Open	RecruitingWB	100	Conduct phone screen
C-76	Open-Interview	AP@TGB	10	Get Approval
C-49	Open	RecruitingWB	10	Get Approval
C-61	Open	RecruitingWB	10	Get Approval
C-135	Open	RecruitingWB	100	Conduct phone screen
C-15	Open	RecruitingWB	100	Conduct phone screen
C-92	Open	EmployeeReferralsWB	100	Conduct phone screen
C-109	Open-Assessment	RecruitingWB	90	Schedule Interview

Create a report that uses a class join in the report definition

Use a class join in the report definition to join the *TGB-HRApps-Work-BenefitsEnrollment* class with the *TGB-HRApps-Work-Onboarding* class.

1. In the Onboarding case type, create a report definition named **Employee medical selection**.
2. Open the Data Access tab.
3. In the **Class joins** section, click **Add class join** to add a row.
4. In the **Prefix** field, enter **BE** to create a prefix for benefits enrollment cases. Use this prefix to reference properties in the Benefits Enrollment class.
5. In the **Class name** field, enter **TGB-HRApps-Work-BenefitsEnrollment**. This is the class you want to join to the report class. Do not change the default value in the **Type** field — **Only include matching rows**.
6. At the end of the row, click **Edit conditions**. The system displays the **Enter filter conditions** dialog.
7. In the **Column** field, enter **BE.pxCoverInsKey**.
8. In the **Relationship** field, leave the default value **is equal**.
9. In the **Value** field, enter **.pzInsKey**.

Enter filter conditions

Filter conditions			
A			
Condition	Column	Relationship	Value
A	BE.pxCoverInsKey	is equal	.pzInsKey

10. Click **Submit** to save your filter condition and close the dialog.

11. Open the Query tab.
12. In the **Edit columns** section, enter the following **Column source** and **Column name** values. Use the BE prefix to find properties in the *TGB-HRApps-Work-BenefitsEnrollment* class you want to use in the report.

Column source	Column name
.pyID	Case ID
.Employee.pyLastName	Last Name
BE.pyStatusWork	Benefits Enrollment Work Status
BE.pyID	Case ID
BE.MedicalPlan.Name	Plan name

When you are done, the report definition looks like the following image.

Column source	Column name	Summa
.pyID	Case ID	<blank>
.Employee.pyLastName	Last Name	<blank>
BE.pyStatusWork	Benefits Enrollment Work St	<blank>
BE.pyID	Case ID	<blank>
BE.MedicalPlan.Name	Plan name	<blank>

(+) Add column

Verify your work

1. To verify your configuration, on the rule form header, click **Actions > Run** to generate the report.

For each onboarding case, the report includes the case ID and the last name of the employee associated with the case. For each joined benefits enrollment case, the report lists the work status of the employee's benefits enrollment case, the case ID, and the selected medical plan.

Case ID	Last Name	Benefits Enrollment Work Status	Case ID	Plan name
O-25	McGrail	Open	B-13	Medical Premium
O-29	Martinez	New	B-19	Medical Advanced
O-44	Jones	Resolved-Completed	B-21	Medical Premium
O-40	Wilson	New	B-27	Medical Advanced

DATA MANAGEMENT

Exposing an application with a service

Exercise: Exposing an application with a SOAP service

Scenario

The Human Resources (HR) department partners with several staffing services to expose open positions to a wider audience of potential job applicants. These staffing services have requested that TGB allow them to submit applicants for open positions directly from their own client management systems.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Create a SOAP service. Configure the service to accept the following input for candidate cases:

- First name
- Last name
- Email
- Candidate comment
- Position applying for
- Experience (list with job title, company, start date, end date)

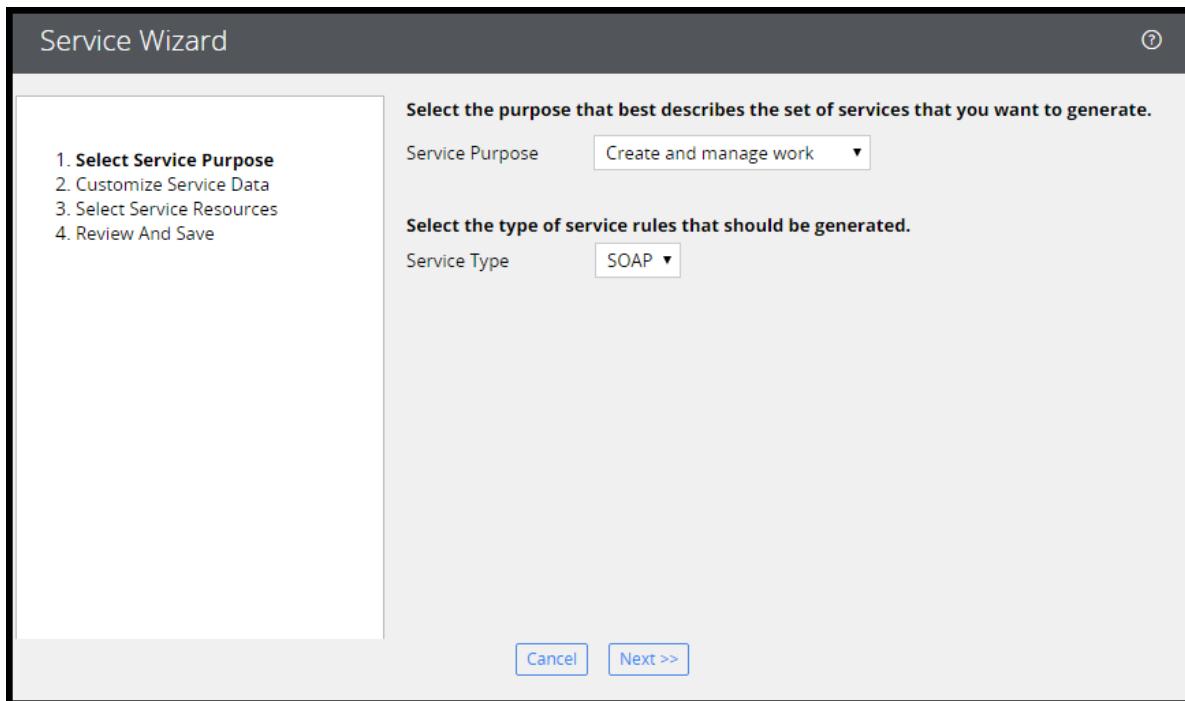
If the candidate case is created by the SOAP service, skip the steps in the Collect Candidate Details process.

Note: If you completed the **Working with temporary cases** exercise, go to the Candidate case and open the *pyStartCase* flow. Then, on the Process tab, deselect **Temporary object** before continuing with this exercise.

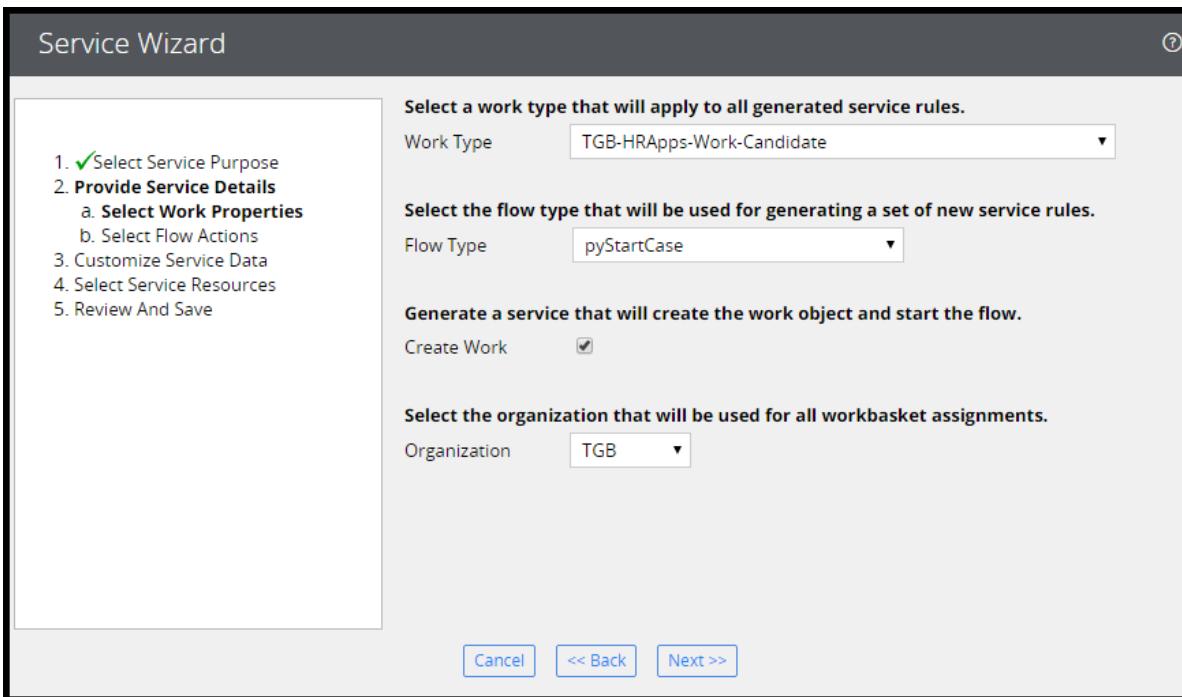
Detailed steps

Run the Service wizard

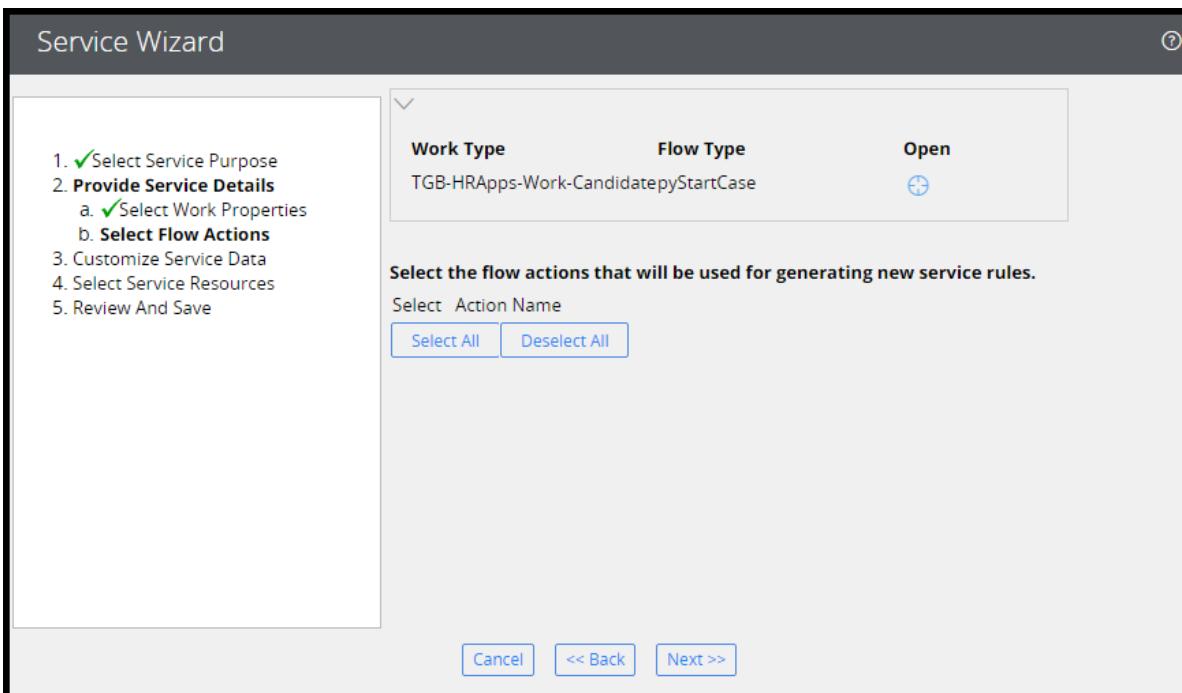
1. Select **Designer Studio > Integration > Services > Service Wizard** to start the Service wizard.
2. Set the **Service Purpose** to **Create and manage work** and the **Service Type** to **SOAP**, then click **Next**.



3. Set **Work Type** to TGB-HRAApps-Work-Candidate and **Flow Type** to pyStartCase, select **Create Work** and set the **Organization** to TGB, then click **Next**.



4. Click **Next**. (The *pyStartCase* flow does not have any flow actions that you want to consider a service to execute.)



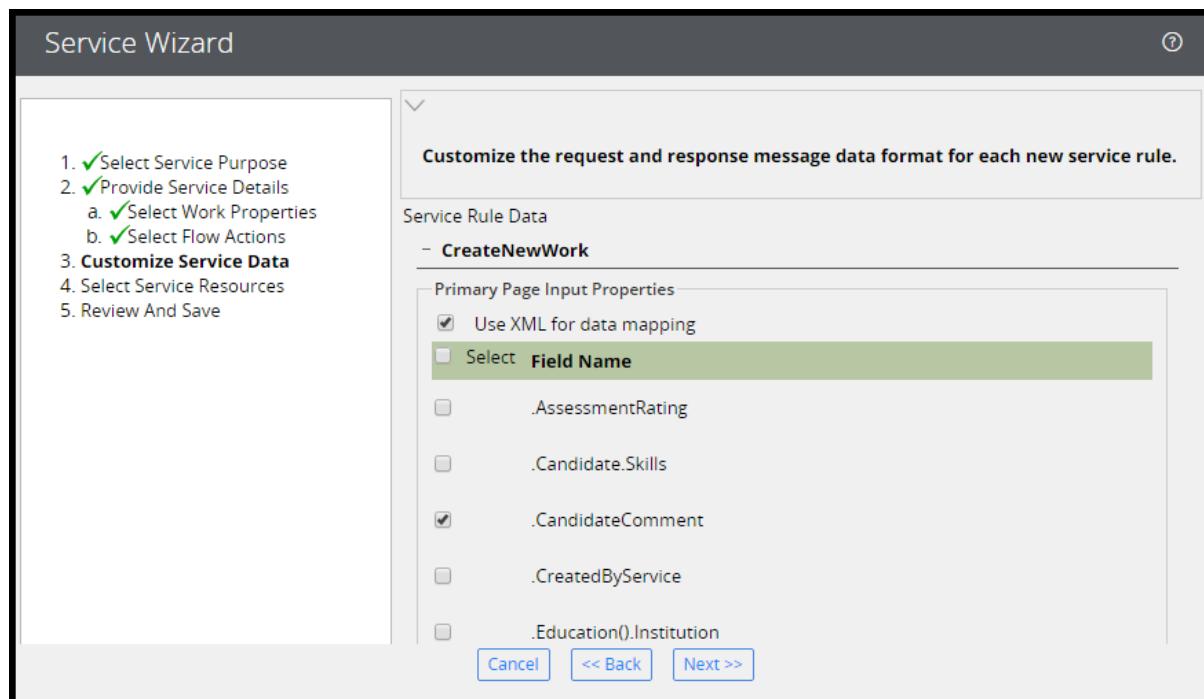
5. Select **Use XML for data mapping** to allow for list structures in the request.

Note: Only properties defined directly in the TGB-HRApps-Work-Candidate case type are shown. Inherited properties are not shown and need to be added manually after the wizard has been completed.

6. Select the following input properties:

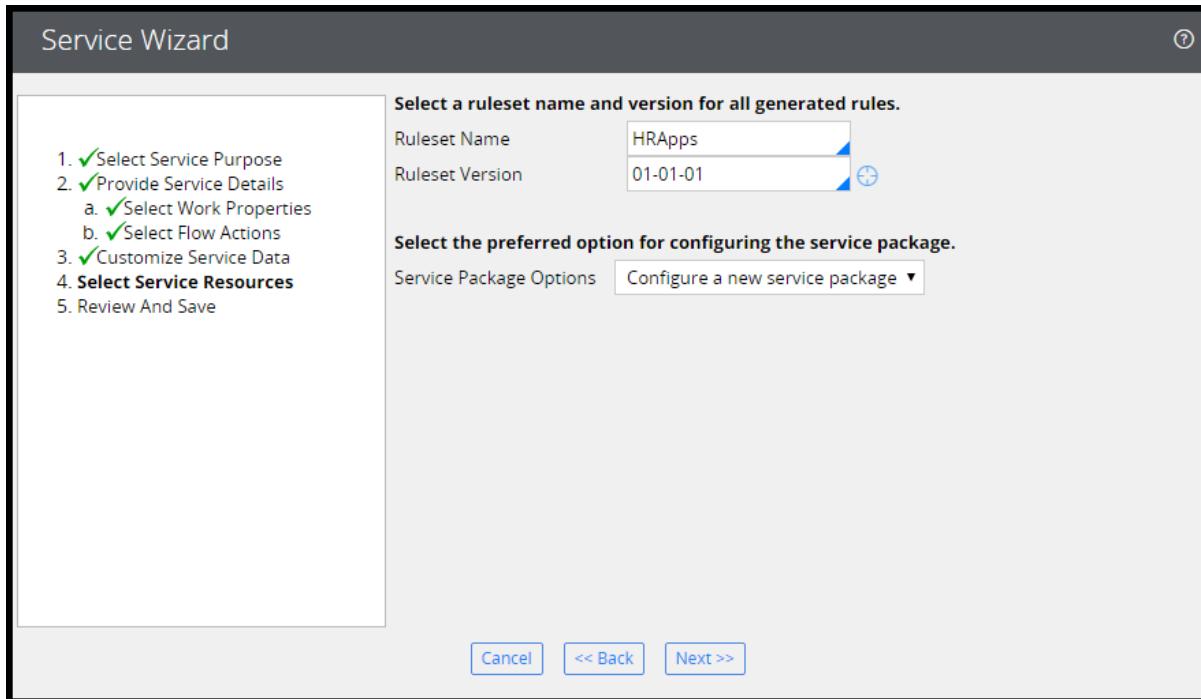
- .CandidateComment
- .PositionAppliedFor
- .Experience().Company
- .Experience().EndDate
- .Experience().JobTitle
- .Experience().StartDate

Click **Next**.



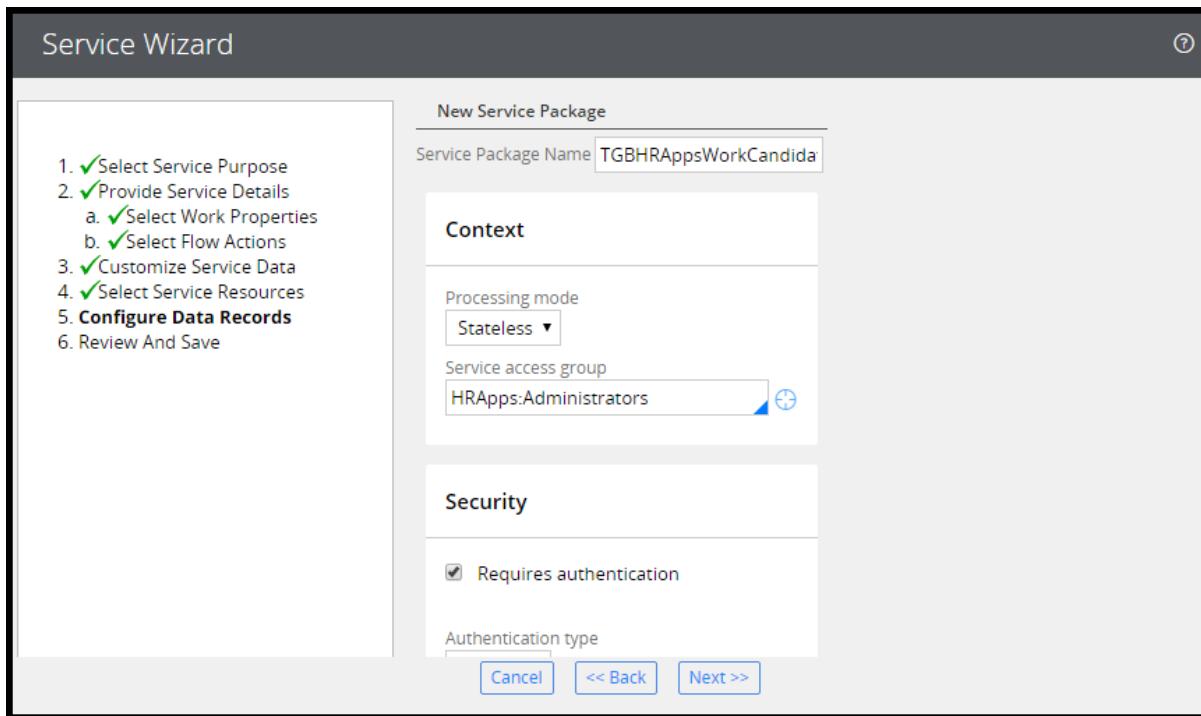
7. Set the **Ruleset Name** to **HRApps** and the **Ruleset Version** to the version in which you are working.

8. From the **Service Package Options** drop-down list, select **Configure a new service package**, then click **Next**.

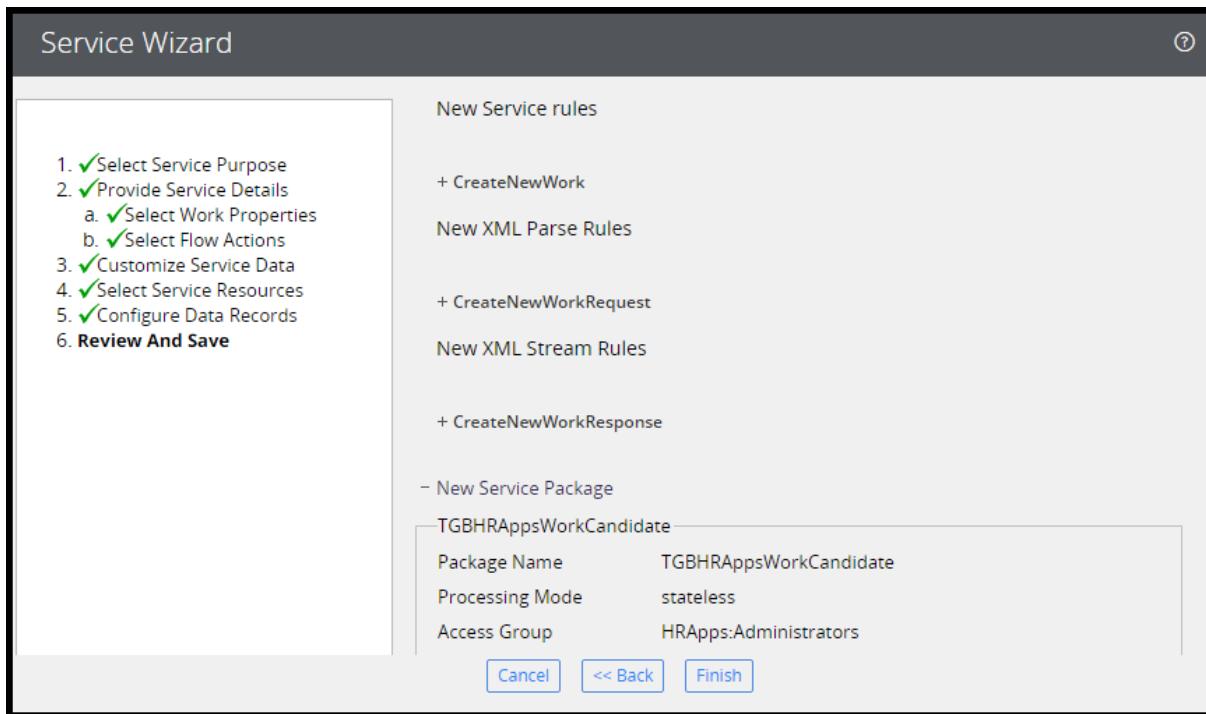


9. Keep the default settings in the Configuration Data Records screen. Click **Next**.

Note: **Requires authentication** is selected by default. This means that you need to provide an operator and password when you call your service.



10. The Review and Save screen displays the records that are created by the wizard. Click **Finish** to create the records.



The final page displays the records created by the wizard.

Service Wizard ?

The service configuration process is now complete.
Click the *Done* button to exit the accelerator.

Generated Service Rules

Type	Package	Class	Method	RuleSet	Version	Open
Rule-Service-SOAP	TGBHRAAppsWorkCandidate	pyStartCase	CreateNewWork	HRApps_Branch_SOAPService	01-01-01	

Generated XML Parse Rules

Applies To	Namespace	Element Name	Ruleset	Version	Open
TGB-HRAApps-Work-Candidate	pyDefault	CreateNewWorkRequest	HRApps_Branch_SOAPService	01-01-01	

Generated XML Stream Rules

Applies To	Stream Name	XML Type	Ruleset	Version	Open
TGB-HRAApps-Work-Candidate	CreateNewWorkResponse	MapFrom	HRApps_Branch_SOAPService	01-01-01	

Service Package Instance

Package Name	Open
TGBHRAAppsWorkCandidate	

WSDL URL

<http://10.61.9.195:9080/prweb/PRSOAPServlet/SOAP/TGBHRAAppsWorkCandidate/pyStartCase?WSDL>

[Done](#)

Add missing input parameters

1. Open the *CreateNewWorkRequest* XML parse rule generated by the wizard.
2. Use the **Add Element** button to add the required input parameters that were not available for selection in the wizard.

The screenshot shows the 'Mapping' tab of the Pega studio. At the top, there are tabs for 'Mapping', 'Pages & Classes', 'XML', 'Specifications', and 'History'. Under the 'Mapping' tab, there is a checkbox 'Allow tree editing?' followed by three buttons: 'Add Attribute', 'Add Element' (which is highlighted with an orange box), and 'Delete'. Below these buttons is a tree view of XML nodes. The root node is 'CreateNewWorkRequest'. Under it, there are several nodes: 'FirstName .Candidate.pyFirstName', 'LastName .Candidate.pyLastName', 'Email .Candidate.pyEmail1' (all three are highlighted with an orange box), 'CandidateComment .CandidateComment', 'PositionAppliedFor .PositionAppliedFor', and 'Experience (.Experience)'. Under 'Experience', there are four more nodes: 'Company .Company', 'EndDate .EndDate', 'JobTitle .JobTitle', and 'StartDate .StartDate'.

3. Enter a Node Name and specify the Property to map the value to for each element.

The screenshot shows the 'User Data' mapping configuration window. It has sections for 'Node Name:' (containing 'FirstName'), 'Node Type:' (set to 'Element'), 'Mapping Mode:' (set to 'Standard'), 'Context Page:', 'Override Class:', 'Context Class:' (containing 'TGB-HRApps-Work-Candidate'), 'Property:' (containing '.Candidate.pyFirstName'), and a 'Group Subscript' checkbox. The 'Node Name:' and 'Property:' fields are highlighted with orange boxes.

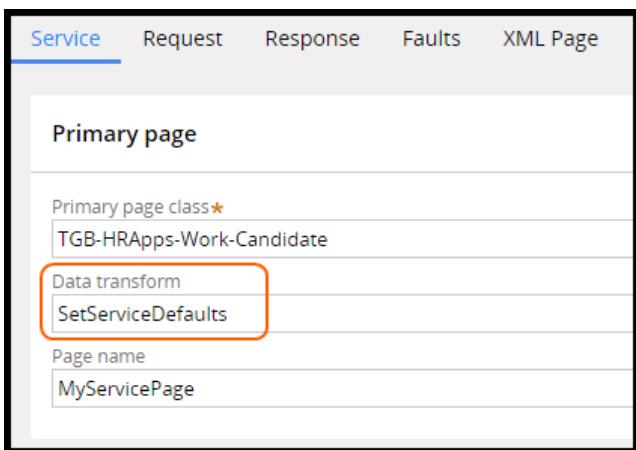
Configure the case to skip the Submission stage for cases submitted using the service

1. Add a Boolean property named *CreatedByService* to the Candidate case type.

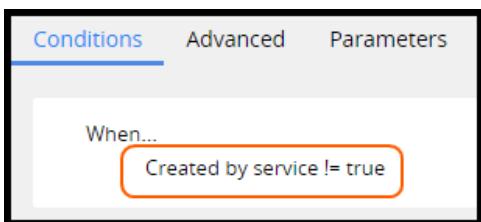
Tip: To add a property, in the App Explorer, right-click the **Candidate** case type and select **Create properties** to open the Data model for the case type.

2. Create a data transform named *SetServiceDefaults* that sets the property *CreatedByService* to true.
3. On the **Service** tab of the service rule, in the **Data transform** field, add the data transform to invoke the data transform when the service primary page is created .

Tip: SOAP service records are in the Integration-Services category. From the Records Explorer, expand Integration-Services > Service SOAP to list available SOAP service records.

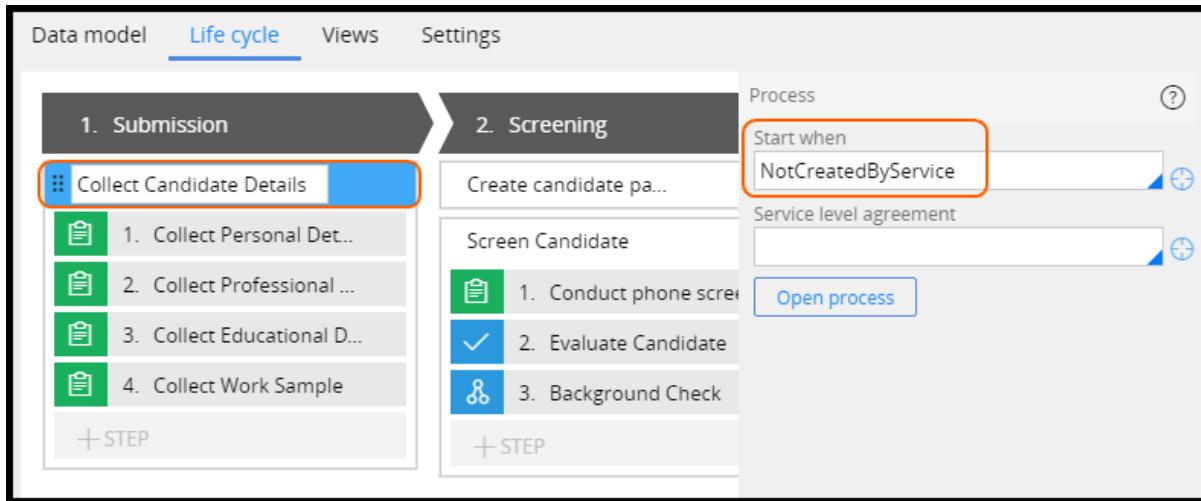


4. Create a when rule named *NotCreatedByService* for the Candidate case type that returns false if the case is created by a service.



5. In the Case Explorer, select the Collect Candidate Details stage.

6. Use the when condition in the **Start when** field for the *Collect Candidate Details* stage.



Verify your work

1. Open the *CreateNewWork* SOAP service rule.
2. Select **Actions > Run** to test the service.

3. Select **Supply SOAP request envelope** and enter some test data, then click **Execute**. For the value of **PositionAppliedFor**, enter REQ1.

Execute TGBHRAppsWorkCandidate.pyStartCase.CreateNewWork

Requestor Context

- Use current requestor context
- Initialize service requestor context

Enter Request Data

- Specify individual request values
- Supply SOAP request envelope

SOAP Request Envelope

```
<?xml version="1.0"?>
<soap:Envelope
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <CreateNewWorkRequest>
      <FirstName>John</FirstName>
      <LastName>Smith</LastName>
      <Email>john@invalidemail.com</Email>
      <CandidateComment>Available for interview next week.</CandidateComment>
      <PositionAppliedFor>REQ1</PositionAppliedFor>
      <!-- this element may appear multiple times --><Experience>
      <Company>Pegasystems Inc.</Company>
      <EndDate>2016-01-01</EndDate>
      <JobTitle>Manager</JobTitle>
      <StartDate>2010-05-01</StartDate>
    </Experience>
  </CreateNewWorkRequest>
</soap:Body>
</soap:Envelope>
```

4. In the App Explorer, right-click the Candidate case type and select **View > Instances** to verify that a new Candidate instance was created.

ID*	Short description	Updated on	Updated by
C-135	Candidate	9/26/2016 12:02 PM	Senior System Architect
C-134	Candidate	9/26/2016 11:13 AM	Senior System Architect
C-133	Candidate	9/26/2016 11:12 AM	Senior System Architect

Tip: If the instance was not created, start the Tracer and rerun the wizard to debug the service. To use the Tracer to debug a service, open the Tracer Settings dialog and under **Event types to trace**, select **Services**.

5. Open the Candidate case instance created by the service.

6. Verify that the **Submission** stage is completed and that the **First name**, **Last name**, and **Position applied for** fields are populated.

Candidate (C-135)

Submission ✓ Screening Interviews Offer

23 hours from now

Conduct phone screen Qualify candidate for the selected position	⌚ Due in 2 days 23 hours from now	
First Name John	Last Name Smith	
Job title —	Manager	Department —
Position applied for Director ▾		

Exercise: Accessing a Pega application using the Pega API

Scenario

IT is adding the ability for users to work on an internal portal. They submitted a requirement to display open positions within TGB to users of the portal. You have been assigned to test the API calls to the HRApps application.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	ssa@tgb	rules

Your assignment

Note: This exercise does not describe how you build the external application. The goal is to demonstrate the HTTP query needed to access your Pega application and the resulting JSON response.

Use the Pega API landing page to query the data page D_PositionsList to return a list of all open positions.

Bonus: Access the data page for a single position, D_Position. Observe the JSON response. Remember, the Pega API landing page does not allow you to pass parameters to a data page. So, you need to build the HTTP query and use a browser window to run it. You also need to add the required ID parameter as part of the query string. The URL you use looks similar to:

`http://<host>:9080/prweb/api/v1/data/D_Position?ID=REQ1`

Running this query in the browser displays a JSON response.

Tip: The JSON response can be difficult to read. To help make it more readable, try downloading a JSON viewer plugin for your browser.

Detailed steps

1. From Designer Studio select **Resources > Pega API** to open the Pega API landing page.

Note: You may not have the ability to get to the Pega API Landing Page but you can get a listing of what you can do by going to `http://<host>:9080/prweb/api/v1/docs`

2. Click **data** to access the services for accessing data.
3. Click **/data/{ID}** to query a data page.
4. In the value field enter **D_PositionList** this gives you a list of all open positions.
5. Click **Try it out!** to get the data from the data page.

6. When prompted for credentials, use the Username **ssa@tgb** and Password **rules**.
7. Observe the resulting JSON response, you should see that three positions are available.

Bonus

1. Open a new browser window.
2. Enter the following URL:

http://<host>:9080/prweb/api/v1/data/D_Position?ID=REQ1

Note: You need to substitute <host> with the IP Address of your VM.

3. Observe the JSON response.

Reading and writing data to the database

Exercise: Reading and writing data to a database

Scenario

During the onboarding process, new employees are assigned to a seating location in a TGB office. The seating location is either a cubicle or desk in a team room. To ensure that employees are assigned to an available office, business partners in the Human Resources (HR) department must be able to update the list of seating locations each time a location is assigned to an employee.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

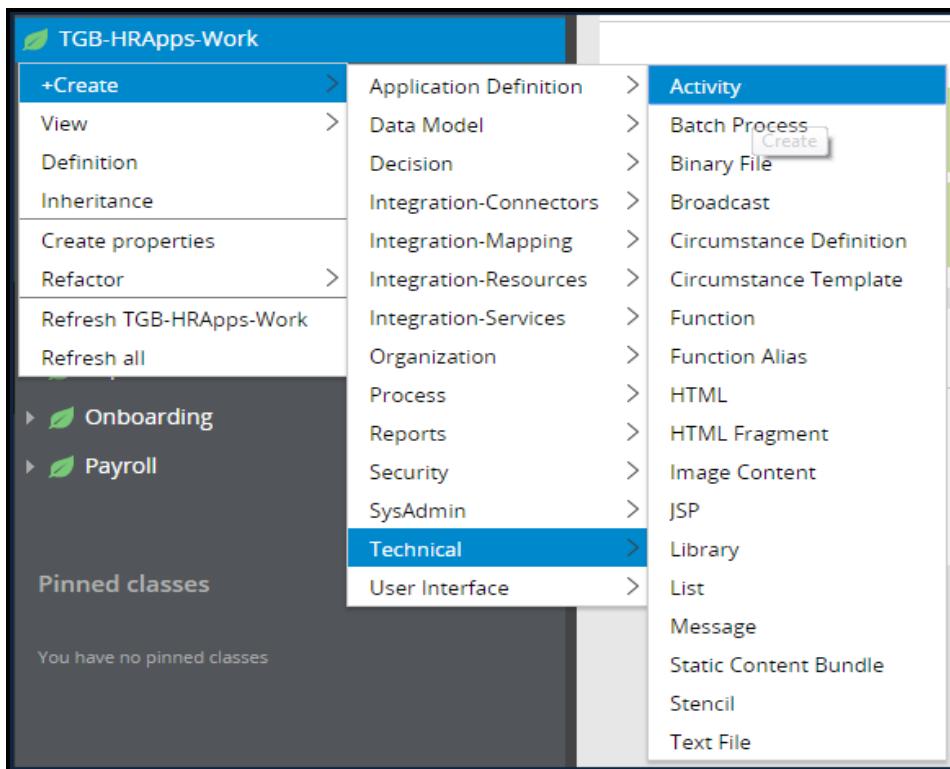
Create an activity that updates the status for a seating location. The status for the seating location is changed to "Unavailable" when the HR business partner selects an available office and submits the select seating location form. The following list includes the recommended tasks for completing this assignment:

- Create an activity to update the seating location status in the local data storage for the seating data type
- Unit test the activity
- Add the activity to the flow action as a post-processing action

Detailed steps

Create an activity to update the status for a given seat

1. Place the activity in the workgroup *TGB-HRAApps-Work* to allow for reuse across the case types. In the App Explorer, right-click the workgroup in the and select: **Create > Technical > Activity**.



2. Name your activity *UpdateSeatingLocation*.
3. Define input parameters for the seating location and status.

This screenshot shows the 'Parameters' tab for the 'UpdateSeatingLocation' activity. The tab has tabs for 'Steps', 'Parameters' (which is selected), 'Pages & Classes', 'Security', 'Specifications', and 'History'. The 'Parameters' section contains a table with two rows:

Name	Description	Data type	Required	In/Out
1 SeatingLocation	Seating location	String	Yes	In
2 Status	Status	String	Yes	In

4. On the **Pages & Classes** tab, define the **SeatingLocation** page.

This screenshot shows the 'Pages & Classes' tab for the 'UpdateSeatingLocation' activity. The tab has tabs for 'Steps', 'Parameters', 'Pages & Classes' (which is selected), 'Security', 'Specifications', and 'History'. The 'Pages & Classes' section contains a table with one row:

Page name	Class
SeatingLocation	TGB-HRApps-Data-Seating

5. Use the **Obj-Open** method to read the seating location from the database.
6. Specify the **SeatingLocation** page as the step page.
7. Select **Lock** so that the object can be saved to the database.
8. Specify the **ID** as the seating location's key with the value of the seating location parameter.

Label	Method	Step page	Description
1. [] Loop When	Obj-Open	SeatingLocation	Open record

Method Parameters

Name	Value
OpenClass	[]
Lock	<input checked="" type="checkbox"/>
ReleaseOnCommit	<input type="checkbox"/>
LockInfoPage	[]
*PropertyName	*PropertyValue
.id	Param.SeatingLocation

9. Use the **Property-Set** method to update the status using the status parameter.

2. [] Loop When	Property-Set	SeatingLocation	Set property values
------------------	--------------	-----------------	---------------------

Method Parameters

*PropertiesName	*PropertiesValue
.status	Param.Status

10. Use the **Obj-Save** method to save the seating location object.

3. [] Loop When	Obj-Save	SeatingLocation	Save record to the database
------------------	----------	-----------------	-----------------------------

Method Parameters

Name	Value
WriteNow	<input type="checkbox"/>
WithErrors	<input type="checkbox"/>
OnlyIfNew	<input type="checkbox"/>

Unit test the activity

1. In the Data Explorer, selecting the **Records** icon to view the seating location records.
2. Identify a seating location record for testing.

The screenshot shows the Pega Data Explorer interface. On the left, there is a sidebar with icons for Recent, Cases, Data, App, Records, and Private. The 'Records' icon is selected. The main area displays a list of data types: Courses, HR Plans, Office, Position, and Seating Locations. 'Seating Locations' is currently selected, highlighted with a blue background. To the right, a detailed view of the 'Seating Locations' data type is shown. The title bar says 'Data Type: Seating Locations' and 'TGB-HRApps-Data-Seating'. The tabs at the top are Data model, Usage, Sources, Records (which is selected), Test cases, and Settings. Below this is a 'Source' section with a dropdown set to 'Local data storage' and an 'Actions' button. A search bar is present. The main table lists seating locations with columns for id*, Status, and office. The row for 'ATL-0203' is highlighted with a red border. The table data is as follows:

id*	Status	office
ATL-0201	Open	Atlanta
ATL-0202	Open	Atlanta
ATL-0203	Open	Atlanta
ATL-0501	Unavailable	Atlanta
ATL-0502	Open	Atlanta

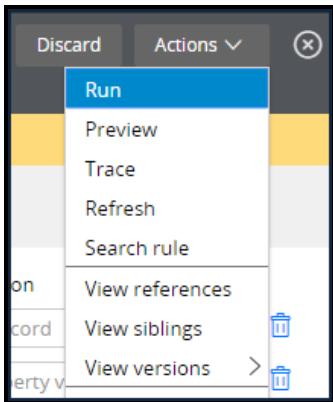
3. Add a commit to the activity to persist the data in the database.

Note: The commit method is only needed when testing the activity. When the activity is invoked in the context of a case, Pega automatically commits the change to the database.

The screenshot shows the activity steps table. It has four columns: Label, Method, Step page, and Description. There are four rows labeled 1 through 4. Row 1: Label is empty, Method is 'Loop When > Obj-Open', Step page is 'SeatingLocation', Description is 'Open record'. Row 2: Label is empty, Method is 'Loop When > Property-Set', Step page is 'SeatingLocation', Description is 'Set property values'. Row 3: Label is empty, Method is 'Loop When > Obj-Save', Step page is 'SeatingLocation', Description is 'Save record to the database'. Row 4: Label is empty, Method is 'Loop When > Commit', Step page is empty, Description is empty. The 'Commit' step in row 4 is highlighted with a red border.

Label	Method	Step page	Description
1.	Loop When > Obj-Open	SeatingLocation	Open record
2.	Loop When > Property-Set	SeatingLocation	Set property values
3.	Loop When > Obj-Save	SeatingLocation	Save record to the database
4.	Loop When > Commit		

4. Select **Actions > Run** to test the activity.



5. Specify the seating location ID identified in step 2. Set the status to **Unavailable**.

6. Click **Run**.

Run Activity: UpdateSeatingLocation

Clipboard Trace Run

CL TGB-HRApps-Work ID UpdateSeatingLocation

Run context

Thread Standard

Page Empty test page

Apply data transform

This activity can only be called from another activity because the "Allow direct invocation from the client or a service" is disabled on the security tab.

Parameter	Value
SeatingLocation*	ATL-0203
Status*	Unavailable

7. Refresh the Records tab to verify that the seating location record was updated.

id*	Status	office
ATL-0201	Open	Atlanta
ATL-0202	Open	Atlanta
ATL-0203	Unavailable	Atlanta
ATL-0501	Unavailable	Atlanta
ATL-0502	Open	Atlanta

8. Remove the commit step in the activity.

Configure the flow action to call the activity as a post-processing action

1. Identify the property and flow action for the seating selection.
 - a. Select **Create > New > Onboarding** to create a new onboarding case.
 - b. Step through the process to the seating location selection screen.
 - c. Open Live UI and identify the property and flow action for the seating location drop-down.

The screenshot shows the Pega Live UI for a seating location selection. At the top, there is a header with the word "Office" and a "Dropdown - .Location" button. Below this is a dropdown menu with the placeholder "Select a seating location...". At the bottom of the screen are three buttons: "Cancel", "Save", and "Submit".

On the right side of the screen, a "Dynamic Layout" tree is displayed. It includes sections for "pyCaseContent", "pyDisplayStages", "pyCaseMessage", "pyCaseActionArea", and "pyCaseActionAreaButtons". A specific section, "Section in cell - non-auto generated", is expanded to show a "Flow Action - SelectSeatingLocation_0" node. This node is further expanded to show a "Section - SelectSeatingLocation_0" node, which contains two "Dynamic Layout" nodes. The first "Dynamic Layout" node contains a "Text Input - .Office" node, which is highlighted with a red box. The second "Dynamic Layout" node contains a "Dropdown - .Location" node, also highlighted with a red box.

Open the *SelectSeatingLocation_0* flow action.

1. On the **Action** tab, configure the *UpdateSeatingLocation* activity to run in the post processing.
2. Specify the *Location* property as the **SeatingLocation** activity parameter.

Set the **Status** activity parameter to "Unavailable".

The screenshot shows the 'Action' tab selected in a configuration interface. The 'Pre-processing' section contains fields for 'Apply data transform' and 'Run activity'. The 'Post-processing' section contains fields for 'Apply cost', 'Apply data transform', 'Run activity' (with 'UpdateSeatingLocation' selected), and a 'Parameters' section. The 'Parameters' section contains two entries: 'SeatingLocation*' with value '.Location' and 'Status*' with value '"Unavailable"'. The 'Status*' entry is highlighted with an orange border.

Layout Validation Action Help setup Security HTML Pages & Classes Specifications

Pre-processing

Occurs before this action is loaded

Apply data transform + Refresh parameters

Run activity + Refresh parameters

Post-processing

Occurs after validation passes and this action is submitted

Apply cost

Apply data transform + Refresh parameters

Run activity + Refresh parameters

Parameters

SeatingLocation*	.Location
Status*	"Unavailable"

Verify your work

To fully test the configuration, run the Onboarding process and populate pyWorkPage with data.

1. Create a new Onboarding request.
2. On the Identify Home Office form, from the Office drop-down list, select a home office.
3. Advance to the Select Seating location step.
4. Verify that the application only returns a list of seating locations for the selected home office.
5. Select a seating location.
6. Verify that the selected seating location is no longer available.

Simulating integration data

Exercise: Simulating integration data

Scenario

As part of each candidate case, TGB performs a background check on the applicant. As part of the background check, a credit report is retrieved from a credit check provider using a SOAP connector. Until recently the WSDL was not available, so the data source was simulated in the *D_CreditReport* data page using the *SimulateCreditCheck* data transform.

With the WSDL available the *CreditCheck* connector was generated using the New SOAP Integration wizard and is ready to be configured for the data page. However, because the service is still not available the connector needs to be simulated.

Note: See lesson **Creating a connector** in the System Architect Essentials (7.2) course for information on how to create a connector.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

The following list includes the recommended tasks for completing this assignment:

- Configuring the *CreditCheck* connector for the *D_CreditReport* data page
- Create a data transform with the test data based on the existing simulation data transform
- Creating a simulation activity for the *CreditCheck* connector
- Verify your work

Simulate the integration with TGB's credit check provider, using the same data that was used when simulating the data source in the data page. The simulation should return one of three individual profiles according to the Taxpayer ID Number (TIN) entered by the candidate.

TIN	Name	Address	City/State	Open accounts	Total balance
111-11-1111	Maria Silva	123 Main St., Apt. 304	Anytown, IL	4	\$5,329
222-22-2222	Edward Grayson	57 Maple Dr.	Mytown, IL	7	\$312,135
333-33-3333	Kim Langston	153 Woodland Terr.	Homeville, IL	9	\$257,394

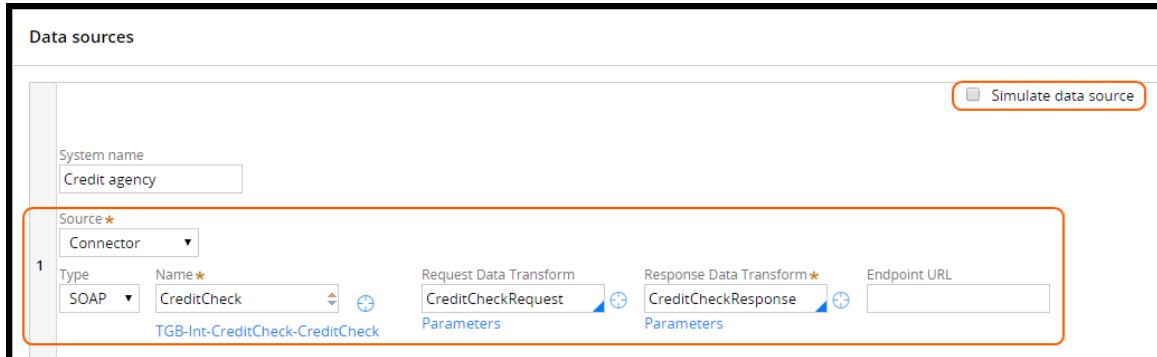
Detailed steps

Source the D_CreditReport data page with the CheckCredit connector

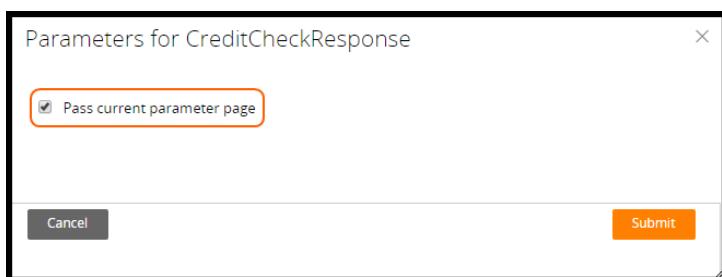
1. Open the *D_CreditReport* data page.

Tip: In the Data Explorer, expand the *Credit report* data type and click the *Credit report* data page to open it.

2. Disable simulation and configure the data page to be sourced by the *CreditCheck* connector.



3. Click the **Parameters** link and select **Pass current parameter page** to verify that the TIN is passed to the request and response data transforms.



Create data transform with the test data

1. Save the *SimulateCreditCheck* data transform in the *TGB-Data-CreditReport* class to the *TGB-Int-CreditCheck-CreditCheck* class. Copying the existing data transform eliminates the need to re-enter all the data.

Note: When saving the rule, the *TGB-Int-CreditCheck-CreditCheck* class path cannot be selected in the *Apply to:* field smart prompt. The class path must be entered manually.

Due to the new context, the form displays undefined property errors.

This record has 15 error(s) in 15 place(s). Target— Property TGB-Int-CreditCheck-CreditCheck.Address is undefined. Target— Property TGB-Int-CreditCheck-CreditCheck.Name is undefined. Target— Property TGB-Int-CreditCheck-CreditCheck.TotalBalance is undefined. Target— Property TGB-Int-CreditCheck-CreditCheck.OpenAccounts is undefined. Target— Property TGB-Int-CreditCheck-CreditCheck.Name is undefined.					
Definition Parameters Pages & Classes Specifications History					
	Action	Target	Relation	Source	
▼ • 1	When	Param.TIN=="111-11-1111"			
• 1.1	Set	.Name	equal to	"Maria Silva"	
• 1.2	Set	.Address	equal to	"123 Main Str., Apt. 304"	
• 1.3	Set	.CityState	equal to	"Anytown, IL"	
• 1.4	Set	.OpenAccounts	equal to	4	
• 1.5	Set	.TotalBalance	equal to	5329	

2. In step 1, specify the *.CreditCheckRequest.TIN* property value for the When action to test.
3. Between step 1 and the 1.1 Set action, insert an **Update Page** action with the *.CreditCheckResponse.CreditReport* as the target . This action sets the context for setting properties on a target page that is different than the primary page. The undefined property errors are corrected when you save the record.
4. Drag the existing Set actions into the Update Page action.

	Action	Target	Relation
▼ • 1	When	.CreditCheckRequest.TIN=="111-11-1111"	
▼ • 1.1	Update Page	.CreditCheckResponse.CreditReport	
• 1.1.1	Set	.Name	equal to
• 1.1.2	Set	.Address	equal to

5. Following the previous procedure, update steps 2 and 3 for the TIN 222-22-2222 and TIN 333-33-3333 target properties.

Create a simulation activity for the CheckCredit connector

1. Select **Designer Studio > Integration > Connectors > Connector Definitions & Simulations** to open the connector simulation landing page.
2. Expand the **SOAP connectors** section.
3. Identify the *TGB-Int-CreditCheck-CreditCheck.CreditCheck* connector.
4. Click the **Simulations** link.

- Data-Admin-Security-SSO-SAML.pySOAPBindingSLO
0 Simulations
- TGB-Int-CreditCheck-CreditCheck.CreditCheck
0 Simulations
- Pega-Search-Manager.pyLPSearchManagerConnectLuceneReIndexStatus
0 Simulations

5. Click the **+** icon to add a simulation activity.
6. In the **SIMULATION ACTIVITY** field, enter **CreditCheckSim** as the name of the simulation activity and then click the **Crosshair** icon to create it.

Simulations for SOAP Connector TGB-Int-CreditCheck-CreditCheck.CreditCheck

SIMULATION ACTIVITY GLOBAL USER SESSION

CreditCheckSim +

7. Create the simulation activity in the same class and ruleset as the connector it simulates.

Activity Record Configuration

Label ★ CreditCheckSim Identifier CreditCheckSim Edit

A short description or title for this record

View additional configuration options

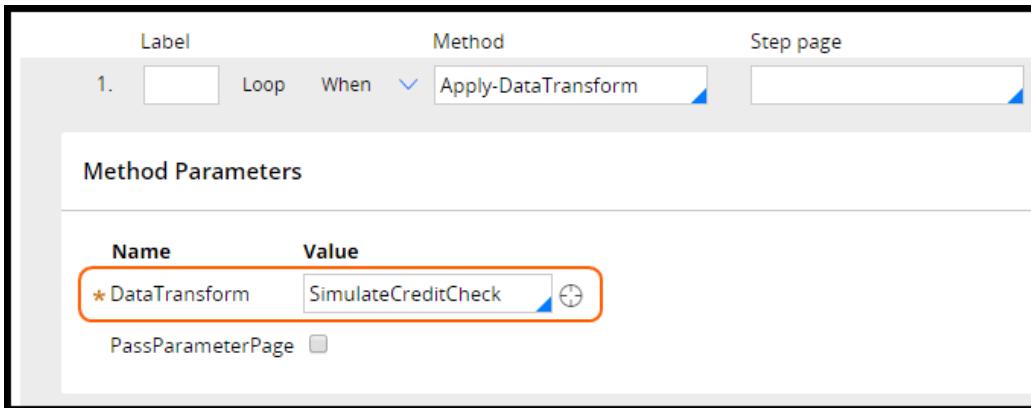
Context

Development branch [No branch]

Apply to ★ TGB-Int-CreditCheck-CreditCheck

Add to ruleset ★ CreditCheck 01-01-01

8. Apply the data transform in the simulation activity.



Verify your work

1. Create a new candidate case.
2. Enter one of the test TINs (111-11-1111 | 222-22-2222 | 333-33-3333).
3. Step through the candidate case to the background check screen.
4. Verify that the test data was returned by the connector simulation.

Addressing integration errors in connectors

Exercise: Configuring error handling for data pages

Scenario

As part of each candidate case, TGB performs a background check on the applicant. As part of the background check, a credit report is retrieved from a credit check provider using a SOAP connector.

If something goes wrong when the SOAP connector is invoked, TGB wants to log errors and display the message "Credit check failed" to the user.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

The recommended tasks for completing this assignment are:

- Configuring the *CreditCheck* connector for the *D_CreditReport* data page
- Detecting errors and invoke a reusable error handling data transform from the response data transform
- Verifying that errors are detected and handled

Detailed steps

Source the *D_CreditReport* data page with the *CheckCredit* connector

- Open the *D_CreditReport* data page.

Note: In the Data Explorer, Expand the *Credit report* data type and click *Credit report* data page to open it.

2. Disable simulation and configure the data page to be sourced by the *CreditCheck* connector.

The screenshot shows the 'Data sources' configuration screen. A system named 'Credit agency' is selected. Under the 'Source' section, a connector named 'CreditCheck' is configured with a Type of 'SOAP'. Request and Response Data Transforms are set to 'CreditCheckRequest' and 'CreditCheckResponse' respectively. The 'Endpoint URL' field is empty. A 'Simulate data source' button is visible at the top right.

3. Click the **Parameters** link and verify that **Pass current parameter page** is selected to ensure that the TIN is passed to the request and response data transforms.

The screenshot shows a dialog box titled 'Parameters for CreditCheckResponse'. It contains a single checkbox labeled 'Pass current parameter page' which is checked. There are 'Cancel' and 'Submit' buttons at the bottom.

Detect and handle errors in the response data transform

1. Open the *CreditCheckResponse* data transform.
2. Enable step 2 to ensure that the *pxDataPageHasErrors* when rule is invoked to check for errors.

The screenshot shows the configuration of a 'When' rule for the 'pxDataPageHasErrors' condition. Step 2.1 is a comment: 'Call template Data transform for automatic error handling of data pages'. Step 2.2 applies the 'pxErrorHandlingTemplate' data transform.

3. Open the standard *pxErrorHandlingTemplate* data transform.
4. Create a copy of the *pxErrorHandlingTemplate* data transform called *ErrorHandlingMaster*.
5. On the **Parameters** tab, define a parameter named *ErrorMessage* for the error message. Remove the other parameters.

Definition	Parameters	Pages & Classes	Specifications	History
Parameters				
Name	Description	Data type	Required	In/Out
1 ErrorMessage		String	Yes	In
(+)				

6. On the **Definition** tab, get messages using the `pxGetPageMessages` function.
7. Use the `pxLogMessage` function to write messages in the log file.
8. Use the `pxClearMessages` function to clear the messages on the data page before adding your custom error message.
9. Use the `pxAddMessageToPage` function to add the `ErrorMessage` input parameter as a message.

Definition	Parameters	Specifications	History
Action	Target	Relation	Source
• 1 Comment	Log the original message for error level in pega logs.		
6 • 2 Set	param.getMessages	equal to	@(Pega-RULES:Default).pxGetPageMessages()
7 • 3 Set	param.logMessage	equal to	@(Pega-RULES:Default).pxLogMessage(param.getMessages)
• 4 Comment	Removes all messages from the primary page.		
8 • 5 Set	param.clearMessage	equal to	@(Pega-RULES:Default).pxClearMessages()
• 6 Comment	Associates a message with the primary page and inv:		
9 • 7 Set	param.addMessage	equal to	@(Pega-RULES:Default).pxAddMessageToPage(Param.ErrorMessage)

Note: You always need to specify a target, even if the function doesn't return anything or if you are not interested in the response.

10. Save the `ErrorHandlingMaster` data transform.
11. Configure the `CreditCheckResponse` data transform to call the `ErrorHandlingMaster` data transform if an error occurs.

12. Click the **Gear** icon to specify the input parameters for the *ErrorHandlingMaster* data transform.

Definition				
	Action	Target	Relation	Source
• 1	Update Page	Primary	with values from	DataSource
• 1.1	Set	.Name	equal to	.CreditCheckResponse.CreditR
• 1.2	Set	.Address	equal to	.CreditCheckResponse.CreditR
• 1.3	Set	.CityState	equal to	.CreditCheckResponse.CreditR
• 1.4	Set	.OpenAccounts	equal to	.CreditCheckResponse.CreditR
• 1.5	Set	.TotalBalance	equal to	.CreditCheckResponse.CreditR
• 2	When	pxDataPageHasErrors		
• 2.1	Apply Data	ErrorHandlingMaster		

13. In the **ErrorMessage** input parameter field, enter **Credit check failed**.

Apply Data Transform - Parameters

Data Transform: ErrorHandlingMaster Class: @baseclass

Pass Current Parameter Page?

Parameter	Value
ErrorMessage*	"Credit check failed"

Cancel **Submit**

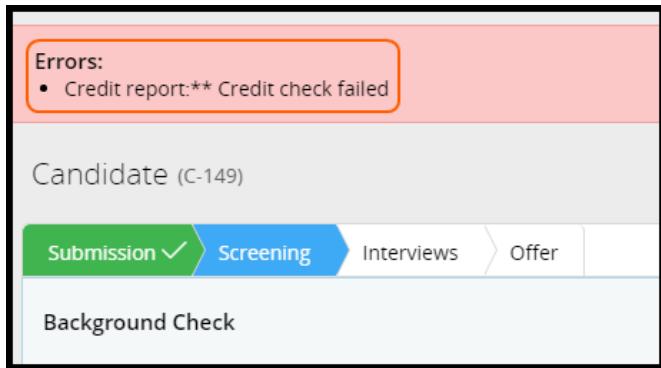
Note: For more information about functions, see the developer help topic [About Function rules](#).

Verify your work

Note: Simulation might be enabled if you completed the Configuring data access patterns exercise prior to this exercise. Disable simulation for the CreditCheck connector before proceeding.

1. Create a new candidate case.
2. Step through the candidate case to the background check screen.

3. Verify that the error message is shown.



Managing integration settings

Exercise: Managing integration settings

Scenario

During development, the credit check integration points to a system with test data. As the application is migrated from a development environment to a test environment and then to a production environment, the endpoint URL needs to be updated.

The table below provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Using a global resource setting to simplify maintenance of the integration settings, implement the endpoint URL for the credit check integration. As the HRApps application is migrated from development to production, the endpoint URL can be updated without unlocking an application ruleset.

To implement the endpoint URL using a global resource setting, you:

- Create the class TGB-Int-CreditCheck-Env to store a page property containing the endpoint URL for the Credit Check connector
- Create a dynamic system setting record to record the value of the endpoint URL

Note: Dynamic system settings are data records, rather than rules. Since data records are not versioned, they are not part of a ruleset. Therefore, they are editable on a production system.

- Create a data transform to set the endpoint URL using a function and dynamic system setting
- Create a data page sourced by the data transform
- Reference the endpoint URL on the Credit Check connect SOAP rule

Detailed steps

Create a data structure for global resource setting data

1. In the App explorer, navigate to the TGB-Int-CreditCheck class.
2. Right-click **SysAdmin** and select **+Create > Class** to display the Create Class form.
3. In the **Label** and **Class Name** fields, enter **TGB-Int-CreditCheck-Env**.
4. In the **Add to ruleset** drop-down list, select **CreditCheck**.

- Click **Create and open** to create the class record.
- On the class record form, in the **Created in version** field, enter or select the highest ruleset version listed.

Note: If you completed the *Creating a new application version* exercise, the highest version should be **01-01-02**. Otherwise, the highest version should be **01-01-01**.

- Click the **History** tab and enter a *Description* of the class and *Usage* notes.
- Click **Save** to update the class record.
- Refresh the Application Explorer to display the class you created.
- Create a single page property named **CreditCheck** in the **TGB-Int-CreditCheck-Env** class.

Tip: Property records are in the Data Model category. From the App Explorer, expand **Env > Data Model > Property**. Right-click **Property** and click **Create**. Then, configure the property mode as single page by setting the property type on the **General** tab of the property record.

- On the **General** tab, in the **Page definition** field, enter **Embed-Env-Connect-SOAP**.
- Click **Save** to complete your configuration of the property record.

Create a dynamic system setting record for the endpoint URL

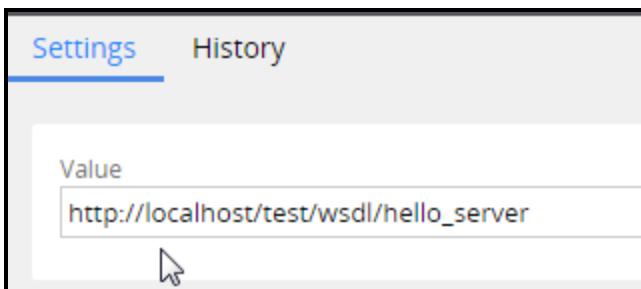
- Right-click the **TGB-Int-CreditCheck-Env** class and select **+Create > SysAdmin > Dynamic System Settings** to open the Create Dynamic System Settings form.
- In the **Short Description** field, enter **Credit check**.
- In the **Owning Ruleset** field, enter **CreditCheck**.
- In the **Setting Purpose** field, enter **GRS/CreditCheck/URL**.

Tip: Remember, when creating a dynamic system setting record, be sure to provide a unique name. In this case, the name you enter consists of three pieces: a namespace (*GRS*), the name of the interface (*CreditCheck*), and the name of the property (*URL*).

- Click **Create and open** to display the Dynamic System Settings record.
- In the **Value** field, enter the URL for the Credit Check connector on your exercise environment: http://localhost/test/wsdl/hello_server.

Note: This URL is listed on the Connect SOAP record for the connector. This record is located in the **TGB-Int-CreditCheck-CreditCheck** class. Expand **CreditCheck > Integration-Connectors > ConnectSOAP**, then open the **CreditCheck** record. Copy the value in the **Service endpoint URL** field.

- Click **Save** to set the value of the dynamic system setting.



Create a data transform to set values for the global resource setting

1. In the **TGB-Int-CreditCheck-Env** class, create a data transform named **ResourceSettings**.

Tip: Data transform records are stored in the Data Model category. From the App Explorer, select **Env > Data Model > Data Transform**. Right-click **Data Transform** and click **Create**.

2. On the **Definition** tab of the ResourceSettings data transform, in the **Target** field, enter or select **.CreditCheck.pyEndpointURL** to set the value of the endpoint URL on the page you created earlier in the exercise.
3. Click the **Gear** icon to the right of the **Source** field to open the Expression builder.
4. Click **Browse** to display the navigation pane for the Expression builder.
5. In the navigation pane, select **Function > Advanced** and select the **Plus sign** icon to the right of **getDataSystemSetting** to add the function to the content pane of the Expression builder.
6. In the content pane, enter the two arguments for the function, **"CreditCheck","GRS/CreditCheck/URL"** and close the parenthesis for the function call.
7. Click **Submit** to close the Expression builder and return to the data transform.

The screenshot shows the 'Definition' tab of a data transform configuration. The table has columns for Action, Target, Relation, and Source. There is one row with the following values: Action is 'Set', Target is '.CreditCheck.pyEndpointURL', Relation is 'equal to', and Source is '@getDataSystemSetting("CreditCheck","GRS/CreditCheck/URL")'. Below the table are buttons for 'Collapse All' and 'Expand All', and a checked checkbox for 'Call superclass data transform'.

Action	Target	Relation	Source
• 1 Set	.CreditCheck.pyEndpointURL	equal to	@getDataSystemSetting("CreditCheck","GRS/CreditCheck/URL")

+ Collapse All Expand All
 Call superclass data transform

8. Click **Save** to complete your configuration of the data transform.

Create a data page sourced by the data transform

1. In the class **TGB-Int-CreditCheck-Env**, create a data page with the name **D_CreditCheckEnv**.

Tip: Data page records are in the Data Model category. From the App Explorer, select **Env > Data Model > Data Page**. Right-click **Data Page** and click **Create**.

Data page definition

Structure
Page ▾

Object type *
TGB-Int-CreditCheck-Env

Edit mode
Read Only ▾

Scope
Thread ▾

2. Configure the data page to reference the data transform you created earlier as the source.

Data sources

System name
Global resource setting

1 Source * Data transform name *
Data Transform ▾ ResourceSettings +
Parameters

Add New Source

A screenshot of the 'Data sources' configuration screen. It shows a single row with the number '1'. The 'Source *' dropdown is set to 'Data Transform' and the 'Data transform name *' dropdown is set to 'ResourceSettings'. There is also a '+' button and a 'Parameters' link. At the bottom left is a 'Add New Source' button with a cursor pointing at it. On the right side of the row, there is a checkbox labeled 'Simulate data source'.

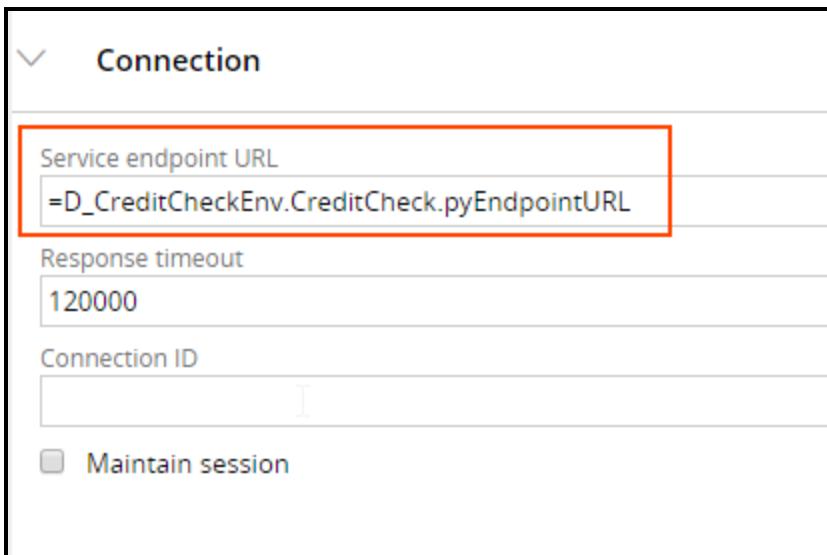
3. Click **Save** to complete your configuration of the data page.

Reference the global resource setting in the connector

1. Open the *CreditCheck* connect SOAP record.

Tip: Connect SOAP records are located in the Connect SOAP category. From the App Explorer, expand **TGB-Int-CreditCheck > Integration-Connectors > Connect SOAP**.

- Under Connection, in the **Service endpoint URL** field, replace the existing endpoint URL with a reference to the global resource setting you created, `=D_CreditCheckEnv.CreditCheck.pyEndpointURL`.



The screenshot shows a configuration interface for a 'Connection'. At the top, there's a section labeled 'Connection'. Below it, the 'Service endpoint URL' field is highlighted with a red border and contains the value `=D_CreditCheckEnv.CreditCheck.pyEndpointURL`. There are other fields like 'Response timeout' (set to 120000), 'Connection ID' (empty), and a 'Maintain session' checkbox.

Note: The syntax for the reference consists of three parts: (1) the name of the data page preceded by an equals sign (`=D_CreditCheckEnv`), (2) the name of the page property that contains the configuration property (`CreditCheck`), and (3) the name of the configuration property (`pyEndpointURL`).

- Click **Save** to complete your configuration of the connect SOAP record.

Verify your work

(Start typing here)

APPLICATION DEBUGGING

Reviewing log files

Exercise: Reviewing log files

Scenario

You have been asked to examine the log files for errors generated by the HRApps application. To practice reading the log files, create a test message and add the message to the PEGA log.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Create an activity that uses the *Log-Message* method to write a message to the PegaRULES log file. Apply the activity to the *Identify_Employee_0* flow action for the *EmployeeEvaluation* case type as a post-processing action. Then, view the log file generated by the Pega server to locate your message.

Detailed steps

Create an activity to add a message to the log file

1. In the **EmployeeEvaluation** class, create an activity record named **WriteToLog**.

Tip: To create an activity record, in the Application Explorer, right-click the class name and select **+Create > Technical > Activity**.

2. In the first step of the activity, in the **Method** field, select **Log-Message**.
3. Expand the method parameters.
4. In the **Message** field, enter "Test message".
5. In the **LoggingLevel** field, enter **InfoForced**.

Note: By default, the PegaRULES log only records messages with the logging level set to **Error**. Setting the level to **InfoForced** overrides this limitation without changing the logging level setting for the server.

- Save the activity.

The screenshot shows the 'Steps' tab in the Pega Designer Studio interface. A single step is selected, labeled '1.' with a blank box for 'Label'. The 'Method' dropdown is set to 'Log-Message', which is highlighted with a red box. To the right, there's a 'Step page' dropdown and a 'Description' field containing 'Log a message'. Below the step list is a 'Method Parameters' section. It contains two rows: 'Message' with value '"Test message"' and 'LoggingLevel' with value 'InfoForced'. There are also two unchecked checkboxes: 'GenerateStackTrace' and 'SendToTracer'. At the bottom of the steps area are buttons for '+Add a step' and 'Collapse all steps'.

- In the Employee Evaluation class, open the **Identify_Employee_0** flow action.

Tip: Flow action records are located in the **Process** category in the Application Explorer.

- Click the **Action** tab.
- In the Post-processing section of the **Action** tab, in the **Run activity** field, enter or select **WriteToLog** to add your activity to the flow action as a post-processing action.
- Save the flow action to commit your configuration change.
- Create a new Employee Evaluation case and complete the Identify Employee form to generate an entry in the PEGA log.

View the log file

- From the Designer Studio menu, select **System > Operations > Logs** to open the System: Operations landing page.
- Under **Utility**, click **Log files**. The log viewer opens in a new browser window, listing the log files written by your server.
- Under **File (click to view)**, click the log file for the **PEGA** Appender to open the log in the window.

The screenshot shows a 'Log Files' viewer window. The title bar says 'Log Files'. Below it is a table with two columns: 'Appender File (click to view)' and 'Download'. The table lists several log files:

Appender File (click to view)	Download
ALERTSECURITY /opt/tomcat/work/Catalina/localhost/prweb/PegaRULES-ALERTSECURITY-2016-Nov-15.log	text or zip
BIX /opt/tomcat/work/Catalina/localhost/prweb/PegaBIX-2016-Oct-15.log	text or zip
CLUSTER /opt/tomcat/work/Catalina/localhost/prweb/PegaCLUSTER-2016-Oct-15.log	text or zip
ALERT /opt/tomcat/work/Catalina/localhost/prweb/PegaRULES-ALERT-2016-Nov-16.log	text or zip
PEGA /opt/tomcat/work/Catalina/localhost/prweb/PegaRULES-2016-Nov-16.log	text or zip
(all logs)	zip

 A red box highlights the 'PEGA' row. At the bottom is a 'Close' button.

Tip: To download a copy of the log file, click **text** or **zip** to the right of the log file. When prompted for a user name and password, enter **admin** and **admin**.

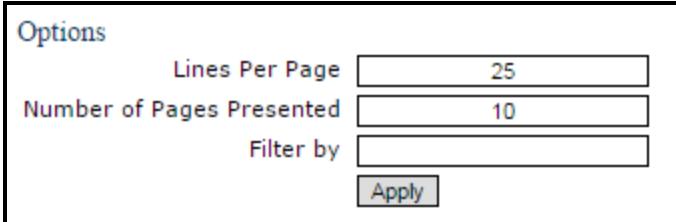
4. In the log file, locate the most recent entry. Confirm that the entry matches the logging level and message from the activity you created earlier in the exercise.



[HRApps:01.01.01] (Work_EmployeeEvaluation.Action) **INFO** 10.61.9.195|10.2.202.62 ssa@TGB - **Test message**

Note: If your log file lists more than one entry, locate the entry with the most recent timestamp.

5. Click **Options** to view the display options for the log file.
6. Clear the contents of the **Filter by** field to display all log events.



Options

Lines Per Page

Number of Pages Presented

Filter by

Apply

7. Click **Apply** to refresh the contents of the log viewer.
8. Above the **Options** link, click the last page of results and locate your message.
9. Close the window containing the log file.
10. Return to the **IdentifyEmployee_0** flow action and remove the activity from the Post-processing section of the **Actions** tab.
11. Save the flow action to prevent the application from writing your test message to the PEGA log.

Tip: To reduce clutter in the Pega log and improve performance, remove activity steps that write InfoForced level messages or convert them to comments when the message is no longer needed.

Analyzing application performance

Exercise: Analyzing application performance

Scenario

Your manager has asked you to test application performance during the development of your application. Your manager wants to know what the performance statistics are for the existing application during a single user session. The statistics include the demands made on processing resources and the system's response time. You must measure the performance for each user interaction with the system. The objective is to establish baseline statistics that help you incrementally identify and address potential performance issues. This approach helps avoid significant rework late in the development phase.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Use the Performance Analyzer (PAL) to report on the performance statistics for each user interaction.

Important: Performance as measured by PAL tends to decrease as rules are added to an application. Focus on the magnitude of these changes to identify potential application issues when you compare the Performance tool results returned from different application states.

First, you create a case and advance it through its entire life cycle. This ensures that rule assembly is performed on any new rules you have created. Rules assembly is a technique for improving performance by generating and compiling the Java code that corresponds to a rule only when needed. You should exclude these statistics when assessing application performance in a run-time environment.

Then, you start the Performance tool and create another case. After you submit a form, you take a reading of the performance statistics. You repeat this process until you have resolved the case.

Note: Remember that no two systems behave identically. The results of your performance testing will likely differ from the sample data shown in the exercise.

Detailed steps

1. On the Developer toolbar, click the **Performance** icon to start the Performance tool. By default, the tool displays PAL.
2. Create an Onboarding case and run the case through to completion.

Note: Remember to first run through a case to ensure that rules assembly has been performed for all of the rules used by the case.

3. Create a new Onboarding case.
4. Go to the Performance tool and, on the header, click **Reset** to clear any readings and initialize all counts to zero..
5. On the header, click **Add reading**. This displays a DELTA row that contains the incremental values between adjacent rows of the Performance report.

FULL	13	13	3.28	0.00	0.00	2.54	0.00	0.01	0.73	0.00	0.00	0.00	0	65	3,570	265	11	0	10	3,686,347
Int #	Int Count	Server Elapsed	RA Elapsed	Rule I/O Elapsed	RDII I/O Elapsed	Connect Elapsed	Other I/O Elapsed	Total CPU	RA CPU	Rule CPU	Other CPU	RA Count	Rule Count	Total Rules Used	Activity Count	RDIIO Count	Connect Count	Other I/O Count	Alert Count	Total Bytes
DELTA	13	13	3.28	0.00	0.00	2.54	0.00	0.01	0.72	0.00	0.00	0.00	0	65	3,574	263	11	0	10	3,686,347
INIT	0	0	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0	0	4	2	0	0	0	0

6. Click the **DELTA** link at the beginning of the row to show the individual items that are summed in each column. You can position your mouse pointer over an individual item to display a tool tip that describes the item.

▼ Elapsed Time Detail For This Requestor (seconds)

- 2.60 Total Elapsed time for the reading
- 0.01 Elapsed time accessing non-rule-resolved instances from database
- 0.01 Elapsed time performing Database Operations
- 0.01 Measures the elapsed time spent when a database operation (such as Delete, Save, or Open) is performed for all non-rule-resolved instances
- 0.00 Elapsed time retrieving Declarative Rules
- 0.00 Elapsed time retrieving non-Rule database lists
- 0.00 Elapsed time processing Storage Streams for non-Rule database lists

7. Return to the Collect Employee Info form and click **Submit**.
8. Return to the Performance tool and click **Add reading** to create another DELTA row. This row shows the resources used when the form was submitted to the system.
9. Continue to advance the case and review the results in the Performance tool until you have completed the case.

As you proceed through your performance analysis, you can save the results to review against later performance tests. On the Performance tool header, click **Save** to download the results to an Excel spreadsheet.

Note: The Performance tool also provides access to DB Trace and the Performance Profiler. For more information, see the help topic [About the Performance tool](#).

APPLICATION ADMINISTRATION

Securing an application

Exercise: Securing an application

Scenario

In the Employee Evaluation case, TGB wants to have an optional, case-wide action to update an employee's goals. Human resources (HR) business partners can run the *UpdateGoals* action at any time. However, employees can only run the *UpdateGoals* action when the case is in the first stage.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules
HR Business Partner	HRPartner@TGB	rules
Employee	User@TGB	rules

Your assignment

Add the *UpdateGoals* flow action to the *Employee Evaluation* case type as an action that users can take throughout the case. Next, create a privilege and add it to the *UpdateGoals* flow action. Then, create an *Access when* condition that returns true when the case is in the first stage. Finally, use the *Access Manager* to add the privilege to HR staff, managers, and employees.

Tip: When creating the *Access When* rule, run a test case then look for the *pxCurrentStage* property on *pyWorkPage* to identify the name of the first stage.

Detailed steps

The following table provides the credentials you need to complete this portion of the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Add the *UpdateGoals* flow action to the employee evaluation case

1. In the **Cases** explorer, open the **Employee Evaluation** case type.
2. On the **Settings** tab, select **Actions**, then add the *UpdateGoals* flow action.

The screenshot shows the 'Edit case type: Employee Evaluation' dialog with the 'Settings' tab selected. Under the 'Actions' section, there is a list of actions: 'pyUpdateCaseDetails', 'pyChangeStage', and 'UpdateGoals'. The 'UpdateGoals' action is highlighted with a red box. There is also a '+ Add action' button.

Create and add a privilege to the *UpdateGoals* flow action

1. Open the *UpdateGoals* flow action.
2. On the **Security** tab in the **Privilege name** field, enter **UpdateGoals**.
3. Click the **Crosshair** to create the privilege.

The screenshot shows the security configuration for the 'UpdateGoals' flow action. The 'Security' tab is selected. In the 'Required privileges' section, a new row is being added, with the 'Privilege class' set to 'TGB-HRApps-Work-EmployeeEvaluation' and the 'Privilege name' set to 'UpdateGoals'. Both fields are highlighted with a red box.

4. Save your changes to the privilege rule form.
5. Save the *UpdateGoals* flow action.

Create an Access When rule for first stage

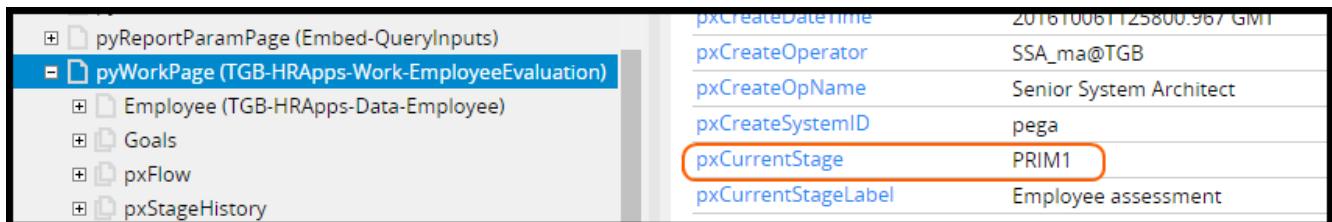
1. Create a new employee evaluation case.

Tip: Select **Create > New > Employee Evaluation** in the Designer Studio header to create a new employee evaluation case.

2. Open the clipboard tool.

Tip: Open the clipboard tool by clicking Clipboard in the Developer Toolbar.

3. Select **User Pages > pyWorkPage** to view the case data.
4. Identify the value of the *pxCurrentStage* property.

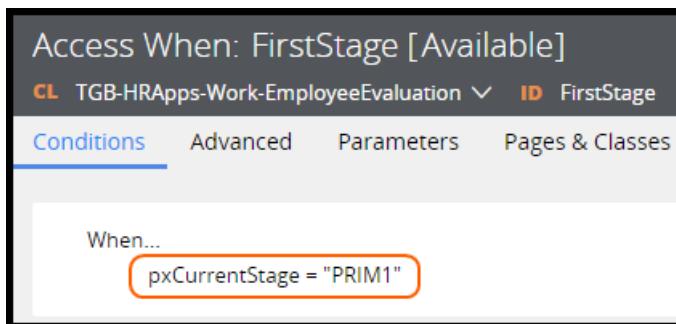


pyReportParamPage (Embed-QueryInputs)	pxCreateDateTime	2016/06/12 23:58:00.967 GMT
pyWorkPage (TGB-HRApps-Work-EmployeeEvaluation)	pxCreateOperator	SSA_ma@TGB
+ Employee (TGB-HRApps-Data-Employee)	pxCreateOpName	Senior System Architect
+ Goals	pxCreateSystemID	pega
+ pxFlow	pxCurrentStage	PRIM1
+ pxStageHistory	pxCurrentStageLabel	Employee assessment

5. Create an *Access When* rule called *FirstStage* in the employee evaluation case type.

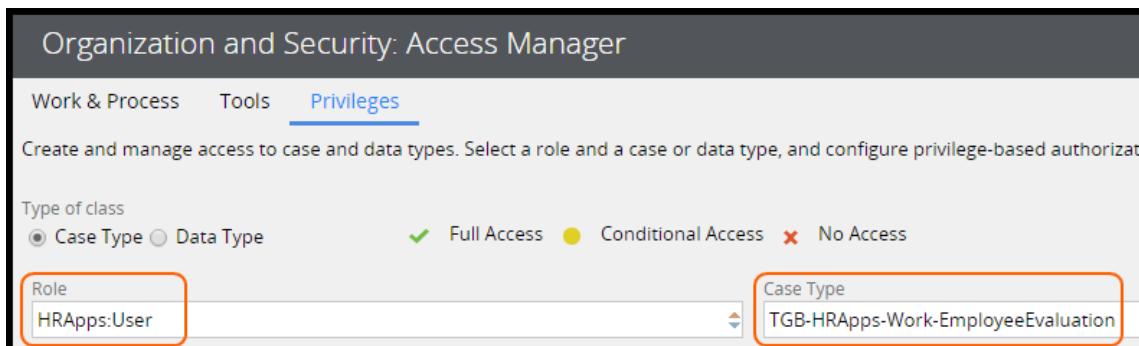
Tip: In the App Explorer, right-click the employee evaluation case type and select **Create > Security > Access When** to create a new access when rule.

6. Return true when *pxCurrentStage* is **PRIM1**.



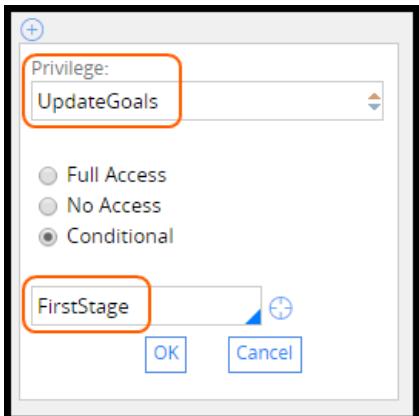
Add and configure the privilege for the roles

1. From the Designer Studio menu, select **Org & Security > Access Manager > Privileges** to open the Privilege tab for the Access Manager.
2. In the **Role** field, enter **HRApps:User**.
3. In the **Case Type** field, enter **TGB-HRApps-Work-EmployeeEvaluation**.



4. Click the plus button to add a privilege.
5. Select the **UpdateGoals** privilege.

6. Select **Conditional**, then enter **FirstStage** in the field.



7. Repeat steps 3 to 5 for the **HRApps:HR** role, and in step 6 select **Full Access**.

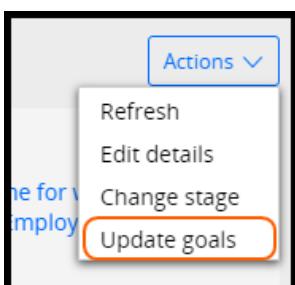
Verify your work

Verify access for HR staff

The following table provides the credentials you need to complete this portion of the exercise.

Role	Operator ID	Password
HR Business Partner	HRPartner@TGB	rules

1. Using the credentials from the table above, log in to the Case Worker portal.
2. Create a new employee evaluation case.
3. Enter a first and last name and click **Submit**.
4. Click the **Actions** button and verify that the **Update goals** action is available.



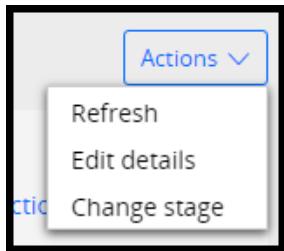
5. Click **Submit** to move the case to the second stage.
6. Click the **Actions** button and verify that the **Update goals** action is still available.

Verify access for employee

The following table provides the credentials you need to complete this portion of the exercise.

Role	Operator ID	Password
Employee	User@TGB	rules

1. Using the credentials from the table above, log in to the Case Worker portal.
2. Create a new employee evaluation case.
3. Enter a first and last name and click **Submit**.
4. Click the **Actions** button and verify that the **Update goals** action is available.
5. Click **Submit** to move the case to the second stage.
6. Click the **Actions** button and verify that the **Update goals** action is not available.



Creating an agent for background processing

Exercise: Configuring batch processing

Scenario

At TGB, employee claims for health insurance benefits are processed by a third party. To ensure that employee claims can be processed in a timely manner, the Human Resources (HR) department wants to queue all of the benefit enrollment cases for transmission to the provider overnight.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	SSA@TGB	rules

Your assignment

Create an agent to process benefit enrollment cases each day at 11:30 P.M. and configure the Review Selections process to queue the case after the case is reviewed.

In order to meet these requirements, you must create a standard agent named Process Benefits Enrollments to run daily at 11:30 P.M. For testing purposes, the agent executes an activity named *ProcessBenefitsEnrollments* to run a report that lists the benefits enrollment cases completed on the current day. The activity is included in the application for the purpose of this exercise.

You must also configure the Review selections process to call the Queue for Agent activity to add the benefits enrollment case to the agent queue.

Detailed steps

Create a standard agent

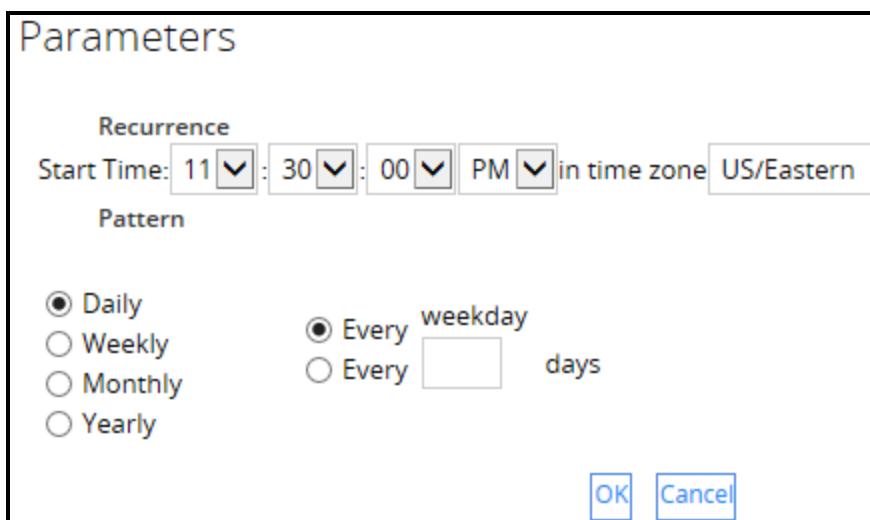
1. In the **Benefits Enrollment** case type, create an agent named **Process Benefits Enrollments**.

Tip: To create the agent, in the Records Explorer, right-click **SysAdmin** and select **+Create > Agents**.

Note: Remember to create the new record in the unlocked HRApps ruleset.

2. On the **Schedule** tab, in the **Agent Name** field, enter **Process Benefits Enrollments**.
3. In the **Pattern** field, select **Recurring**.
4. In the **Interval (sec)** field, select **Advanced**.

- Click the **Advanced** button. This displays a Parameters pop-up dialog.
- In the **Start Time** field, enter 11:30 P.M. Leave the default pattern of **Daily**. These settings indicate that the agent executes and transmits the information at 11:30 P.M. every day. When the dialog is complete, it looks like the following image.



- Click **OK** to save your updates and close the dialog.
- Expand the first row to display additional details.

Scheduled Agents			
	Agent Name	Pattern	Interval (sec)
1	Process Benefits Enrollments	Recurring	
press enter to expand row			

- In the **Class** field, enter **TGB-HRApps-Work-BenefitsEnrollment**.
- In the **Activity name** field, enter **ProcessBenefitsEnrollments**.

Scheduled Agents				
	Agent Name	Pattern	Interval (sec)	
1	Process Benefits Enrollments	Recurring		Advanced
	Class *	Activity name *	Params	Max record
	TGB-HRApps-Work-BenefitsEnrollment	ProcessBenefitsEnrollments	Params	

- On the **Security** tab, select the **Bypass activity authentication** check box. In most situations, agents do not need access groups for authentication.
- Click **Save**.

The agent runs once a day at 11:30 P.M. and executes the *ProcessBenefitsEnrollments* activity.

Add the Queue for Agent activity to the Review selections process

1. From the Cases explorer, open the *Benefits Enrollment* case type.
2. In the *Review Selections* stage, open the *Review Selections* process.
3. Add a Utility shape to the process.
4. Open the Utility properties panel.
5. In the **Utility** field, enter **Queue benefits enrollments**.
6. In the **Rule** field, enter **QueueForAgent**.
7. In the **AgentRuleSet** field, enter **HRApps**.

Note: Enter the name of the name of ruleset manually if HRApps does not display in the select list.

8. In the **AgentName** field, enter **ProcessBenefitsEnrollments**.
9. Click **Submit** to save your updates and close the panel.

10. Reconnect the Review Selections connector to the utility. The process looks like the following image.



Verify your work

From the Designer Studio menu, select **System > Operations > System Management Application** and open the node your exercise system is running on.

Note: For testing, you can reconfigure the agent to run more frequently by configuring a periodic interval of 100 seconds.

Reconfigure the agent

1. Open the Process Benefits Enrollment agent.
2. In the **Pattern** field, change the setting from **Recurring** to **Periodic**.
3. In the **Interval (sec)** field, update the value to **300**.
4. Click **Save** to update the agent configuration.

Test the agent

1. From the Designer Studio menu, select **System > Operations > System Management Application** and open the node your exercise system is running on.
2. On the left panel, expand the Agent Management menu and select **Agents**.
3. On the right panel, locate your agent. The agent should be enabled (a green check mark appears in the **Runnable?** column).

4. Verify that the difference between the **Last run finish** and **Next run time** values is 100 seconds.

Runnable?	Running Now	RuleSet	#	Description	Mode	Scheduling	Last run start	Last run finish	Next run time
<input type="radio"/>		HRApps		Process Benefits Enrollment	Standard	Every 100 s	10/6/16 11:56:15 AM EDT	10/6/16 11:56:15 AM EDT	10/6/16 11:57:55 AM EDT
							10/6/16 1:00:00 AM	10/6/16 1:00:00 AM	10/7/16 1:00:00 AM

Migrating an application

Exercise: Migrating an application

Scenario

Now that development of the HRApps application is complete, you have been asked to migrate the application to a testing system. The quality assurance (QA testers) require that you provide them with an application archive of the HRApps application. The QA testers will import the application archive on the testing system to test the application ahead of application delivery. Your application contains a ruleset for delegating rules in a production environment, which must be kept unlocked.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Senior System Architect	ssa@tgb	rules

Your assignment

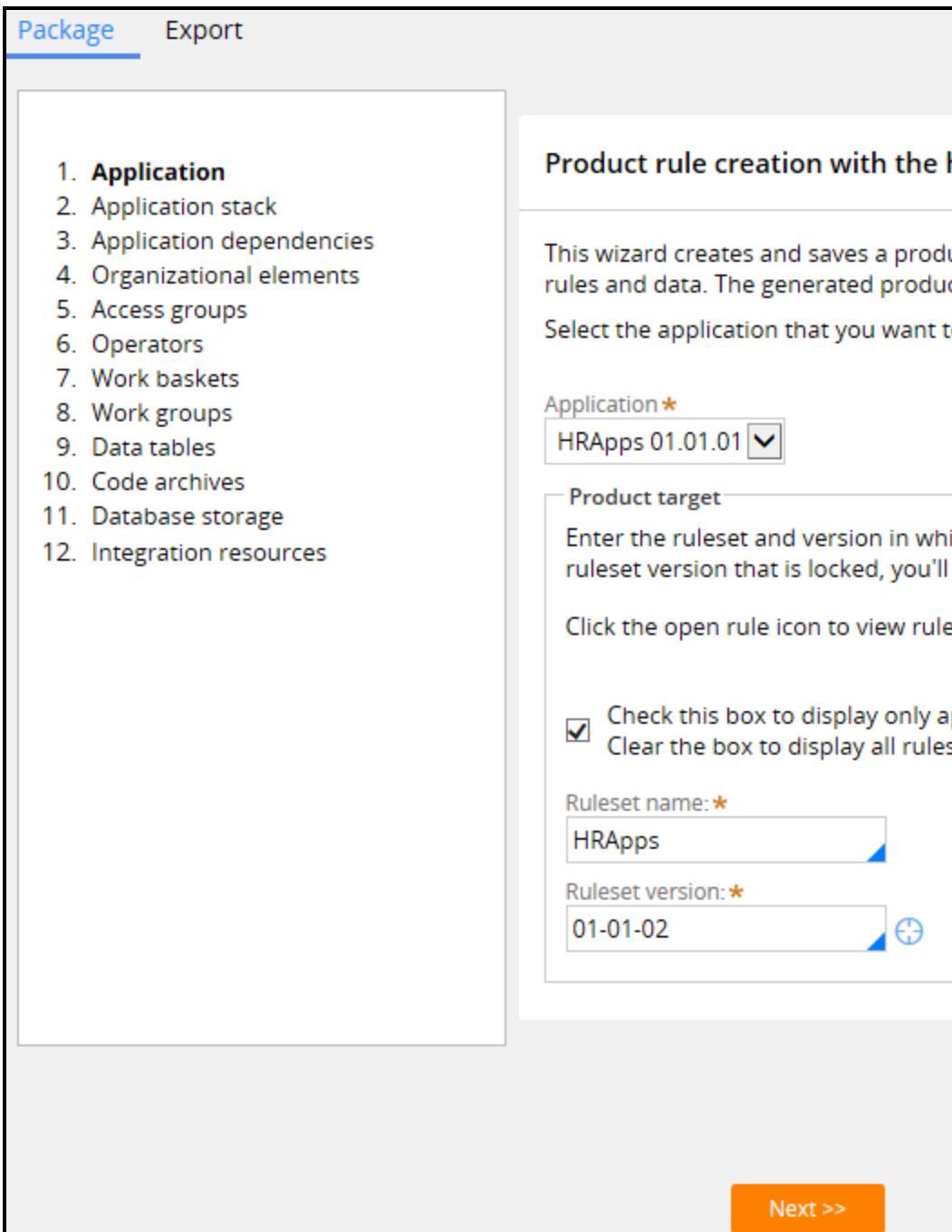
- Use the Application Packaging wizard to create a product rule for the current version of the HRApps application. Include the product rule in an unlocked application ruleset. The product rule includes all of the associated data types and data instances and the built-on application (not PegaRULES). The rule also includes all the developer documentation added in the rule History tabs.
- Modify the product rule so that the delegated production ruleset is included in the archive. Then, verify that contents of the product rule are complete before creating the archive. Finally, use the product rule to create and save the application archive to your system.

Detailed steps

Create the product rule

1. From the Designer Studio menu, select **Application > Distribution > Package** to open the Application Packaging wizard.

2. In the **1. Application** step, select the highest version of the HRApps application, and then select the highest unlocked version of the HRApps ruleset.



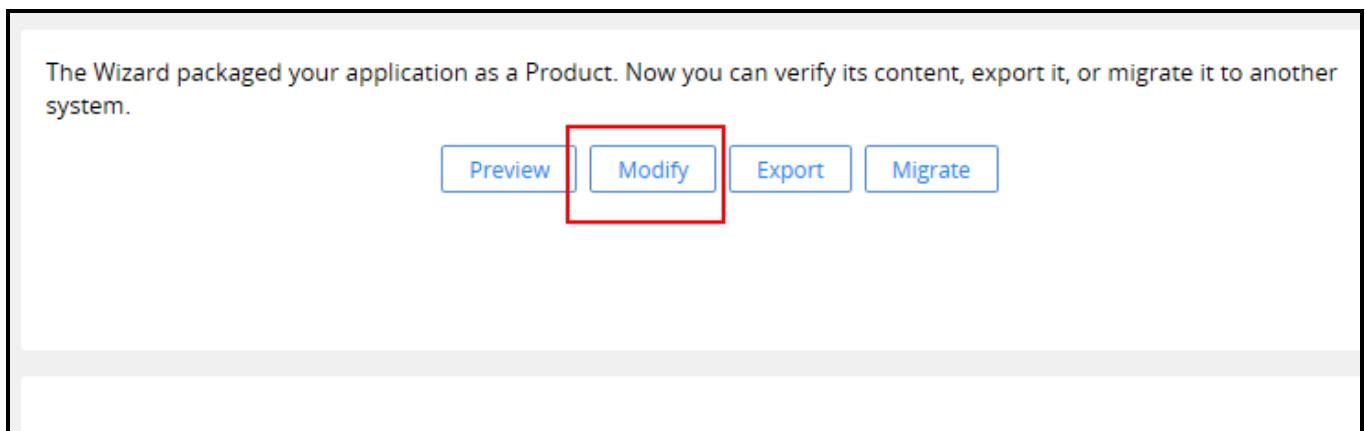
3. Click **Next** to advance to the next step.
4. In the **2. Application stack** step, accept the default selections and click **Next** to include both your implementation and the built-on application, HRFW.
5. In the **3. Application dependencies** step, leave the field blank and click **Next**.
6. In the **4. Organizational elements** step, leave the default selections and click **Next**.
7. In the **5. Access groups** step, leave the default selections and click **Next**.
8. In the **6. Operators** step, leave the default selections and click **Next**.

9. In the **7. Work baskets** step, leave the default selections and click **Next**.
10. In the **8. Work groups** step, leave the default selections and click **Next**.
11. In the **9. Data tables** step, leave the default selections and click **Next**.
12. In the **10. Code archives** step, leave the field blank and click **Next**.
13. In the **11. Database storage** step, leave the default selections and click **Next**.
14. In the **12. Integration resources** step, leave the default selections and click **Finish**.

The wizard creates a product rule in the ruleset you specified.

Modify the product rule

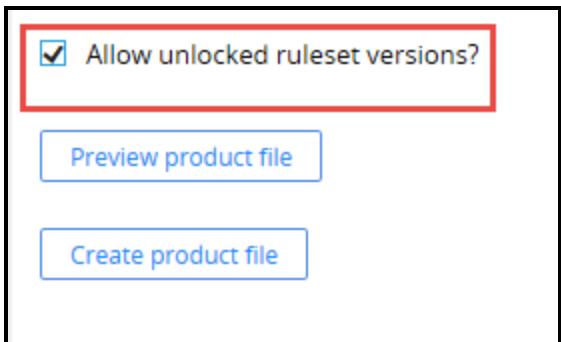
1. When you click **Finish**, the following landing page is displayed. Click **Modify** to open the product rule.



2. Select the **Custom/Production rulesets** check box so that the delegated ruleset is added to the application archive. Do not modify the other default settings. This ensures that data associated with the application rulesets, rule history instances, and the application's data types are added to the archive.

Include associated data	Custom/Production rulesets	Shared/Component rulesets	Delta mode	Include rule history	Include data types
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

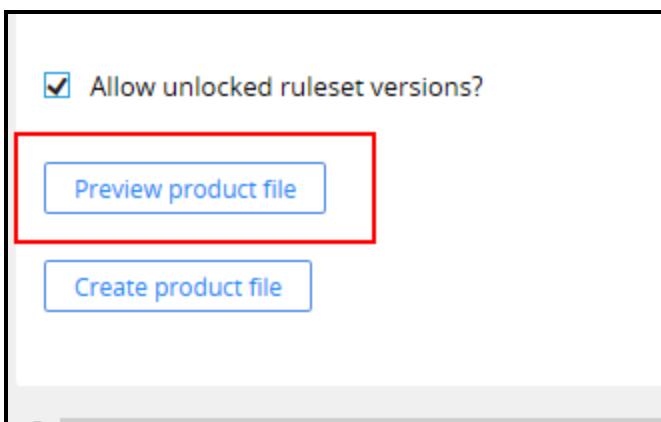
3. Select the **Allow unlocked ruleset versions?** check box at the bottom of the form. Since the delegated ruleset must remain unlocked, selecting this option ensures that an error does not occur when you attempt to create the archive.



4. Click **Save**. Your product rule is now ready to be packaged into an application archive.

Verify your work

1. Click **Preview product file** to review and verify the rules, data, and other objects that you specified are packaged in the application archive.



In the following example, under the Instances column, you can click the number to verify the data instances. Your results may vary depending on how you completed the exercises.

Product Preview

Class	Instances
▼ Data-	319
▼ Data-Admin-	58
► Data-Admin-DB-	10
▼ Data-Admin-Operator-	21
Data-Admin-Operator-AccessGroup	<u>9</u>
Data-Admin-Operator-ID	<u>12</u>
Data-Admin-OrgDivision	<u>2</u>
Data-Admin-OrgUnit	<u>2</u>
Data-Admin-Organization	<u>1</u>
► Data-Admin-System-	1
Data-Admin-WorkBasket	<u>18</u>
Data-Admin-WorkGroup	<u>3</u>
► Data-Rule-	10
► Data-Tag-	251
► History-	1369
▼ Schema	(8)
Class - Table	Instances
► Database Schema Changes	8

Cancel **Submit**

2. Click **Submit** to close the dialog.

3. On the landing page, click **Export** to start the archive process.

The Wizard packaged your application as a Product. Now you can verify its content, export it, or system.

Preview

Modify

Export

Migrate

This displays the Create Product File dialog.

4. Enter a name for the application archive.

Create Product File

Please enter the desired name (of jar/zip type) for the product to be created on the server. Once crea

HRApps01-01-01QA

OK

Cancel

5. Click **OK**. This starts the archive process and displays a progress dialog. The number of records may vary depending on how you completed the exercises.

[Archive created. Click here to save HRApps01-01-01QA.zip file locally](#)

100%

Time Left: 0 seconds

Total Records: 2811

Records Exported: 2811

Records Skipped: 0

[Total Errors: 0](#)

6. When the process is finished (100%), click the **Archive created** link to save the ZIP file to your system. The archive is now available to be imported into the QA system.

DECISION DESIGN

Inside Pega decision management

Exercise: Update the prioritization for a strategy

Scenario

Management wants to change the prioritization so that call center agents have access to all product offers, rather than only the top three offers.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Strategy Designer	StrategyDesigner@WhatIsInTheBox	rules

Your assignment

Modify the Next-Best-Action strategy so that it returns all offers. Then test and verify the change.

Detailed steps

Step through the Top Offers case

1. Create a Top Offers case.

Tip: Select **Run > TopOffers** to create a top offers case.

2. Select **Louise Simpson** for **Customer** and **Product offer** for the **Call Reason**.

The screenshot shows a user interface for creating a 'Top Offers' case. At the top, there are tabs for 'Assignment', 'Information', and 'Audit'. The 'Assignment' tab is active. Below the tabs, the case name 'Top Offers' is displayed next to a search bar containing the email 'StrategyDesigner@WhatIsInTheBox'. In the main area, there are two dropdown fields: 'Customer *' which is set to 'Louise Simpson', and 'Call reason *' which is set to 'Product offer'. Both of these fields are highlighted with orange borders.

3. Click **Submit** to proceed.
4. Reject Apple iPad and Samsung Galaxy S offers and accept the Samsung Galaxy Tab offer.

Note: Three offers are listed.

Offered proposition		
Name	Acceptance probability	Outcome
Apple iPad		<input type="radio"/> Accept <input checked="" type="radio"/> Reject
Samsung Galaxy S		<input type="radio"/> Accept <input checked="" type="radio"/> Reject
Samsung Galaxy Tab		<input checked="" type="radio"/> Accept <input type="radio"/> Reject

5. Click **Submit** to proceed.
6. Examine the interaction history for this customer, then click **Submit**.

Interaction results for this customer								
Fact ID	Subject ID	Issue	Group	Proposition Name	Offered on ▾	Operator	Outcome	
3525610817391545456	CE-1	Sales	Tablets	Samsung Galaxy Tab	11/8/16 11:57 PM	StrategyDesigner@WhatIsInTheBox	Accept	
3525610817391545455	CE-1	Sales	Phones	Samsung Galaxy S	11/8/16 11:57 PM	StrategyDesigner@WhatIsInTheBox	Reject	
3525610817391545454	CE-1	Sales	Tablets	Apple iPad	11/8/16 11:57 PM	StrategyDesigner@WhatIsInTheBox	Reject	

7. Click the **cross** button to close the case.

Create a change request

1. On the Dashboard, click **New revision** to create a new revision.

The screenshot shows a dashboard interface. At the top, it says "Dashboard". Below that, a message reads "Welcome back, Strategy Designer" followed by "There are no open revisions." At the bottom, there is an orange button labeled "New revision ...".

2. Select **Create new revision**.
3. Enter name: **Sales policy change**.

4. Enter objectives: **Enable call center agents to have access to all product offers.**

Define the Objectives for this Revision

This revision is assigned to you.

Create new revision
 Base this revision on a previous one

Name *
Sales policy change

Objectives *
Enable call center agents to have access to all product offers

+Add item

5. Click **Continue** to create a change request.
6. Click the **Edit** link to update the Change request.
7. Change the **Name** to Product Offers strategy modifications and the **Description** to Modify Product Offers strategy by changing the prioritization component return all product offers.

Change request [Edit](#)

Name
Product Offers strategy modifications

Assigned to
Me

Objective
Enable call center agents to have access to all product offers

Description
Modify Product Offers strategy by changing the prioritization component return all product offers

Urgency
50

Due by
11/16/2016

8. Click **Submit**.

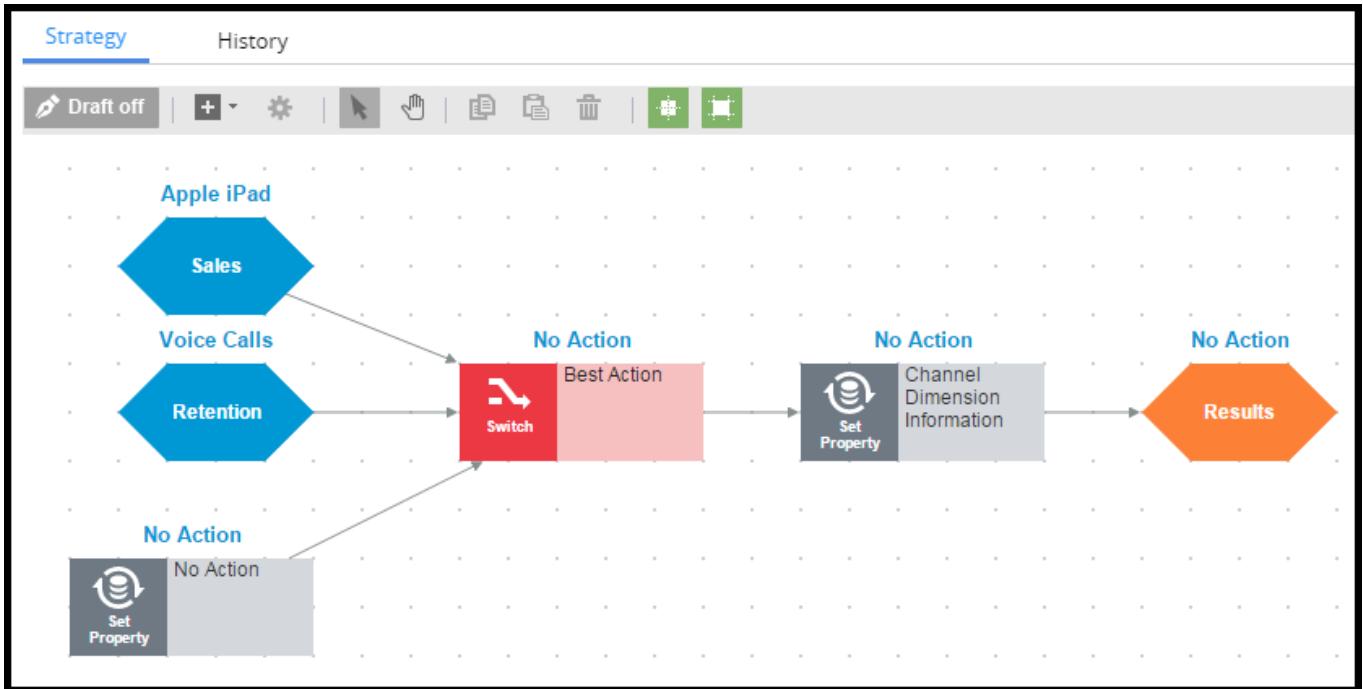
9. Select all Business rules and click **Next**.

DSM Essentials data customer				
	Name	Type	Updated	Updated by
<input checked="" type="checkbox"/>	Next Best Action (Results in SR)	Strategy	1 year ago	DSME Administrator
<input checked="" type="checkbox"/>	Product Offers (Results in Sales)	Strategy	1 year ago	DSME Administrator
<input checked="" type="checkbox"/>	Retention (Results in Retention)	Strategy	1 year ago	DSME Administrator
<input checked="" type="checkbox"/>	Sales Strategy (Results in Sales)	Strategy	1 year ago	DSME Administrator
Retention				
	Name	Type	Updated	Updated by
<input checked="" type="checkbox"/>	Retention (Defined on DSM Essentials data customer)	Strategy	1 year ago	DSME Administrator
Sales				
	Name	Type	Updated	Updated by
<input checked="" type="checkbox"/>	Product Offers (Defined on DSM Essentials data customer)	Strategy	1 year ago	DSME Administrator

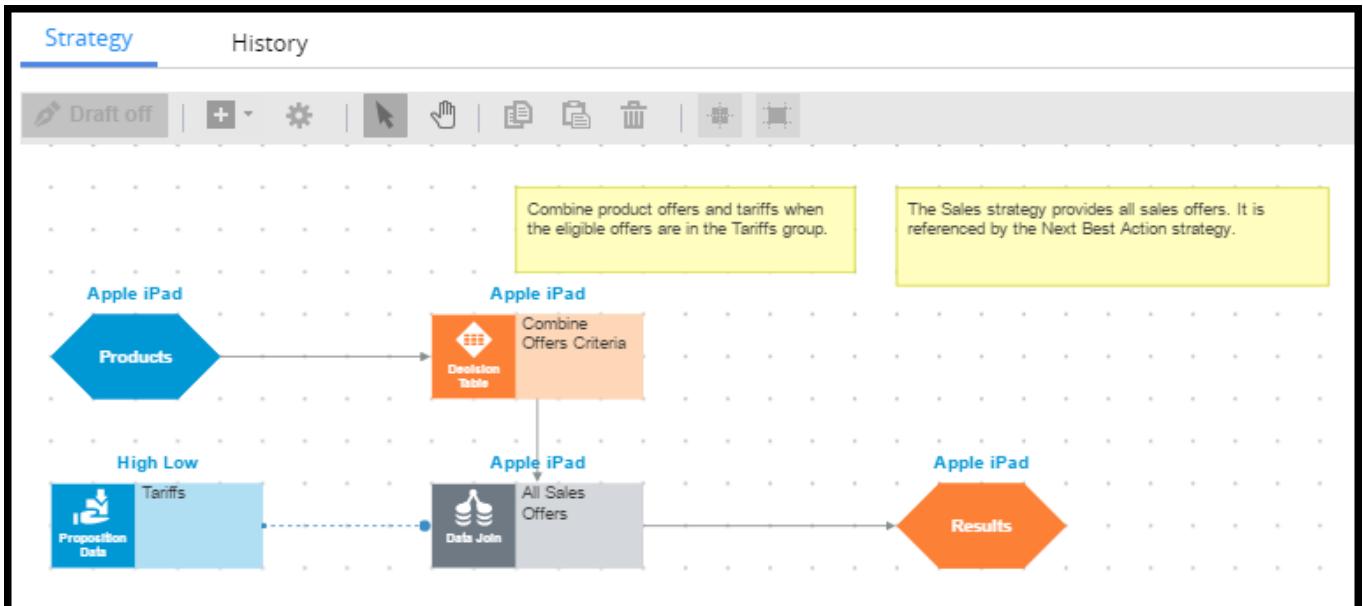
10. Click **Submit** to complete the change request.

Identify the component used to prioritize offers

1. Open the newly created change request.
2. Open the referenced Next-Best-Action strategy.

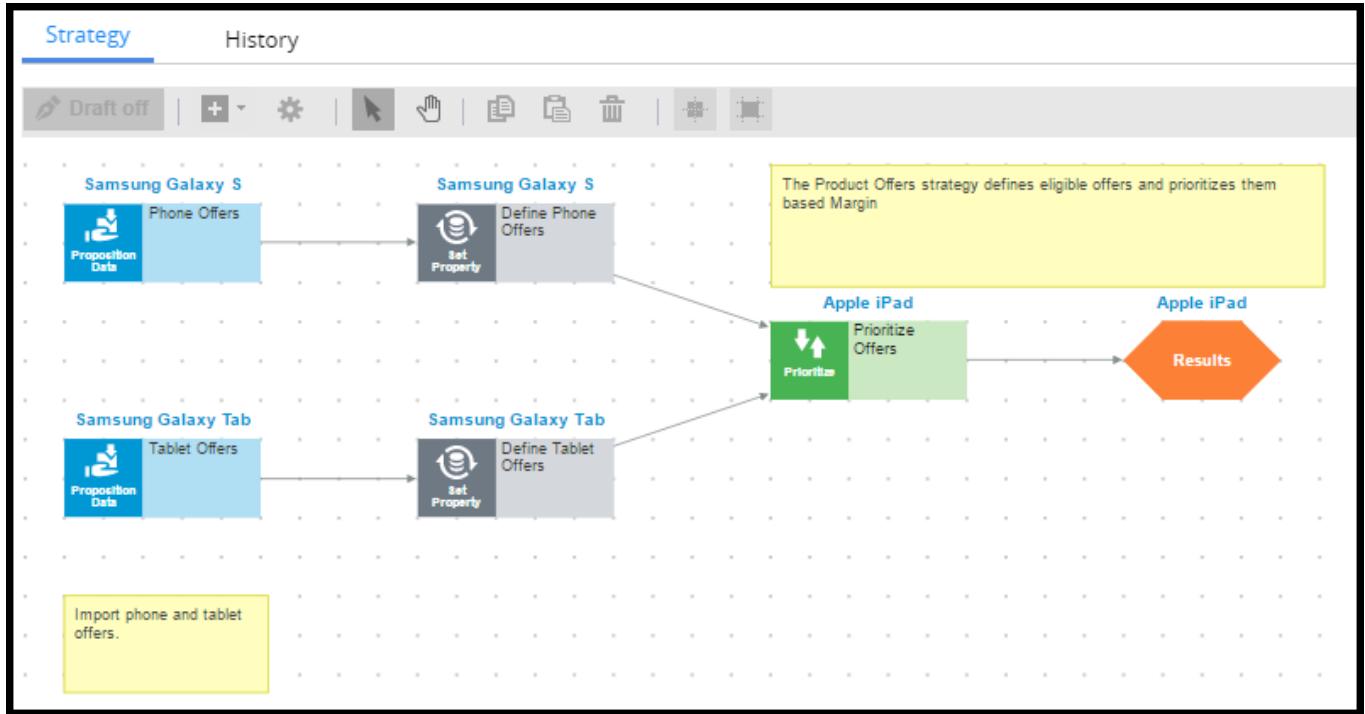


3. Right-click **Sales** shape and select **Open strategy** to drill down into the *Sales* substrategy.



4. Right-click **Product** shape and select **Open strategy** to drill down into the *Products* sub-strategy.

- Identify the **Prioritize** shape component.



- Close the open rules to return to the change request.

Modify numbers of offers displayed and submit the revision

- Click **Product Offers** to open it.

Name	Rule type	Updat...
Next Best Action	Strategy	1 year
Product Offers	Strategy	1 year
Retention	Strategy	1 year
Sales Strategy	Strategy	1 year

- Right-click the **Prioritize** component and select **Properties** to open the properties panel.

3. Change **Output** check box from **Top** to **All**, then click **Submit**.

The screenshot shows the 'Prioritize Properties' dialog box. It includes fields for Name (Prioritize Offers), Component ID (PrioritizeOffers), and Description (radio buttons for 'Use generated' and 'Use custom'). Below these are sections for 'Source components' (with 'Prioritize' selected) and 'Expression' (.Margin). Under 'Order by', the 'Highest first (9 to 1)' radio button is selected. The 'Output' section at the bottom is highlighted with an orange border around the 'All' radio button. At the bottom are 'Cancel' and 'Submit' buttons.

4. Save the *Product Offers* strategy.
5. Close the *Product Offers* strategy.
6. On the change request, click **Submit** to proceed.
7. Enter a comment and click **Submit** to complete the change request.
8. Close the confirmation window.
9. Refresh the revision window to confirm that the change request was completed.
10. On the revision, click **Submit**.
11. Enter a comment and click **Submit** to complete the revision.
12. Close the confirmation window to return to the dashboard.

Verify your work

1. Create a Top Offers case.

Tip: Select **Run > TopOffers** to create a top offers case.

2. Select **Louise Simpson** for **Customer** and **Product offer** for the **Call Reason**.
3. Click **Submit** to proceed.

4. Verify that four offers are displayed.

Offered proposition		
Name	Acceptance probability	Outcome
Apple iPad	<input checked="" type="radio"/> Accept	<input type="radio"/> Reject
Samsung Galaxy S	<input checked="" type="radio"/> Accept	<input type="radio"/> Reject
Samsung Galaxy Tab	<input checked="" type="radio"/> Accept	<input type="radio"/> Reject
Apple iPhone	<input checked="" type="radio"/> Accept	<input type="radio"/> Reject

3. Select an offer and click **Submit**.
4. Examine the interaction history for this customer.
5. Click **Submit** to complete the case.
6. Click the cross button to close the case.

Enriching business applications with Next-Best-Action

Exercise: Changing the strategy for an interaction

Scenario

Management wants to remove the sales propositions and instead offer retention propositions when the call reason is both customer care and product offer.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
System Architect	SystemArchitect@WhatIsInTheBox	rules

Your assignment

Modify the Top Offers application to offer retention propositions instead of sales offers. First, change the *NextBestAction* interaction rule, and point it to a copy of the *NextBestAction* strategy called *NextBestRetention*. Then, modify the *NextBestRetention* strategy to only offer retention propositions.

Detailed steps

Step through the Top Offers case

1. Create a Top Offers case.
Tip: Select **Run > TopOffers** to create a top offers case.
2. Select **Louise Simpson** for **Customer** and **Product offer** for the **Call Reason**.

The screenshot shows the 'Assignment' tab of the Top Offers application. At the top, there are tabs for 'Assignment', 'Information', and 'Audit'. Below the tabs, the title 'Top Offers' is displayed next to a search bar containing the text 'StrategyDesigner@WhatIsInTheBox'. Under the 'Assignment' tab, there are two dropdown fields: 'Customer *' which is set to 'Louise Simpson', and 'Call reason *' which is set to 'Product offer'. Both of these fields have orange outlines around them, indicating they are selected or highlighted.

- Click **Submit** to proceed.
- Accept the Samsung Galaxy Tab offer, and reject the rest.

Offered proposition

Name	Acceptance probability	Outcome
Apple iPad		<input type="radio"/> Accept <input checked="" type="radio"/> Reject
Samsung Galaxy S		<input type="radio"/> Accept <input checked="" type="radio"/> Reject
Samsung Galaxy Tab		<input checked="" type="radio"/> Accept <input type="radio"/> Reject

- Click **Submit** to accept the offer.
- Examine the interaction history for this customer and click **Submit**.

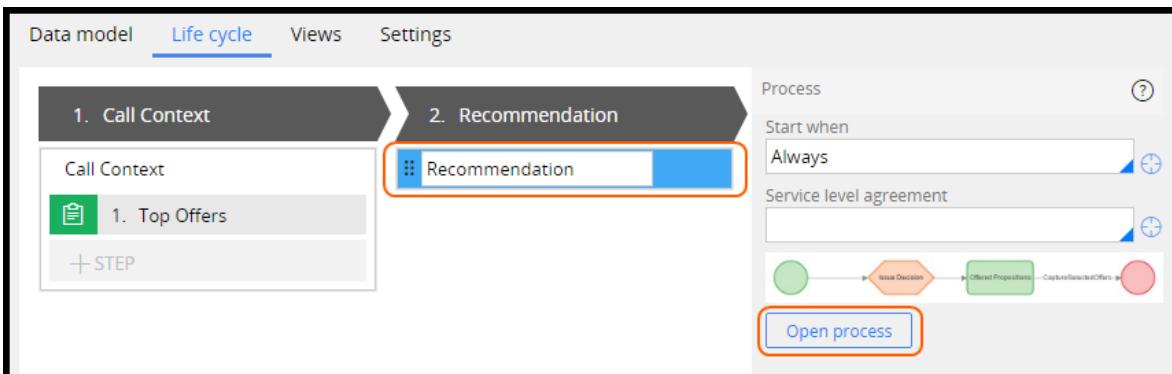
Interaction results for this customer

Fact ID	Subject ID	Issue	Group	Proposition Name	Offered on▼	Operator	Outcome
3525610817391545456	CE-1	Sales	Tablets	Samsung Galaxy Tab	11/8/16 11:57 PM	StrategyDesigner@WhatIsInTheBox	Accept
3525610817391545455	CE-1	Sales	Phones	Samsung Galaxy S	11/8/16 11:57 PM	StrategyDesigner@WhatIsInTheBox	Reject
3525610817391545454	CE-1	Sales	Tablets	Apple iPad	11/8/16 11:57 PM	StrategyDesigner@WhatIsInTheBox	Reject

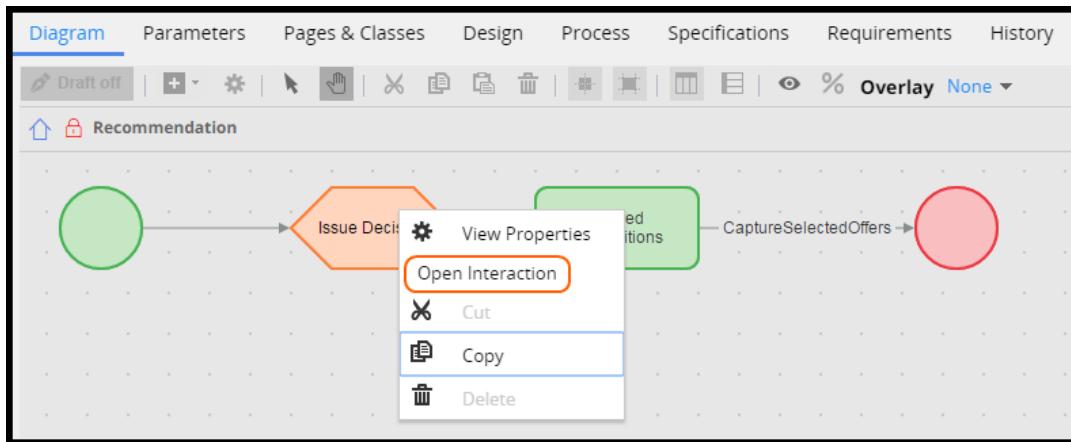
- Click the **Cross** button to close the case.

Create a new strategy

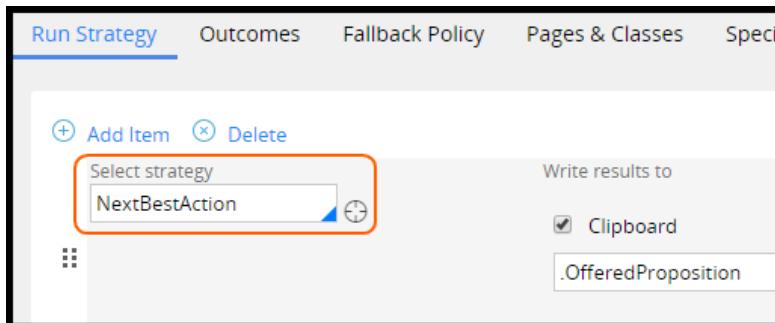
- Open the Case Explorer and select the **TopOffers** case.
- Select the **Recommendation** process.
- Click **Open process**.



- Right-click the **Issue Decision** flow shape and select **Open Interaction**.



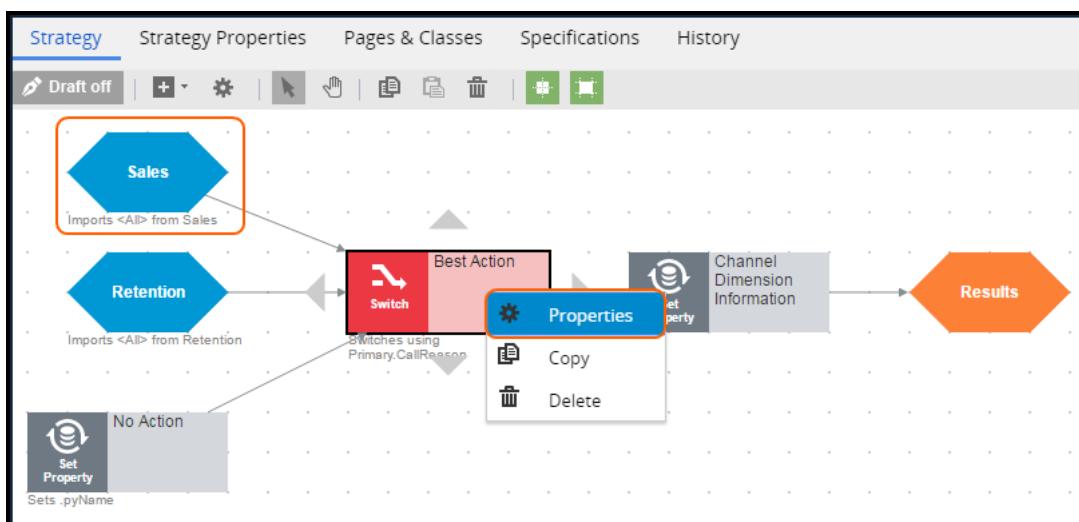
- Click the **Crosshair** icon to open the *NextBestAction* strategy.



- Click **Save as** to save the *NextBestAction* strategy as *NextBestRetention* strategy into the *WhatIsInTheBox:01-01-01* ruleset.

Configure the strategy

- Delete the **Sales** decision shape from the *NextBestAction* strategy.
- Right-click the **Best Action** switch shape and select **Properties**.



- Change the **Condition** when selecting **Retention** to **True**, delete any other conditions.

Switch Properties

Name *	Best Action
Component ID	BestAction
Description	<input checked="" type="radio"/> Use generated <input type="radio"/> Use custom
Switch	
⊕ Add Item ⊖ Delete	
Component	Condition
Select	Retention if True
Otherwise	No Action

- Save and check in the *NextBestAction* strategy rule.

Update the interaction

- Go back to the *NextBestAction* interaction rule.
- Click **Save as** to save the *NextBestAction* interaction into the *WhatIsInTheBox:01-01-01* ruleset.
- Change the strategy to *NextBestRetention*, then save the interaction.

Verify your work

- Create a Top Offers case.

Tip: Select **Run > TopOffers** to create a top offers case.

- Select **Louise Simpson** for **Customer** and **Product offer** for the **Call Reason**.

Assignment	Information	Audit
Top Offers	StrategyDesigner@WhatIsInTheBox	
Customer *	Louise Simpson	
Call reason *	Product offer	

- Click **Submit** to proceed.

4. Verify that the voice calls and text messages retention propositions are displayed.

Offered proposition		
Name	Cost	Outcome
Voice Calls	\$100.00	<input checked="" type="radio"/> Accept <input type="radio"/> Reject
Text Messages	\$50.00	<input checked="" type="radio"/> Accept <input type="radio"/> Reject

The importance of propositions

Exercise: Measuring a reject rate for a proposition

Scenario

Management wants to add a new proposition — International calls — for customer care. They also want to add a new key performance indicator for measuring the reject rate for a proposition.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Strategy Designer	StrategyDesigner@PropositionManagement	rules

Your assignment

As Strategy Designer, create a new change request and add a new proposition International Calls to the Discount group. Set the Cost proposition property to 70. Add a new key performance indicator for the rejection rate (reject rate = total rejected / volume). Use the Visual Business Director to examine the key performance indicators.

Detailed steps

Gather interaction history

1. Select **Run > TopOffers** to create a top offers case.
2. Under **Customer**, select **Mary Turner**, and under **Call reason**, select **Customer Care**.

The screenshot shows a software interface for creating a 'Top Offers' case. At the top, there are three tabs: 'Assignment', 'Information', and 'Audit'. The 'Assignment' tab is currently active. Below the tabs, the title 'Top Offers' is displayed. Under the 'Assignment' section, there are two dropdown menus. The first dropdown is labeled 'Customer *' and has 'Mary Turner' selected. The second dropdown is labeled 'Call reason *' and has 'Customer care' selected. Both of these dropdowns are outlined with a thick orange border.

3. Click **Submit** to proceed.

4. Accept the **Voice Calls** offer and reject the **Text Messages** offer.

Offered proposition		
Name	Cost	Outcome
Voice Calls	\$100.00	<input checked="" type="radio"/> Accept <input type="radio"/> Reject
Text Messages	\$50.00	<input type="radio"/> Accept <input checked="" type="radio"/> Reject

5. Click **Submit** to accept the offer.
6. Examine the interaction history for this customer and click **Submit**.
7. Repeat steps 1-5, but this time accept the **Text Messages** offer and reject the other offer.

Examine interaction history reports

1. Click **Browse Reports**.

The screenshot shows the Strategy Designer interface. On the left, a sidebar lists navigation options: Dashboard, Manage, Revision Management, Monitor Adaptive Models, Monitor Interaction History, Run Simulations, **Browse Reports** (which is highlighted with an orange border), and Plan. The main area is titled 'Dashboard' and displays a message: 'Welcome back, Strategy Designer. There are no open revisions.' Below this is a button labeled 'New revision ...'. Further down, there's a section titled 'My DM worklist' with columns for Urgency, Name, ID, and Due. A message at the bottom states 'No work assigned'.

2. In the Public categories section, click the **Interaction History** to list the interaction history reports.
3. Click **Acceptance Rate** to open the chart and review the values.
4. Click **Channel Distribution** to open the chart and review the values.
5. Click **Interaction History** to open the chart and review the values.

6. Click **Proposition Distribution** to open the chart and review the values.

Showing reports in category: Interaction History

Recent reports All reports Search reports

Title	Email Notification	
Acceptance Rate		
Channel Distribution		
Interaction History		
Proposition Distribution		

Private categories
No items

Public categories

Adaptive Decisioning	5
Interaction History	4
Service Level Performance	2

Launch Visual Business Director and navigate around the chart

1. On the Dashboard, click the **Manage** link.
2. Open the **Data sources** tab.
3. Click the **Actuals** link to launch the Visual Business Director.

Adaptive models Key performance indicators Data sources

Data sources contain records representing the contents of the Interaction History (Actuals), or records generated by running an interaction rule or data flow that generates simulation data. Use the links in the Name column to launch the Visual Business Director (VBD) planner for a given data source. Configure the start date of monitoring data by expanding the data source entry.

Delete Clear

Name	Start date	End date	# of records
Actuals	Nov 8, 2016	Nov 13, 2016	9

4. Drill down the proposition dimension to display proposition names.
5. Change the individual key performance indicators and observe the changes to the histogram.
6. Change the **Channel** dimension to display the **Operator** dimension instead.
7. Close Visual Business Director.

Create a change request for the International Calls proposition

1. Click **New revision** on the Dashboard to create a new revision.

Dashboard

Welcome back, Strategy Designer
There are no open revisions.

New revision ...

2. Click **New revision** to create a new revision.
3. Under **Name**, enter **Additional Retention Offer**.
4. Under **Objectives**, enter **Improve Retention offer portfolio**.

Define the Objectives for this Revision

This revision is assigned to you.

Create new revision
 Base this revision on a previous one

Name *
Additional Retention Offer

Objectives *
Improve Retention offer portfolio

+Add item

5. Click **Continue** to create a change request.
6. Click **Edit** to update the Change request.
7. Change the **Name** to **International Calls** and the **Description** to **Add a new proposition: International Calls**.

Change request Edit

Name
International Calls

Assigned to
Me

Objective
Improve Retention offer portfolio

Description
Add a new proposition: International Calls

Urgency
50

Due by
11/21/2016

8. Select the *Discounts* business rule.

Include Rules

Include Rules > Review & Refine

1 Rules Included

Search [+] Expand/collapse all

Discounts

Name	Type	Updated	Updated by
Discounts	Decision Data	1 year ago	DSME Administrator

9. Click **Next** to proceed.
10. Click **Submit** to complete the change request.

Add a new proposition

1. Click the **International Calls** link to open the newly created change request.
2. Click the **Discounts** link to open the Decision Data rule.
3. Click **New...** to create a proposition.

Data History

BUSINESS ISSUE GROUP
Retention Discounts

New ... Import ... Export Delete

Name	Description	Active
No decision data available		

4. Under **Name**, enter **International Calls**.
5. Under **Description**, enter **Offer 25% off international calls**.

- Under Cost, enter **70**.

Create proposition

Business Issue Group
Retention Discounts

Name *
International Calls

Description
Offer 25% off international calls

Active
 Always
 Never
 Within a defined time period

Cost
70

QTY

- Click **Create** to complete the proposition.
- Click **Save** on the Decision Data rule to add the proposition.

Add a new reject rate key performance indicator

- Select the **Manage** link.
- Select **Key performance indicators** tab.
- Click **Volume** to examine the volume key performance indicators.

Dashboard

Manage

Adaptive models **Key performance indicators** **Data sources**

Create and maintain Key performance indicators (KPIs) based on outcomes of your adaptive models. These KPIs can be used to track, compare and monitor business performance over time.

New ...

Name	Description	Updated
Volume	Count of Accept, Reject	1 year ago
Total rejected	Count of Reject	1 year ago
Total accepted	Count of Accept	1 year ago
Accept rate	Ratio of Total accepted over Volume	1 year ago

- Click **New...** to create a new key performance indicator.
- Under **Name**, enter **Reject Rate**.
- From the drop-down, select **is ratio of**.
- Select **Total rejected** over **Volume**.

8. Click **Submit** to save the new key performance indicator.

The screenshot shows the 'Create KPI' dialog box. The 'Name' field contains 'Reject Rate'. The 'is ratio of' dropdown is selected. Below it, 'Total rejected' is compared to 'Count of Reject' over 'Volume' compared to 'Count of Accept, Reject'. The 'Description' section shows 'Use generated' is selected. Comparison charts show green bars for higher values being better. Buttons at the bottom are 'Cancel' and 'Submit'.

9. Click **Save** to save the change request.
10. Click **Submit** to close the change request.
11. Enter a **Comment**, then click **Submit**.
12. Click **Close** to close the change request.
13. Click **Refresh** to refresh the revision.
14. Click **Submit** to close the revision.
15. Enter a **Comment**, then click **Submit**.

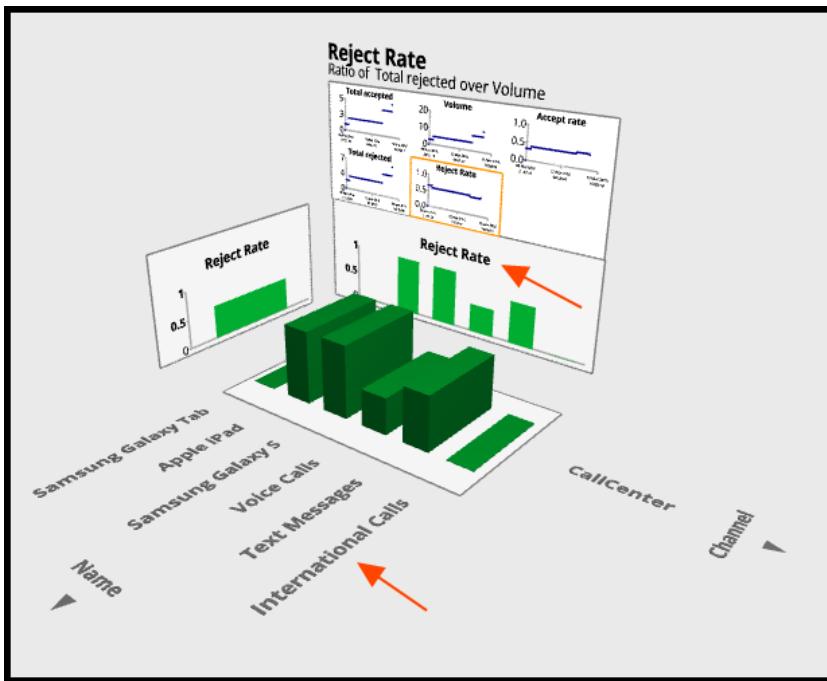
Verify your work

1. Create a new Top Offers case for **Customer Mary Turner** and **Call reason Customer Care**, then click **Submit**.

- Accept the **International Calls** offer and reject the other offers.

Offered proposition			
Name	Cost	Outcome	
International Calls	\$70.00	<input checked="" type="radio"/> Accept	<input type="radio"/> Reject
Voice Calls	\$100.00	<input type="radio"/> Accept	<input checked="" type="radio"/> Reject
Text Messages	\$50.00	<input type="radio"/> Accept	<input checked="" type="radio"/> Reject

- Click **Submit** to accept the offer.
- Examine the interaction history for this customer and click **Submit**.
- On the dashboard, click the **Manage** link.
- Open the **Data sources** tab.
- Click the **Actuals** link to launch Visual Business Director.
- Expand the proposition dimension. Verify that the newly added International calls proposition and the newly added Reject Rate key performance indicator is displayed.



How to design a decision strategy

Exercise: Creating a next-best-action decision strategy

Scenario

Management wants a new next-best-offer decision strategy that returns the best two tablet offers based on the margin. In addition, the 4G tablet offer should only be made to frequent callers.

The following table provides the credentials you need to complete the exercise.

Role	Operator ID	Password
Strategy Designer	StrategyDesigner@StrategyBuild	rules

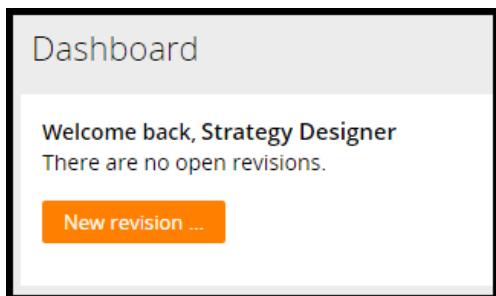
Your assignment

As Strategy Designer, create a new change request to revise the *Next Best Offer* strategy. In the *Next Best Offer* strategy, import the proposition data for each tablet offered. Only offer the 4G tablet to frequent callers (customers with the property *FrequentCaller* set to Yes). Extend the strategy further to prioritize offers by margin. Calculate the margin using the expression (Margin = (Price – Cost) / Points). Ensure that only the two top propositions are returned.

Detailed steps

Create a change request for the strategy

1. On the Dashboard, click **New revision** to create a new revision.



2. Under **Name**, enter Build Next-Best-Offer Strategy.

3. Under **Objectives**, enter Present top two Tablet offers.

Define the Objectives for this Revision

This revision is assigned to you.

Create new revision
 Base this revision on a previous one

Name *****
Build Next-Best-Offer Strategy

Objectives *****
Present top two Tablet offers

+Add item

4. Click **Continue** to create a change request.
5. Click **Edit** to update the Change request.
6. Change the **Name** to Next Best Offer and the **Description** to Build the Next Best Offer strategy, which selects all propositions from the Tablets group and prioritizes the offers based on Margin. Only offer the 4G tablet to customers who call frequently.

Change request Edit

Name
Next Best Offer

Assigned to
Me

Objective
Present top two Tablet offers

Description
Build the Next Best Offer strategy, which selects all propositions from the Tablets group and prioritizes the offers based on Margin. Only offer the 4G tablet to customers who call frequently.

Urgency Due by
50 11/21/2016

7. Select the *Next Best Offer* business rule.

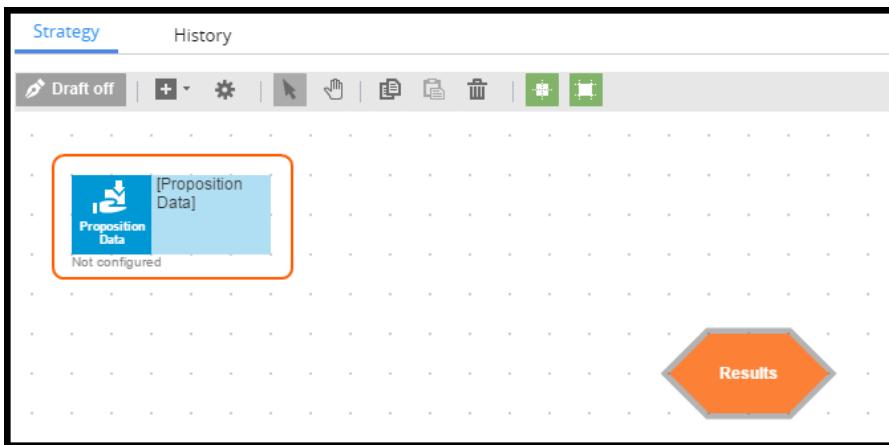
The screenshot shows a list of business rules under the 'DSM Essentials data customer' section. The 'Sales' section is expanded, showing two entries: 'Next Best Offer (Results in Sales)' and 'Next Best Offer (Defined on DSM Essentials data customer)'. Both entries have checkboxes next to them, which are checked. The entire list is enclosed in a black border, and the first two entries are highlighted with an orange rounded rectangle.

Name	Type
Next Best Offer (Results in Sales)	Strategy
Next Best Offer (Defined on DSM Essentials data customer)	Strategy

8. Click **Next** to proceed.
9. Click **Submit** to save the change request.

Configure the proposition data components

1. Open the newly created change request.
2. Click the **Next Best Offer** link to open the Next Best Offer strategy.
3. Right-click the canvas and select **Import > Proposition Data** to add a proposition data component.



4. Right-click the new **Proposition Data** shape and select **Properties**.
5. In the **Group** field, in the **Import from** section, select **Tablets**.

- Select the **Apple iPad** proposition.

The screenshot shows the 'Proposition Data Properties' dialog. At the top, there are fields for 'Name*' (Apple iPad), 'Component ID' (AppleiPad), and 'Description' (radio buttons for 'Use generated' and 'Use custom'). Below this is a tabbed section with 'Proposition data' selected, showing an 'Import from' panel. This panel contains dropdown menus for 'Business issue*' (Sales), 'Group' (Tablets), and 'Proposition' (Apple iPad). The entire 'Import from' panel is highlighted with an orange border.

- Repeat steps 3-6, but this time select **Generic 4G Tablet** proposition.
- Repeat steps 3-6, but this time select **Samsung Galaxy Tab** proposition.

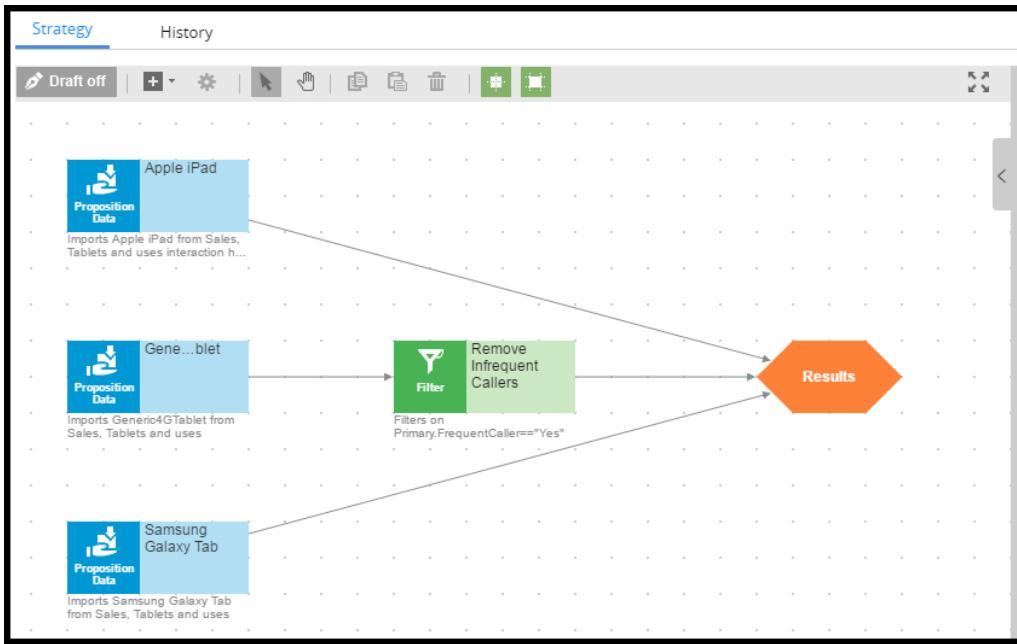
Configure the filter component

- Right-click the canvas and select **Arbitration > Filter** to create a filter component.
- Right-click the new filter component and select **Properties**.
- Under **Name**, enter **Remove Infrequent Callers**.
- Add a filter condition **Primary.FrequentCaller == "Yes"** using the expression editor.

The screenshot shows the 'Filter Properties' dialog. It includes fields for 'Name*' (Remove Infrequent Callers), 'Component ID' (RemoveInfrequentCallers), and 'Description' (radio buttons for 'Use generated' and 'Use custom'). Below these is a 'Source components' section with a 'Filter' tab selected. Under 'Type', the 'Filter condition' radio button is selected. A 'Filter condition' input field contains the expression 'Primary.FrequentCaller=="Yes"', which is also highlighted with an orange border. There is a gear icon to the right of the input field.

- Click **Submit** to save the filter properties.
- Connect the 4G Tablet **Proposition Data** component to the **Filter** component.

7. Connect the **Filter** and the remaining two **Proposition Data** components (Apple iPad and Samsung Galaxy Tab) to the **Results** component.



8. Click **Save** to complete the configuration and add the three offers to the strategy.

Test the Next Best Offer Strategy

1. Open the Test Panel.

Note: Click the toggle on the right side of the screen.

2. Select **Data Transform** and select **FrequentCaller**.

3. Click **Save & run**. You should see three propositions offered.

The screenshot shows the 'Single case' tab selected in the Test Panel. Under 'Single case settings', 'Data transform' is set to 'FrequentCaller'. The 'Results for strategy' section displays three rows of data:

Field	Result
Business issue	Sales
Component	Remove Infrequent Callers
Cost	20
Group	Tablets

Page navigation controls show 'Page 1 of 3'.

4. Change the data transform to **InfrequentCaller**.
5. Click **Save & run**. You should see two propositions offered.

The screenshot shows the 'Single case' tab selected in the Test Panel. Under 'Single case settings', 'Data transform' is set to 'InfrequentCaller'. The 'Results for strategy' section displays two rows of data:

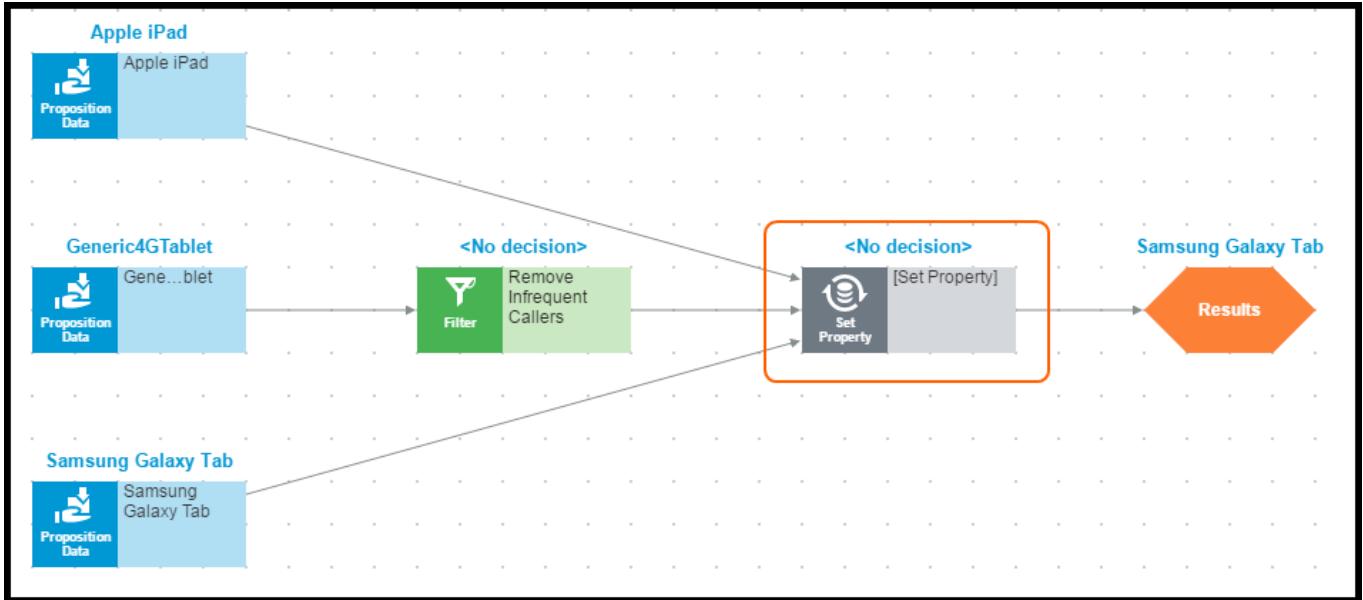
Field	Result
Business issue	Sales
Component	Samsung Galaxy Tab
Cost	---
Group	Tablets

Page navigation controls show 'Page 1 of 2'.

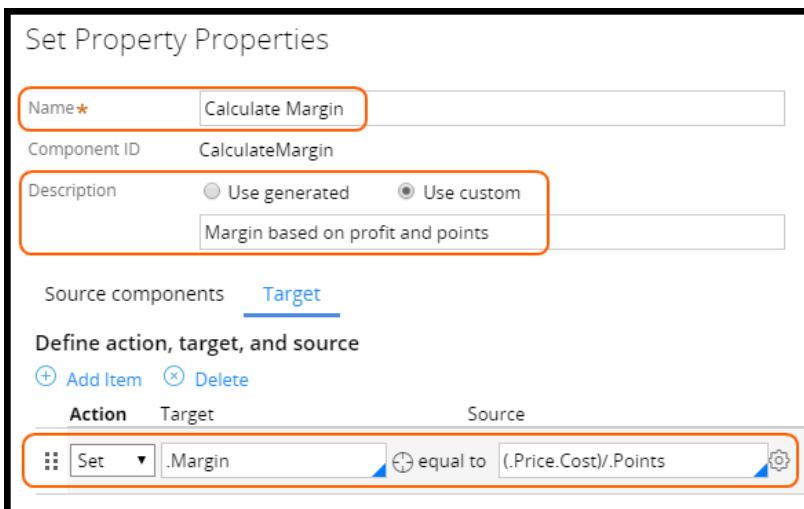
6. Close the Test Panel.

Calculate the margin

1. Right-click the canvas and select **Enrichment > Set Property** to add the Set Property components.
2. Connect the **Filter** component and two proposition data components to the **Set Property** component.



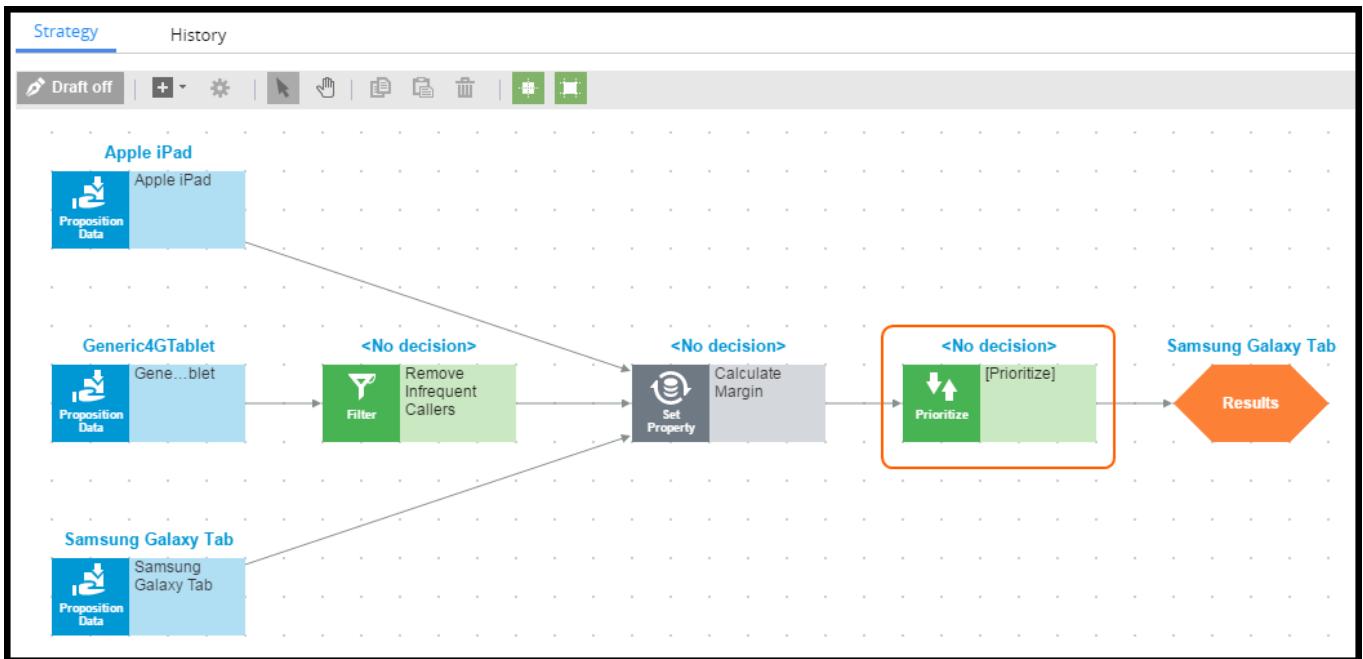
3. Right-click the Set Property component and select **Properties** to open the property settings.
4. Under **Name**, enter **Calculate Margin**.
5. Select **Use Custom**, and under **Description**, enter **Margin based on profit and points**.
6. Click **Add item**, set the **Target** to **.Margin**, and use the expression editor to set the **Source** to **(.Price - .Cost) / .Points**.



7. Click **Submit** save the component.
8. Click **Save** to complete the margin calculation.

Prioritize the propositions based on margin

1. Right-click the canvas and select **Arbitration > Prioritize** to add the prioritize component.
2. Connect the component to the **Set Property** component and **Results**.



3. Right-click the **Prioritize** component and select **Properties** to open the property settings.
4. Under **Name**, enter **Prioritize By Margin**.
5. Use **.Margin** in the prioritization formula.
6. Select **Order by Highest first**.

7. Set the **Output** to **Top 2** and click **Submit**.

The screenshot shows the 'Prioritize Properties' dialog box. At the top, there is a 'Name*' field containing 'Prioritize By Margin' with a red border around it. Below it are 'Component ID' (PrioritizeByMargin) and 'Description' (with radio buttons for 'Use generated' and 'Use custom'). Under 'Source components', there is a 'Prioritize' button and two radio buttons: 'Prioritize values' (selected) and 'Sort alphabetical'. In the 'Expression*' field, '.Margin' is entered, with a gear icon to its right. The 'Order by' section has two radio buttons: 'Highest first (9 to 1)' (selected) and 'Lowest first (1 to 9)'. Finally, the 'Output' section at the bottom has two radio buttons: 'Top' (selected) and 'All', with a value '2' in a text input field next to it.

8. Click **Save** to complete your changes to the strategy.

Test the strategy

1. On the right side of the screen, click the toggle to open the Test Panel.
2. Select **Data Transform** and select **FrequentCaller**.

3. Click **Save & run**. You should see two propositions being offered.

The screenshot shows the 'Single case' tab selected in the top navigation bar. Under 'Single case settings', the 'Data transform' section is expanded, showing 'FrequentCaller' selected. Below this, there are 'Save & run' and 'Clear' buttons. A dropdown menu labeled 'Show' is set to 'Name'. The 'Results for strategy' section displays a table with two pages of results. The first page is shown, with the page number '1' highlighted by a red box. The table has columns 'Field' and 'Result'. The data rows are:

Field	Result
Business issue	Sales
Component	Prioritize By Margin
Cost	20
Group	Tablets

4. Change the data transform to **InfrequentCaller**.
5. Click **Save & run**. You should see two propositions being offered.

The screenshot shows the 'Single case' tab selected in the top navigation bar. Under 'Single case settings', the 'Data transform' section is expanded, showing 'InfrequentCaller' selected. Below this, there are 'Save & run' and 'Clear' buttons. A dropdown menu labeled 'Show' is set to 'Name'. The 'Results for strategy' section displays a table with two pages of results. The first page is shown, with the page number '1' highlighted by a red box. The table has columns 'Field' and 'Result'. The data rows are:

Field	Result
Business issue	Sales
Component	Prioritize By Margin
Cost	---
Group	Tablets

6. Close the Test Panel.

Verify your work

1. Create a Top Offers case.

Tip: Select **Run > TopOffers** to create a top offers case.

2. Select **Sarah Hart** for **Customer** and **Product offer** for the **Call Reason**.
3. Click **Submit** to proceed. Verify that the following two offers are displayed.

Offered proposition		
Name	Acceptance probability	Outcome
Generic4GTablet		<input checked="" type="radio"/> Accept <input type="radio"/> Reject
Samsung Galaxy Tab		<input checked="" type="radio"/> Accept <input type="radio"/> Reject

3. Repeat 1-4 for **CustomerMichael Anderson** and verify that the following two offers are displayed.

Offered proposition		
Name	Acceptance probability	Outcome
Samsung Galaxy Tab		<input checked="" type="radio"/> Accept <input type="radio"/> Reject
Apple iPad		<input checked="" type="radio"/> Accept <input type="radio"/> Reject

COURSE SUMMARY

Next steps for Senior System Architects

Senior System Architect summary

Now that you have completed this course, you should be able to:

- Identify the tasks and responsibilities of the senior system architect on a Pega Implementation
- Create and extend a Pega application
- Manage rules and rulesets
- Leverage the Enterprise Class Structure (ECS) to promote rule reuse between case types and applications
- Configure roles, access groups, and privileges
- Manage data access within an application
- Create wizards to configure a sequence of assignments
- Design applications for multiple device types and screen sizes, including mobile devices
- Manage integration settings
- Incorporate next-best-action decision-making into applications
- Test your application design to analyze rule behavior and identify configuration errors

Next steps

Completion of Senior System Architect 7.2 helps to prepare students to take the Certified Senior System Architect exam. To practice taking the exam, enroll in the CSSA Practice Exam course in Pega Academy.

Certification

Become a Certified Senior System Architect



Certification Benefits

Industry Credibility

The Senior System Architect course and practice exam prepare you to take the Certified Senior System Architect exam

Pega Discovery Network

Join the Pega Discovery Network (PDN) today. The PDN is an online community for Pega developers, customers, and partners to find everything they need to successfully implement Pega solutions.



Discover information on a variety of development topics.



Share ideas, ask questions, and participate in a variety of discussions with other members of the Pega community.



Learn at your own pace with our Pega Academy self-study courses or with a certified Pega trainer.

Have a product question? Want to engage with other Pega users, employees, and partners? Check out our Pega Developer Network to converse with the PDN community at pdn.pega.com

For self-study training on Pega applications, go to [Pega Academy](#).