System Architect Essentials II

EXERCISE GUIDE (v. 7.1)

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

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

Although Pegasystems Inc. strives for accuracy in its publications, any publication may contain inaccuracies or typographical errors. This document or Help System could contain technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Pegasystems Inc. may make improvements and/or changes in the information described herein at any time.

This document is the property of:
Pegasystems Inc.
1 Rogers Street
Cambridge, MA 02142
Phone: (617) 374-9600
Fax: (617) 374-9620
www.pega.com

Document Name: sae2_71ml6_ExerciseGuide_20141114

PVS: sae2_71ml6_pvs_20141112.exe

Date: 20141114

Table of Contents

Module 02: Effective Application Development with PRPC

This lesson group includes the following lessons:

- HR Services Application Operational Walkthrough (no exercise)
- The Building Blocks of a PRPC Application (no exercise)
- Managing the Building Blocks of a PRPC Application (no exercise)
- Guided Application Development Using Guardrails

Exercise: Understanding Guardrails

Scenario

When developing applications with Pega 7 it is very important that you continually monitor your application's guardrail warnings. By continually monitoring the guardrail warnings you are able to solve potential problems ahead of time that can save you development time and maintenance on future releases. For this exercise, the goal is to get you familiar with how to find out detailed information about the guardrail warnings in your application.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Observe the existing guardrail warnings in the HR Services Application.

Use the guardrail summary to determine the answers to the following questions:

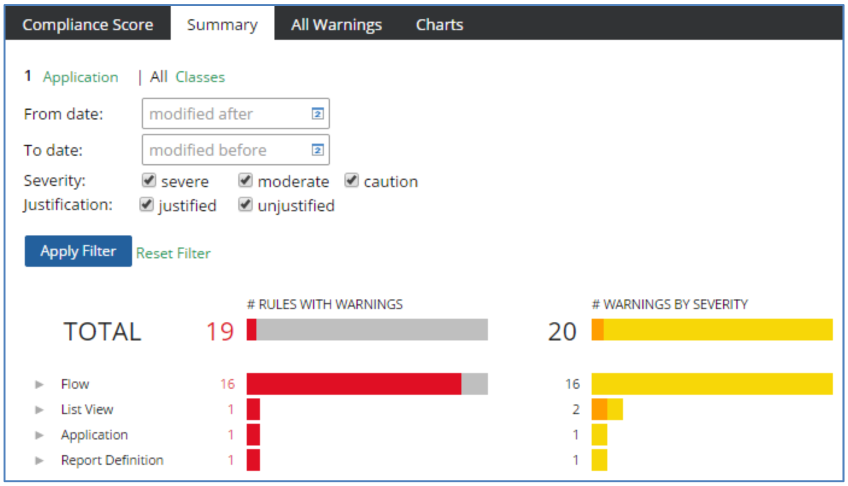
- How many Flow guardrail warnings are there?
- How many List View caution level guardrail warnings are there?
- Which rule type has the most Severe warnings?

Expand the warnings for the Flow category, select a rule and determine what the guardrail warning is.

Procedure

Follow the steps below:

1. Log in as the System Architect using the credentials provided in the scenario.
2. Select **Application > Guardrails > All Warnings** from the Designer Studio menu to go to the guardrail warning page.
3. Use the summary chart to determine the answers to the questions (your numbers may vary than what you see below).



1 Application | All Classes

From date: modified after

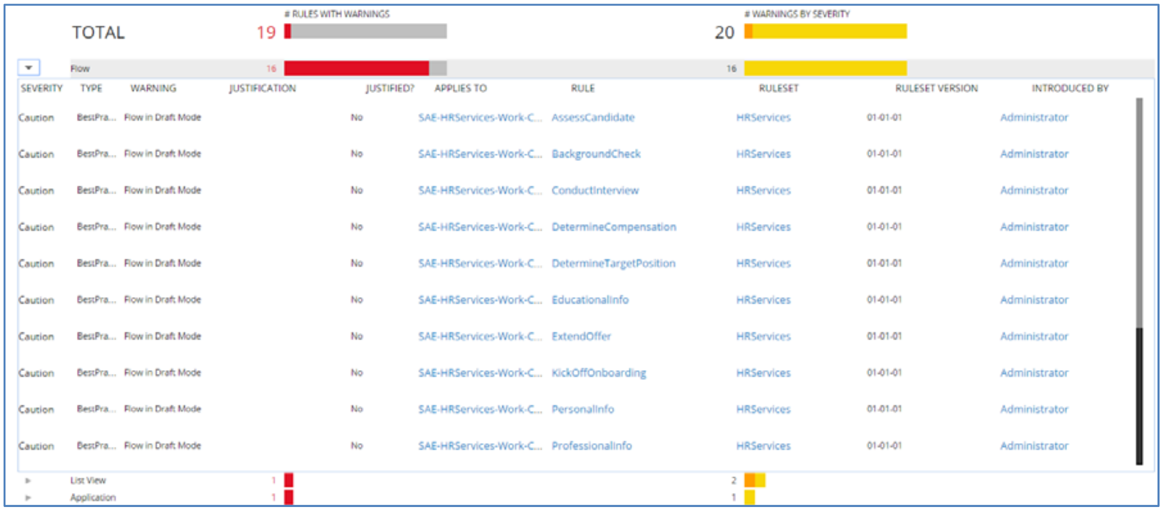
To date: modified before

Severity: ☑ severe ☑ moderate ☑ caution

Justification: ☑ justified ☑ unjustified

		# RULES WITH WARNINGS	# WARNINGS BY SEVERITY
TOTAL		19	20
▶	Flow	16	16
▶	List View	1	2
▶	Application	1	1
▶	Report Definition	1	1

4. Expand the Flow category and examine the violations, then select one of the rules to examine.



		# RULES WITH WARNINGS	# WARNINGS BY SEVERITY
TOTAL		19	20
▼	Flow	16	16

SEVERITY	TYPE	WARNING	JUSTIFICATION	JUSTIFIED?	APPLIES TO	RULE	RULESET	RULESET VERSION	INTRODUCED BY
Caution	BestPra...	Flow in Draft Mode		No	SAE-HRServices-Work-C...	AssessCandidate	HRServices	01-01-01	Administrator
Caution	BestPra...	Flow in Draft Mode		No	SAE-HRServices-Work-C...	BackgroundCheck	HRServices	01-01-01	Administrator
Caution	BestPra...	Flow in Draft Mode		No	SAE-HRServices-Work-C...	ConductInterview	HRServices	01-01-01	Administrator
Caution	BestPra...	Flow in Draft Mode		No	SAE-HRServices-Work-C...	DetermineCompensation	HRServices	01-01-01	Administrator
Caution	BestPra...	Flow in Draft Mode		No	SAE-HRServices-Work-C...	DetermineTargetPosition	HRServices	01-01-01	Administrator
Caution	BestPra...	Flow in Draft Mode		No	SAE-HRServices-Work-C...	EducationalInfo	HRServices	01-01-01	Administrator
Caution	BestPra...	Flow in Draft Mode		No	SAE-HRServices-Work-C...	ExtendOffer	HRServices	01-01-01	Administrator
Caution	BestPra...	Flow in Draft Mode		No	SAE-HRServices-Work-C...	KickOffOnboarding	HRServices	01-01-01	Administrator
Caution	BestPra...	Flow in Draft Mode		No	SAE-HRServices-Work-C...	PersonalInfo	HRServices	01-01-01	Administrator
Caution	BestPra...	Flow in Draft Mode		No	SAE-HRServices-Work-C...	ProfessionalInfo	HRServices	01-01-01	Administrator

▶	List View	1	2
▶	Application	1	1

5. View the warning for the flow rule.

Reference Information

- PRPC Help: Guardrails landing page

Module 03: Designing Enterprise Applications Using Case Management

This lesson group includes the following lessons:

- Best Practices for Case Management Design (no exercise)
- Managing Enterprise Apps Using Stage-Based Case Design
- Best Practices for Effective Case Decomposition
- Best Practices for Effective Process Decomposition
- Guardrails for Case Management Design

Exercise: Define the Behavior of a Business Transaction Using Case Types and Stages

Scenario

You are part of a project team tasked with providing a solution for automating the onboarding of new employees. The onboarding process begins with the acceptance of an offer and runs throughout the first 90 days of employment. In this first release, the plan is to manually create the new employee record prior to their start date. Once the new employee record is created, a welcome kit should be sent within 48 hours of the new employee accepting the offer.

Prior to the new employee's start date, the hiring manager needs to request assets for the new employee and begin setting up the new employee's enablement plan. HR should begin setting up the orientation schedule as well. When the new employee starts, they begin orientation. During orientation, they should enroll in the benefits program and setup their payroll account.

All new employees go through a "30-60-90" evaluation program. This program has an evaluation period set at 30, 60 and 90-day intervals to review their performance measurements with their mentor and manager.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

The first goal when defining a case type is to summarize the business transaction by defining the stages for the case type.

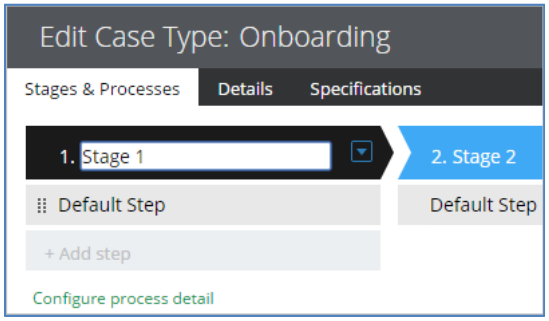
Add a case type to the HR Services application to support the onboarding process. Focus on the top-level case – that of onboarding a new employee - and identify the stages for the top-level case.

Remember, stages are a first level of organizing the different tasks required to complete work associated with a case.

Hints

1. Add an **Onboarding** case type using the Cases Explorer menu.

2. Edit the existing stages by clicking on the stage.

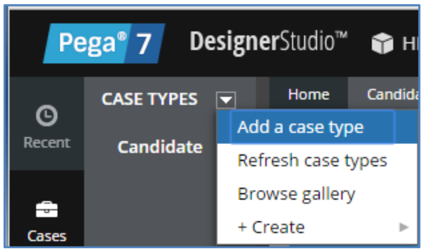


3. Limit stage names to no more than 2 words and use a noun or noun phrase.

Procedure

Follow the steps below:

1. From the Case Designer, add a new case type in the HR Services application. Name the case type **Onboarding**.



2. Rename the three default stages with an appropriate name and add additional stages if necessary.

   Tip: A good set of stage names to being with would be: **Pre-Arrival**, **Orientation**, and **30-60-90 Evaluation**.

Reference Information

- PDN: Configure Stages for Your Application (Node ID: 46526).

Exercise: Decompose a Case into Steps

Scenario

The onboarding process begins with the acceptance of an offer and runs throughout the first 90 days of employment. In this first release, the plan is to manually create the new employee record prior to their start date. Once the new employee record is created, a welcome kit should be sent within 48 hours of the new employee accepting the offer.

Prior to the new employee's start date, the hiring manager needs to request assets for the new employee and begin setting up the new employee's enablement plan. HR should begin setting up the orientation schedule as well.

When the new employee starts, they begin orientation. During orientation, they should enroll in the benefits and setup their payroll account.

All new employees go through a "30-60-90" evaluation program. This program is an evaluation period set at 30, 60, and 90-day intervals to review their performance measurements with their mentor and manager.

The goal for this iteration is to identify the key steps. Do not worry about specifying whether a step should be a single-step assignment, a multi-step process, or a subcase. If you identify a number of related steps in the scenario that you believe should be in a single process, it's OK to add a step in the appropriate stage and configure it as a "Multi-Step Process."

If you believe there are parts of the scenario that would best be served as a subcase, you still need to add the step to the appropriate stage. However, do not configure it as a subcase yet. We will do that in a later exercise.

Focus your attention on identifying the steps and leave the step configuration for the next iteration.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Our summarization of the Onboarding case type revealed three to four stages.

- We identified a setup or "Intake" stage and a "Pre-Arrival" stage to define what happens before the new employee starts. However, these two stages could be combined into a single stage.
- We identified a stage — which we named "Orientation" — where we define what takes place during the new employee's orientation.
- Finally, we identified a stage — which we named "30-60-90 Program" — where we define what takes place during the new employee's evaluation period.

Finally, although not specifically stated in the scenario, we decided to add an alternate stage named "Cancel Onboarding" to define what happens if the new employee does not complete the onboarding program; just in case.

Turning our attention to the steps in each stage, in our solution, we have implemented the following:

- Two steps in the setup stage where we will create the employee record and send the welcome kit;
- Four steps in the pre-arrival stage where we will set up the orientation schedule and the enablement plan, request assets for the new employee, and pause the case until the new employee's start date;
- Two steps in the orientation stage where the employee will enroll in the benefits program and set up their payroll account;
- And one step in the employee evaluation stage.

Remember, a step is a distinct action that produces a singular outcome. When naming steps, use a "verb plus noun" naming convention - perform "this action" on "this object."

Hints

1. We have determined that one approach to this case type would be to have four steps (Intake, Pre-Arrival, Orientation and 30-60-90 Evaluation).
2. We should have a Cancel Onboarding stage in case a new employee does not complete the onboarding process.
3. We have determined that we should create an employee record, send a welcome kit, setup an orientation schedule, request employee assets, setup and enablement plan, wait for the new employee to arrive, perform benefits enrollment, setup payroll and evaluate the employee. It's also possible that we will need to cancel onboarding.

Procedure

Follow the steps below:

1. Following the best practices and guidelines you studied in this lesson, add the appropriate steps to each stage.
2. Remember to account for the possible delay between when the pre-arrival work is completed and when the new employee actually starts their orientation.

Reference Information

- PDN: Design Your Application with Case Lifecycle Management

Exercise: Effective Process Decomposition

Scenario

The onboarding process begins with the acceptance of an offer and runs throughout the first 90 days of employment. In this first release, the plan is to manually create the new employee record prior to their start date.

Once the new employee record is created, a welcome kit should be sent within 48 hours of the new employee accepting the offer.

Prior to the new employee's start date, the hiring manager needs to request assets for the new employee and begin setting up the new employee's enablement plan. HR should begin setting up the orientation schedule as well.

When the new employee starts, they begin orientation. During orientation, they should enroll in the benefits programs and setup their payroll account. When an employee begins the benefits enrollment process, they can select healthcare coverage for medical, dental and vision options. They can also add dependents to their healthcare plans. When all coverage options have been made, the benefits selection is reviewed and finalized.

To setup payroll, the employee must provide information on how pay will be received. This can include providing account information for automatic deposit of their pay into their bank account.

All new employees go through a "30-60-90" evaluation program. This program is an evaluation period set at 30, 60, and 90-day intervals to review performance measurements with their mentor and manager.

In this third iteration, we will configure each of the Onboarding steps as a *Single Step Assignment*, *Multi Step Process*, or *Case*.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

In the second iteration of defining the onboarding case type, we identified the following steps:

- Create Employee Record
- Send Welcome Kit
- Setup Orientation Schedule
- Request Assets
- Setup Enablement Plan
- Wait for New Employee Start Date
- Benefits Enrollment

- Payroll Setup
- Evaluate Employee

Given the information provided in the scenario, in our next iteration we will configure each of the steps as single step assignments, multi-step process, or subcases.

We will leave all the steps in the first two stages as single step assignments, except the *Wait for New Employee Start Date*. At this step, we want to pend the case until the new employee's start date. To do this, we will need to use a *Wait* shape. Remember, a single step assignment can only ever contain a single assignment. Adding any other shape – even if it is only one other type of shape – will require us to use a *Multi Step Process*.

Tip: There should be two subcases in this scenario (enrolling for benefits and setting up payroll), which we need to define as case types in our application before we can add them to the Onboarding case map.

Note: Since the process for adding a subcase is the same regardless of the number added, we will detail only the requirements for enrolling for benefits. You can add requirements for payroll setup later.

Our approach to the benefits enrollment process is to create a case type that has the Onboarding case type as a parent. The benefits enrollment case type would have the following configuration:

Benefits Enrollment	
Stage	Steps
Personal Information	Confirm Personal Information
Benefit Election	Select Medical Coverage
	Select Dental Coverage
	Select Vision Coverage
Dependent Information	Add Dependents
Review and Finish	Confirm Benefit Elections
	Confirm Dependent Coverage

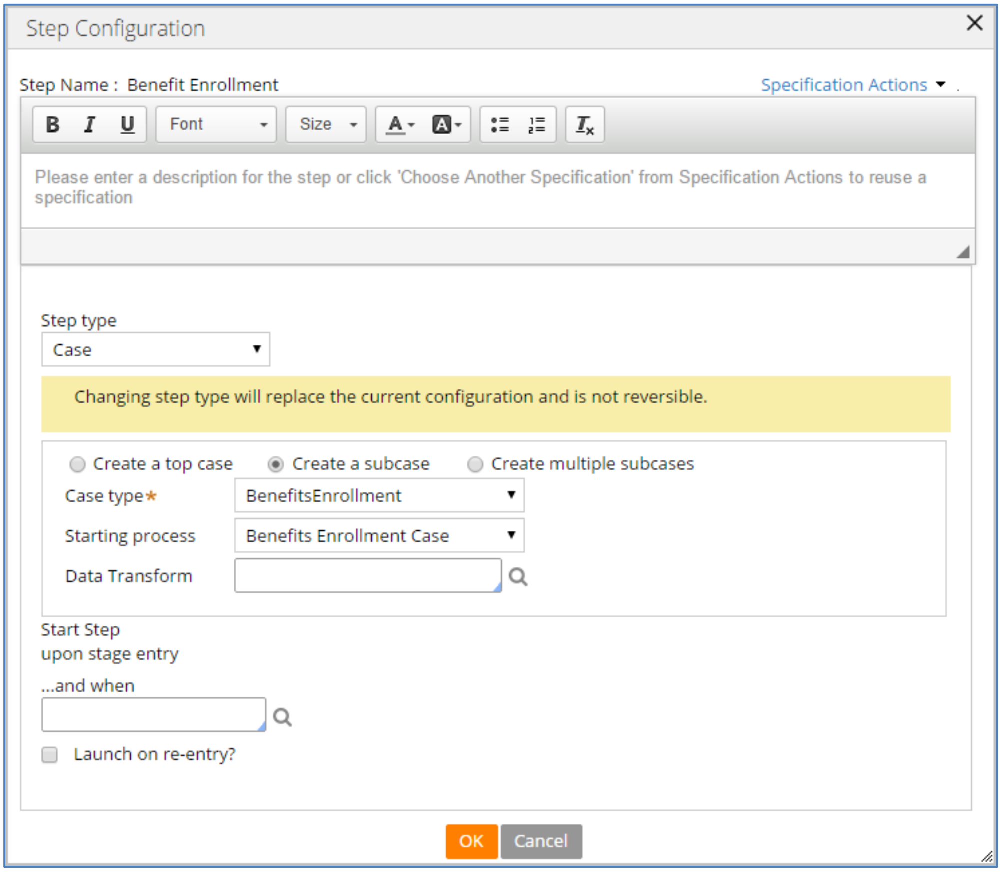
Hints

1. Most of the initial steps are single step assignments.
2. There could be a delay between when the pre-arrival work is completed and the new hire's start date. This step would need to be a multi-step process to accommodate the Wait shape used to pend the case.
3. The benefits enrollment work is best suited as a subcase.
4. Setting up payroll seems to be best suited as a subcase as well.
5. If all of the other steps in your case design are "Single-Step Assignments," that is OK. Case decomposition is an iterative effort. As we learn more of the details, some of the steps may change.

Procedure

Follow the steps below:

1. Configure each of the steps as either a *Single-Step Assignment* or *Multi-Step Process* according to the information provided in the scenario.
2. Add two new case types to the HR Services application (one for Benefits Enrollment and one for Setup Payroll) by selecting **Add a case type** from the Case Explorer menu. Set each to be a subcase and select the Onboarding case type as the parent.
3. Decompose the Benefits Enrollment case type to include the appropriate stages and steps to allow an employee to select healthcare benefit coverage. Use the table provided in the Approach for guidance. You do not need to fill in any stages or steps for the Setup Payroll case type.
4. Reconfigure the appropriate steps in the Onboarding case type as "Case" and use the "Create a subcase" option. Then, select the appropriate subcase from the case type list. Do this for both the Benefit Enrollment and Setup Payroll steps.



Step Name : Benefit Enrollment

Please enter a description for the step or click 'Choose Another Specification' from Specification Actions to reuse a specification

Step type: Case

Changing step type will replace the current configuration and is not reversible.

○ Create a top case ● Create a subcase ○ Create multiple subcases

Case type*: BenefitsEnrollment

Starting process: Benefits Enrollment Case

Data Transform:

Start Step
upon stage entry

...and when

□ Launch on re-entry?

Reference Information

- PDN: Design Your Application with Case Lifecycle Management (Node ID: 46551)

Exercise: Designing Enterprise Applications Using Case Management Exercise Verification

Scenario

Business people often discuss the way their case management application should function by describing the stages that a case goes through as it is worked to completion. In this series of lessons we learned how to design an application by describing these stages with a case type as the framework. We then defined the steps, or processes, that happen in each stage.

The example case management application we used was that of a new hire onboarding process.

The onboarding process begins with the acceptance of an offer and runs throughout the first 90 days of employment. In this first release, the plan is to manually create the new employee record prior to their start date. Once the new employee record is created, a welcome kit should be sent within 48 hours of the new employee accepting the offer.

Prior to the new employee's start date, the hiring manager needs to request assets for the new employee and begin setting up the new employee's enablement plan. HR should begin setting up the orientation schedule as well.

When the new employee starts, they begin orientation. During orientation, they should enroll for benefits and setup their payroll account.

All new employees go through a "30-60-90" evaluation program. This program is an evaluation period set at 30, 60 and 90-day intervals to review their performance measurements with their mentor and manager.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Onboarding Case

Our approach to designing the Onboarding case type was to add the following stages (in bold) and steps:

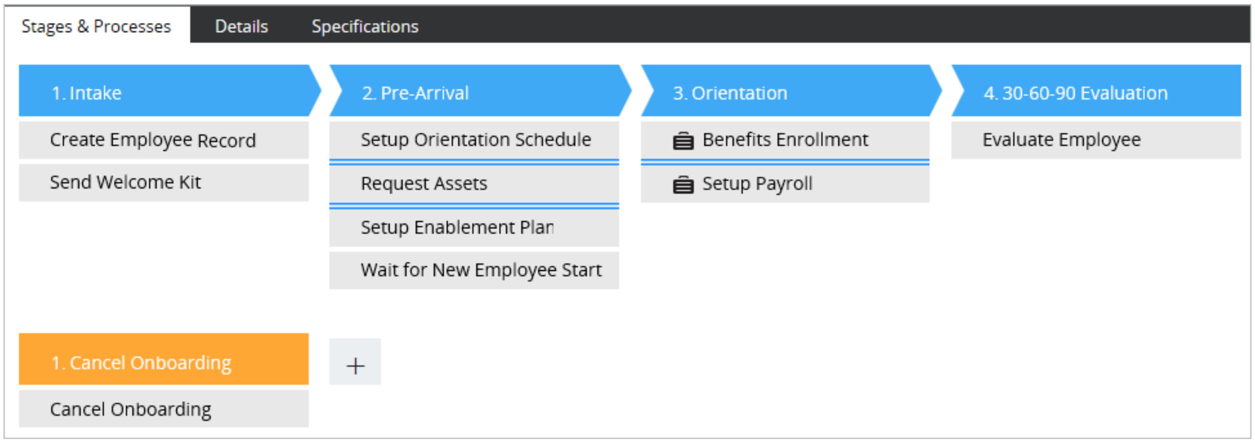
Intake
Create Employee Record (Single Step Assignment)
Send Welcome Kit (Single Step Assignment)

Pre-Arrival
Setup Orientation Schedule (Single Step Assignment)
Request Assets (Single Step Assignment – Start step upon stage entry)
Setup Enablement Plan (Single Step Assignment – Start step upon stage entry)
Wait for New Employee Start (Multi Step Process)

Orientation
Benefits Enrollment (Subcase)
Setup Payroll (Subcase – Start step upon stage entry)

30-60-90 Evaluation
Evaluate Employee (Single Step Assignment)

Cancel Onboarding (Alternate Stage)
Cancel Onboarding (Multi Step Process)



1. Intake	2. Pre-Arrival	3. Orientation	4. 30-60-90 Evaluation
Create Employee Record	Setup Orientation Schedule	Benefits Enrollment	Evaluate Employee
Send Welcome Kit	Request Assets	Setup Payroll	
	Setup Enablement Plan		
	Wait for New Employee Start		

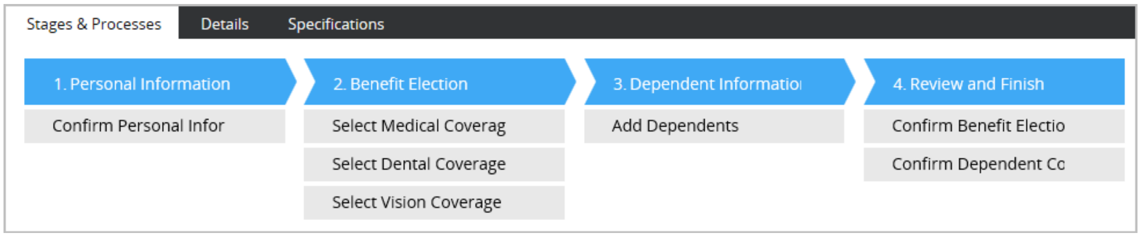
1. Cancel Onboarding
Cancel Onboarding

Benefits Enrollment Subcase

Our approach to designing the Benefits Enrollment case type was to add the following stages (in bold) and steps:

Selection
Select Medical Coverage (Single Step Assignment)
Select Dental Coverage (Single Step Assignment)
Select Vision Coverage (Single Step Assignment)

Completion
Finalize Enrollment



1. Personal Information	2. Benefit Election	3. Dependent Informatio	4. Review and Finish
Confirm Personal Infor	Select Medical Coverag	Add Dependents	Confirm Benefit Electio
	Select Dental Coverage		Confirm Dependent Co
	Select Vision Coverage		

Reference Information

- PDN: Design Your Application with Case Lifecycle Management (Node ID: 46551)

Module 04: Creating an Effective Data Model

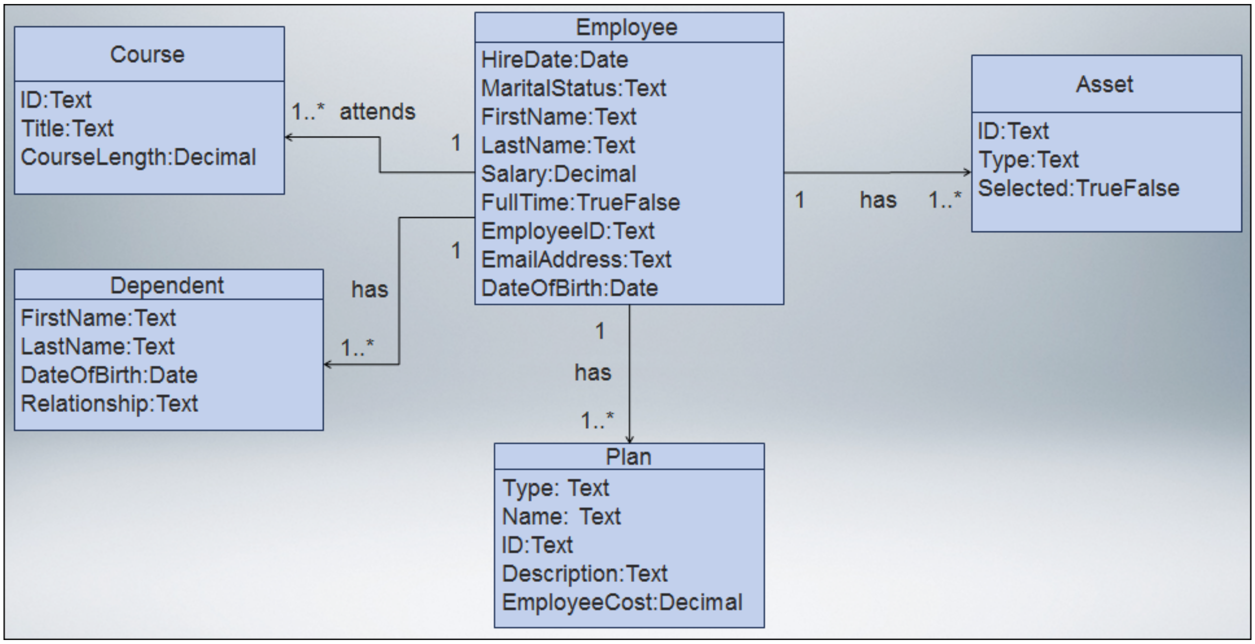
This lesson group includes the following lessons:

- Best Practices for Designing a Data Model (no exercise)
- Best Practices for Managing Data
- Best Practices for Managing Reference Data
- Sharing Data Across Cases and Subcases
- Guardrails for Data Models

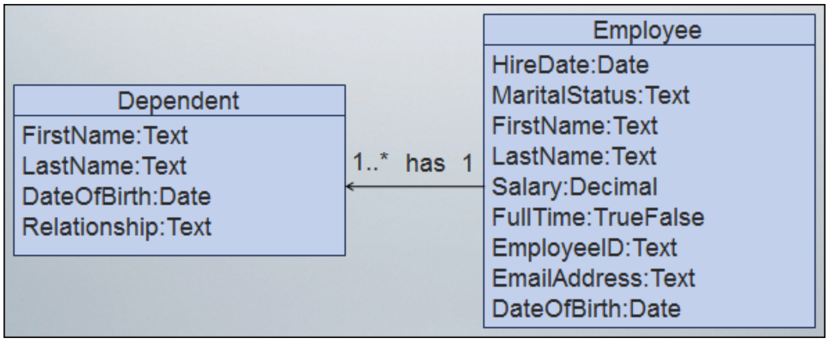
Exercise: Implementing a Data Model Using Data Classes

Scenario

It is time to start implementing the data model for the Onboarding application. Shown below is the complete data model that we will use throughout the lab exercises. You will implement this data model over a series of labs. The central data object you will work with is the Employee object.



You will start by implementing data classes for an employee and their dependents. The Employee and Dependent data model is shown below:



After you have created your data classes, add them as properties to the work class for the Onboarding case type.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Create the Data Objects

Use the data explorer to create a new Employee data object type and a new Dependent data object type.

Both data objects types should inherit from the following classes:

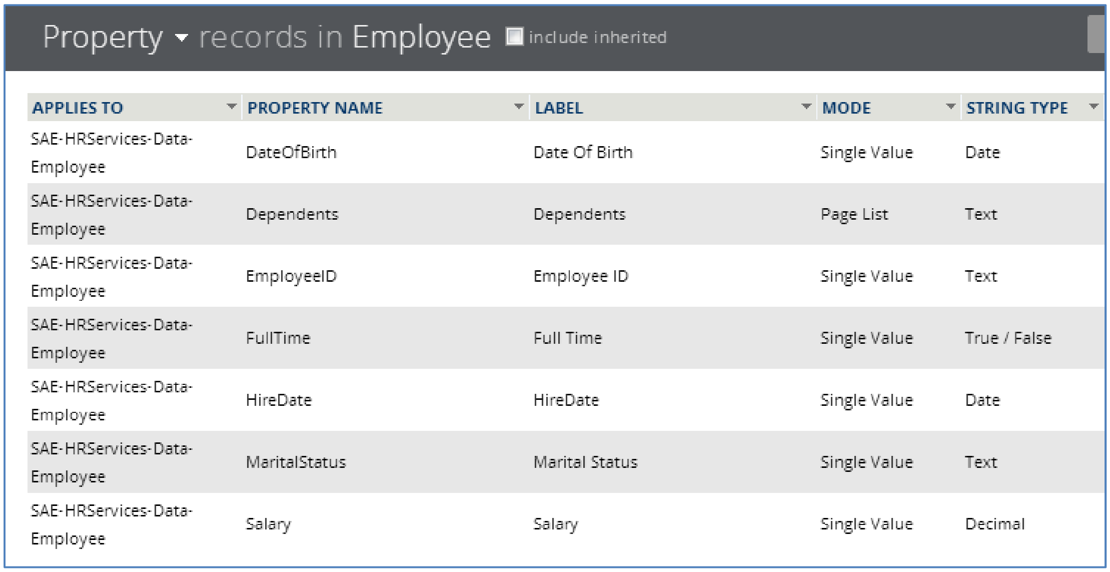
Inheritance Type	Value
Parent Class (Directed)	Data-Party
Parent Class (Pattern)	SAE-HRServices-Data

Use the Employee and Dependent data model (as described in the Scenario section) to add the properties needed for the Employee and Dependent data object types.

The Data-Party class already contains several commonly used properties. For example it already has properties for first name (pyFirstName), last name (pyLastName), and email address (pyEmail1). Therefore you only need to implement the new properties.

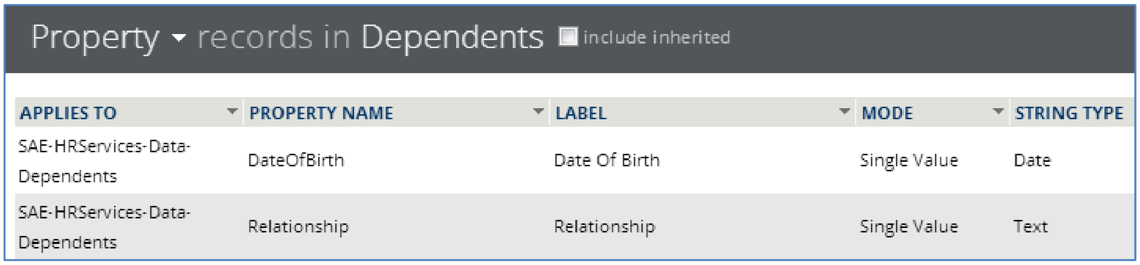
After creating the Dependent data object type, add a page list property to the Employee data object called Dependents to track it.

When completed. if you select these classes in the Data Explorer, you should see something similar to the following:



Property ▾ records in Employee ☐ include inherited

APPLIES TO	PROPERTY NAME	LABEL	MODE	STRING TYPE
SAE-HRServices-Data-Employee	DateOfBirth	Date Of Birth	Single Value	Date
SAE-HRServices-Data-Employee	Dependents	Dependents	Page List	Text
SAE-HRServices-Data-Employee	EmployeeID	Employee ID	Single Value	Text
SAE-HRServices-Data-Employee	FullTime	Full Time	Single Value	True / False
SAE-HRServices-Data-Employee	HireDate	HireDate	Single Value	Date
SAE-HRServices-Data-Employee	MaritalStatus	Marital Status	Single Value	Text
SAE-HRServices-Data-Employee	Salary	Salary	Single Value	Decimal

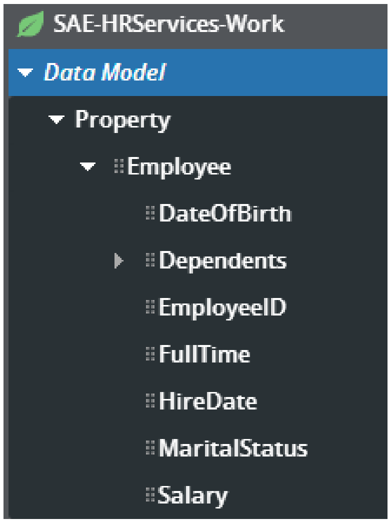


Property ▾ records in Dependents ☐ include inherited

APPLIES TO	PROPERTY NAME	LABEL	MODE	STRING TYPE
SAE-HRServices-Data-Dependents	DateOfBirth	Date Of Birth	Single Value	Date
SAE-HRServices-Data-Dependents	Relationship	Relationship	Single Value	Text

Add Employee to the work class

Now that you have created your data classes, using the Application Explorer add an Employee property to the SAE-HRServices-Work class. After completion your SAE-HRServices-Work class should look similar to the following:

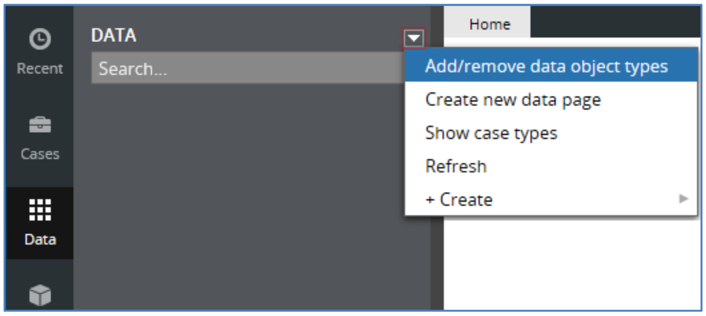


SAE-HRServices-Work

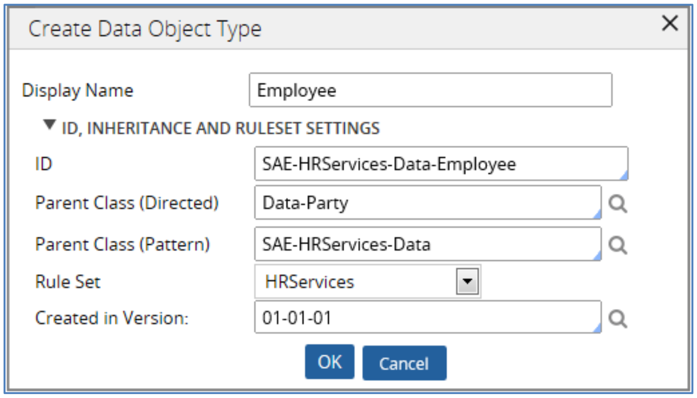
- Data Model
  - Property
    - Employee
      - DateOfBirth
      - Dependents
      - EmployeeID
      - FullTime
      - HireDate
      - MaritalStatus
      - Salary

Hints

1. Select **Add/remove data object types** from the Data Explorer menu to begin creating a new data class.



2. The Create Data Object Type dialog should include references to the Data-Party (Directed) and SAE-HRServices-Data (Pattern) parent classes as shown below.


Create Data Object Type

Display Name: Employee

▼ ID, INHERITANCE AND RULESET SETTINGS

ID: SAE-HRServices-Data-Employee

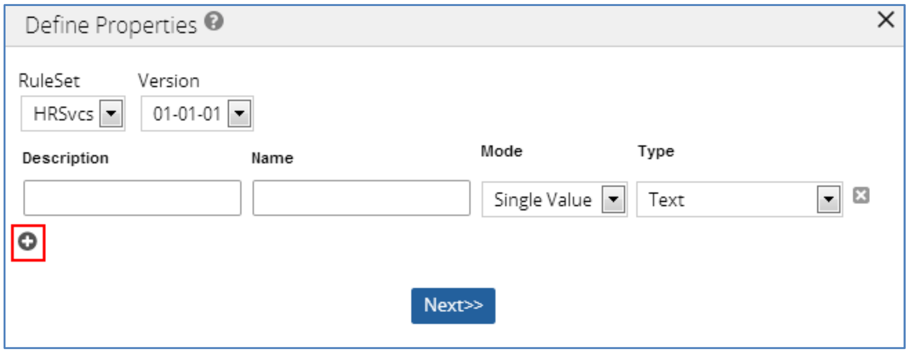
Parent Class (Directed): Data-Party

Parent Class (Pattern): SAE-HRServices-Data

Rule Set: HRServices

Created in Version: 01-01-01

3. To add multiple properties to a data class, click the Add Item icon.


Define Properties

RuleSet: HRSvcs

Version: 01-01-01

Description	Name	Mode	Type
		Single Value	Text

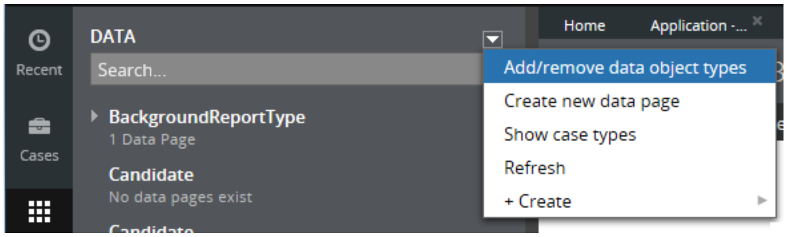
4. Right-click the SAE-HRServices-Work class to add a new property.

Procedure

Follow the steps below:

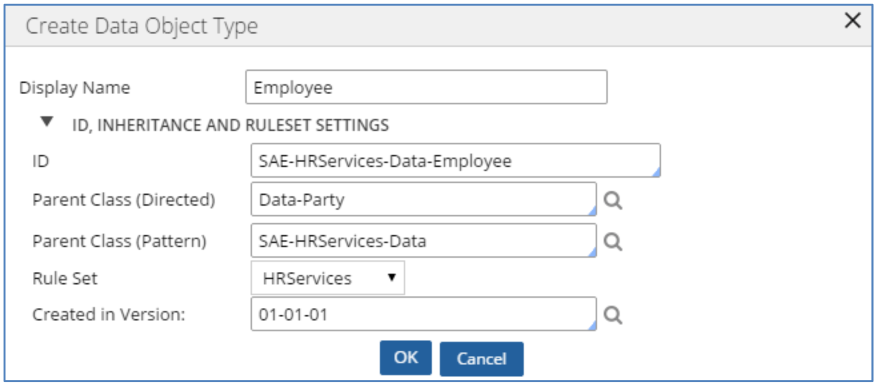
1. Go to the Data Explorer.

2. Use the Data Explorer menu to add each new data object type.



3. Create the Employee data object type using details in the Scenario section.

4. Expand the ID, INHERITANCE AND RULESET SETTINGS option to change the directed and pattern class inheritance as specified in the Approach section.



Create Data Object Type

Display Name: Employee

▼ ID, INHERITANCE AND RULESET SETTINGS

ID: SAE-HRServices-Data-Employee

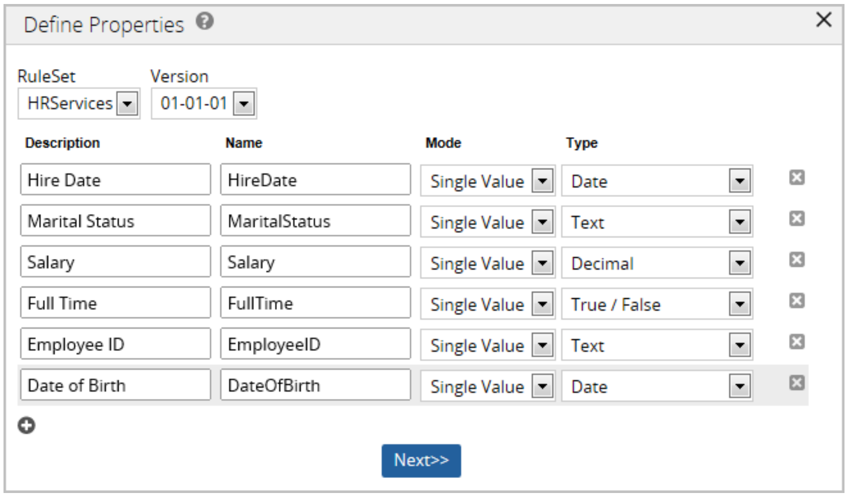
Parent Class (Directed): Data-Party

Parent Class (Pattern): SAE-HRServices-Data

Rule Set: HRServices

Created in Version: 01-01-01

5. In the Define Properties dialog, ensure all property types match those listed in the scenario.



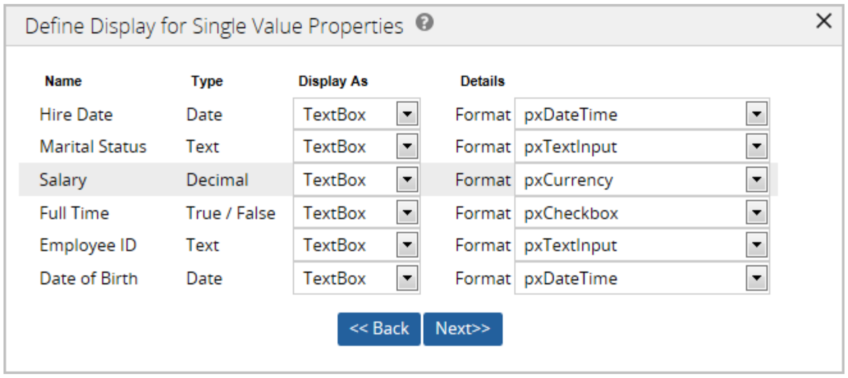
Define Properties

RuleSet: HRServices

Version: 01-01-01

Description	Name	Mode	Type
Hire Date	HireDate	Single Value	Date
Marital Status	MaritalStatus	Single Value	Text
Salary	Salary	Single Value	Decimal
Full Time	FullTime	Single Value	True / False
Employee ID	EmployeeID	Single Value	Text
Date of Birth	DateOfBirth	Single Value	Date

6. In the Define Display for Single Value Properties dialog, the Salary display format can be changed to pxCurrency.

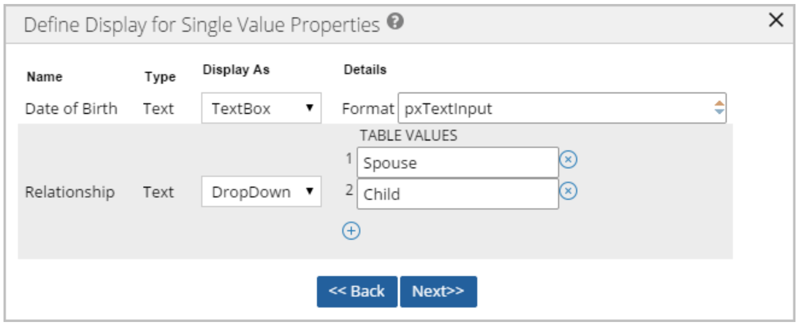


Define Display for Single Value Properties

Name	Type	Display As	Details
Hire Date	Date	TextBox	Format pxDateTime
Marital Status	Text	TextBox	Format pxTextInput
Salary	Decimal	TextBox	Format pxCurrency
Full Time	True / False	TextBox	Format pxCheckbox
Employee ID	Text	TextBox	Format pxTextInput
Date of Birth	Date	TextBox	Format pxDateTime

Important: Ensure all of the property types have been entered correctly before confirming properties creation. Once a data object type is created, the property types cannot be changed.

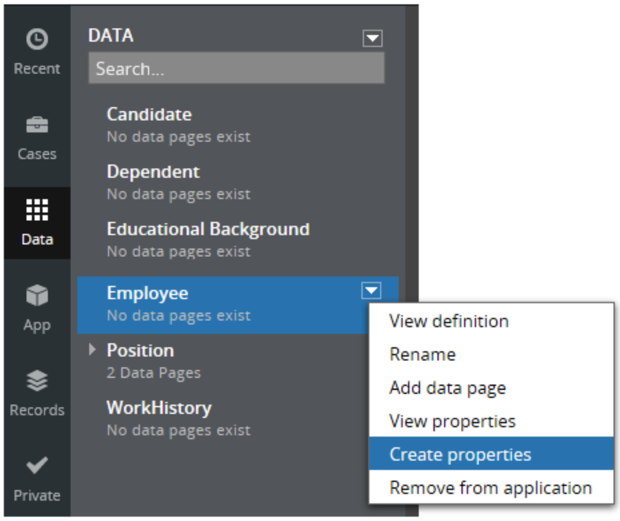
7. Create a Dependent data object type using property details in the Scenario section. Remember to change the directed and pattern inheritance as you did for the Employee data object type. Consider setting the display as a DropDown list and entering table values during creation.



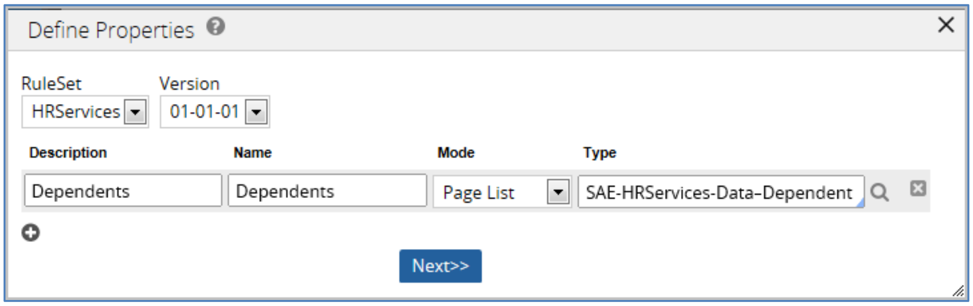
Define Display for Single Value Properties

Name	Type	Display As	Details
Date of Birth	Text	TextBox	Format pxTextInput
Relationship	Text	DropDown	TABLE VALUES 1 Spouse 2 Child

8. Add a Dependents page list property to the Employee data class by right-clicking **Employee** in the Data Explorer and selecting **Create Properties** from the options menu.



9. Select the Dependents page type SAE-HRServices-Data-Dependent from the list.

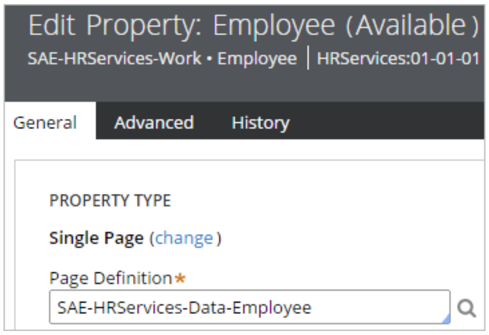


Define Properties

RuleSet: HRServices   Version: 01-01-01

Description	Name	Mode	Type
Dependents	Dependents	Page List	SAE-HRServices-Data-Dependent

10. Go to the Application Explorer and add a new Employee property (Right-click **SAE-HRServices-Work** and select **+Create → Data Model → Property**). Give the new property a label of **Employee**, apply to SAE-HRServices-Work and then click **Create and open**. Change the property type to *Single Page* and set the page definition to SAE-HRServices-Data-Employee.



Edit Property: Employee (Available)

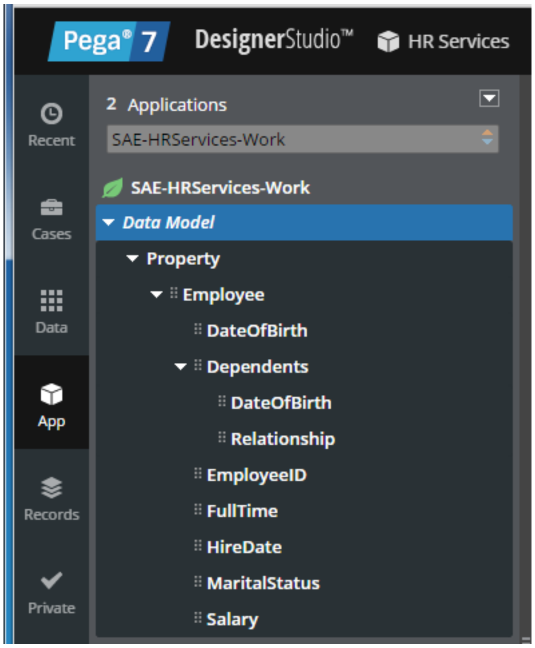
SAE-HRServices-Work • Employee | HRServices:01-01-01

PROPERTY TYPE

Single Page (change)

Page Definition*: SAE-HRServices-Data-Employee

11. Save your changes and then open the application explorer. Try to find your new property in the SAE-HRServices-Work property model.



2 Applications

SAE-HRServices-Work

```
SAE-HRServices-Work
▼ Data Model
  ▼ Property
    ▼ Employee
        DateOfBirth
      ▼ Dependents
          DateOfBirth
          Relationship
        EmployeeID
        FullTime
        HireDate
        MaritalStatus
        Salary
```

Reference Information

- Pega 7 Help: About the Data Explorer

Exercise: Managing Reference Data in Data Tables

Scenario

In this exercise you will continue to build the data model for the Onboarding case type in the HR Services application. An Onboarding case needs to access a list of courses available to new employees, and a list of assets that a manager can allocate to a new employee. Since this information will not change very often, we have determined that it can be stored in local data tables.

In this exercise, you will implement the Asset and Course data objects as data tables. When you run the data table wizard you have the option to have the wizard create data pages for you. The wizard creates two data pages — a list data page and a lookup data page. The list data page returns all the contents of your data page using a report definition that the wizard creates. The lookup data page takes a parameter and returns a row from your data table. The Course and Asset data objects can be seen below:

Course
ID: Text Title: Text CourseLength:Decimal

Asset
ID: Text Type: Text Selected:TrueFalse

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

You will use the data table wizard to create two data tables and the data pages that are associated with them.

Build the Courses data table based on the table below:

Property	Value
Class Name	SAE-HRServices-Data-Courses
Description	Courses data table
Derives From	Data-
Create Data Pages	checked
Lookup Data Page	D_CourseLookup
List Data Page	D_CoursesList
Properties	Use Course data object defined in Scenario

Use default for any values not listed.

Build the Assets data table based table below:

Property	Value
Class Name	SAE-HRServices-Data-Assets
Description	Assets data table
Derives From	Data-
Create Data Pages	checked
Lookup Data Page	D_AssetLookup
List Data Page	D_AssetsList
Properties	Use Asset data object defined in Scenario

Populate the courses data table with a list of courses that a new hire might be required to take. Use the table below to add the course data to the Courses data table.

ID	Label	Course Title	Course Length
001	Product Introduction	Introduction to Application Development	2.5
002	Advanced Course	Advanced Application Development	2.5
003	Business Analysis	Business Development	1
004	Administration	Application Administration	3

Populate the Assets data table using data in the following table.

ID	Label	Asset Type	Selected
F1	Cubicle	Facilities	False
F2	Office	Facilities	False
F3	Phone	Facilities	False
H1	Laptop	Hardware	False
H2	Desktop	Hardware	False
H3	Mouse	Hardware	False
H4	Monitor	Hardware	False
H5	Docking Station	Hardware	False
I1	Email Address	IT	False
S1	MS Office	Software	False
S2	Chrome	Software	False
S4	Virus Protection	Software	False

Finally, to keep track of which assets and courses have been selected for an employee, create two Page List properties as part of the SAE-HRServices-Work class:

- AssetList with a page definition of SAE-HRServices-Data-Assets
- SelectedCourses with a page definition of SAE-HRServices-Data-Courses

Hints

1. Data tables can be found using the Designer Studio menu under Data Models.
2. After you've created a data table you can edit it to add data instances.

Procedure

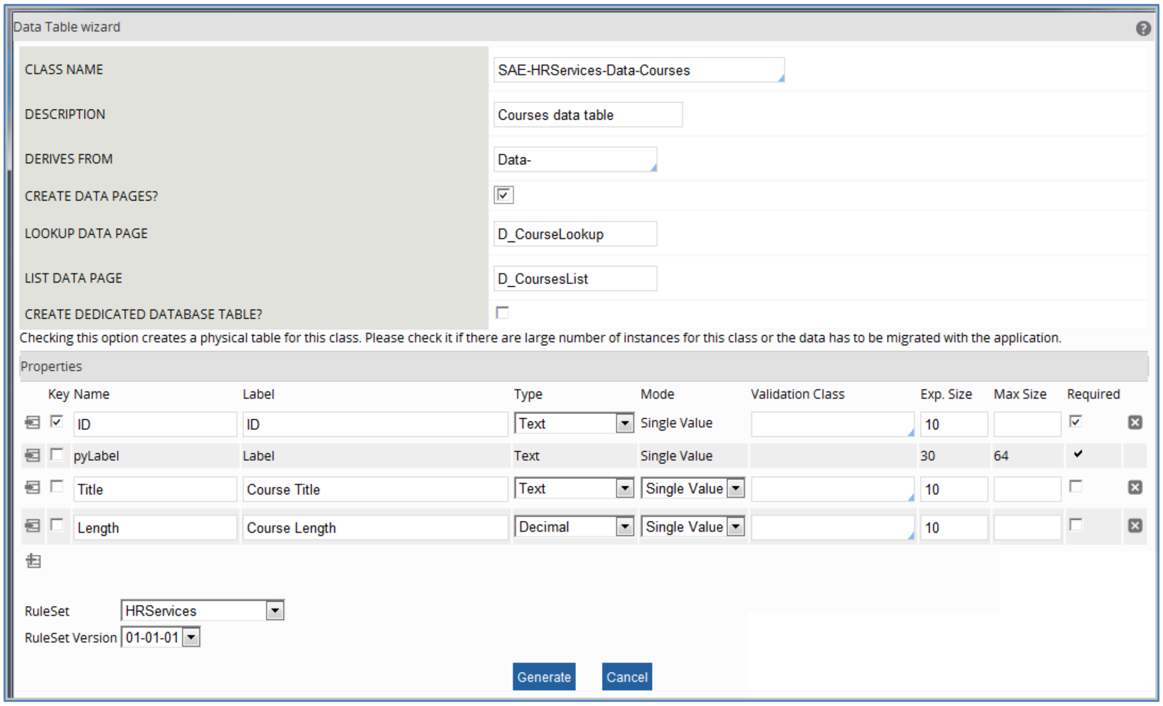
Follow the steps below:

1. Access Data Tables from the Designer Studio menu under Data Model.

   Note: If a new window doesn't appear, allowing you to create a new data table, trying using the Internet Explorer web browser.

2. Add a new Data Table named *Courses* using properties from the Course table described in the Approach section.

   IMPORTANT: Ensure that all properties have the correct Type before generating the data table. Failure to do so will cause your results to vary from those seen in later exercises.



Data Table wizard

CLASS NAME	SAE-HRServices-Data-Courses
DESCRIPTION	Courses data table
DERIVES FROM	Data-
CREATE DATA PAGES?	☑
LOOKUP DATA PAGE	D_CourseLookup
LIST DATA PAGE	D_CoursesList
CREATE DEDICATED DATABASE TABLE?	☐

Checking this option creates a physical table for this class. Please check it if there are large number of instances for this class or the data has to be migrated with the application.

Properties

Key	Name	Label	Type	Mode	Validation Class	Exp. Size	Max Size	Required
☑	ID	ID	Text	Single Value		10		☑
☐	pyLabel	Label	Text	Single Value		30	64	✓
☐	Title	Course Title	Text	Single Value		10		☐
☐	Length	Course Length	Decimal	Single Value		10		☐

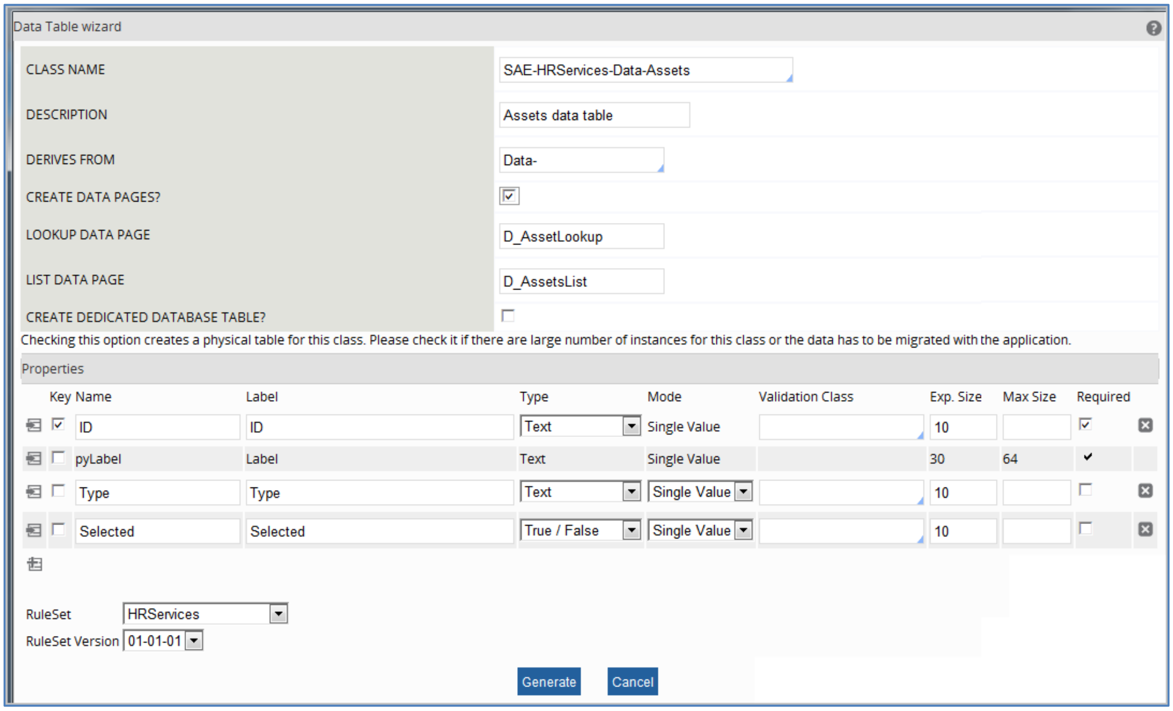
RuleSet: HRServices

RuleSet Version: 01-01-01

Note: If you expect this table to contain more than 500 rows, select **Create Dedicated Database Table**. This will cause the wizard to associate a new database table to hold the rows of this data table, and not use the default pr_other table.

3. To populate the Courses database table, use the **List Editor** link in the Data Table wizard results dialog or the Edit icon in the Data Model – Data Tables list.

4. Add a new Data Table named Assets using properties from the Asset table described in the Approach section. Remember to set the correct property types before generating the table.



Data Table wizard

CLASS NAME	SAE-HRServices-Data-Assets
DESCRIPTION	Assets data table
DERIVES FROM	Data-
CREATE DATA PAGES?	☑
LOOKUP DATA PAGE	D_AssetLookup
LIST DATA PAGE	D_AssetsList
CREATE DEDICATED DATABASE TABLE?	☐

Checking this option creates a physical table for this class. Please check it if there are large number of instances for this class or the data has to be migrated with the application.

Properties

Key	Name	Label	Type	Mode	Validation Class	Exp. Size	Max Size	Required
☑	ID	ID	Text	Single Value		10		☑
☐	pyLabel	Label	Text	Single Value		30	64	✓
☐	Type	Type	Text	Single Value		10		☐
☐	Selected	Selected	True / False	Single Value		10		☐

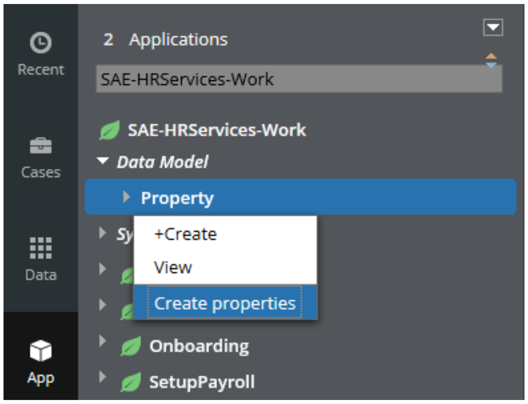
RuleSet HRServices

RuleSet Version 01-01-01

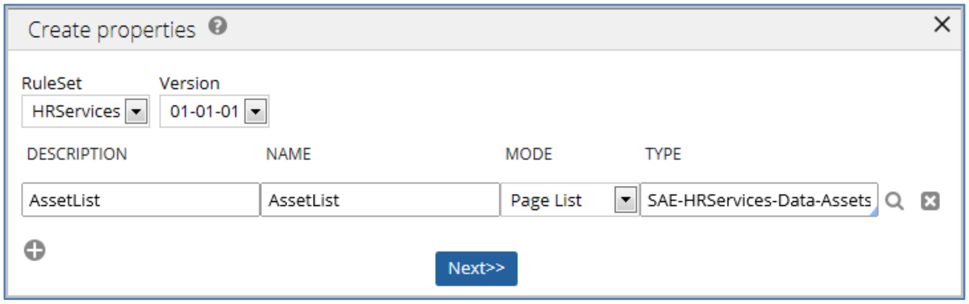
Generate Cancel

5. Populate both the Courses and Assets data tables with the data described in the Approach section.

6. From the Application Explorer, create a new property in the context of SAE-HRServices-Work.



7. Name the new property *AssetList*. Set the property mode to Page List and the page definition to SAE-HRServices-Data-Assets.



RuleSet: HRServices

Version: 01-01-01

DESCRIPTION	NAME	MODE	TYPE
AssetList	AssetList	Page List	SAE-HRServices-Data-Assets

8. Add a page list property named *SelectedCourses* with the same record context used for AssetList. Set the page definition to SAE-HRServices-Data-Courses.

   Note: You can also create the property using **Create → Data Model → Property**.

9. Refresh the Application Explorer SAE-HRServices-Work class and then expand **Data Model → Property** to view the new page list properties.

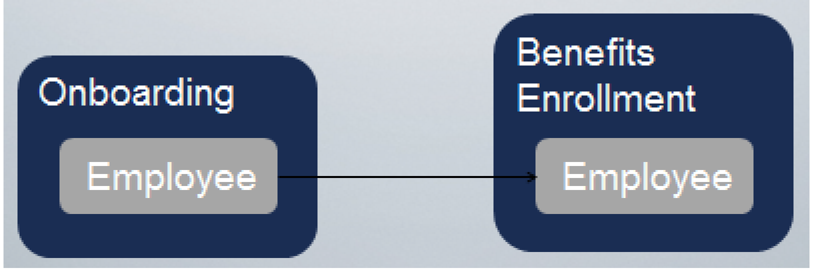
Reference Information

- Pega 7 Help: Data Model category — Data Tables page

Exercise: Propagating Data from a Case to a Subcase

Scenario

When a case creates a subcase it is often times necessary for the case to propagate data to the subcase. In the HRServices application, the Onboarding case creates a BenefitsEnrollment subcase. When this occurs, the Employee data that has been collected needs to be propagated to the BenefitsEnrollment subcase.



In this exercise you will configure the Onboarding case to propogate the Employee property to the BenefitsEnrollment subcase.

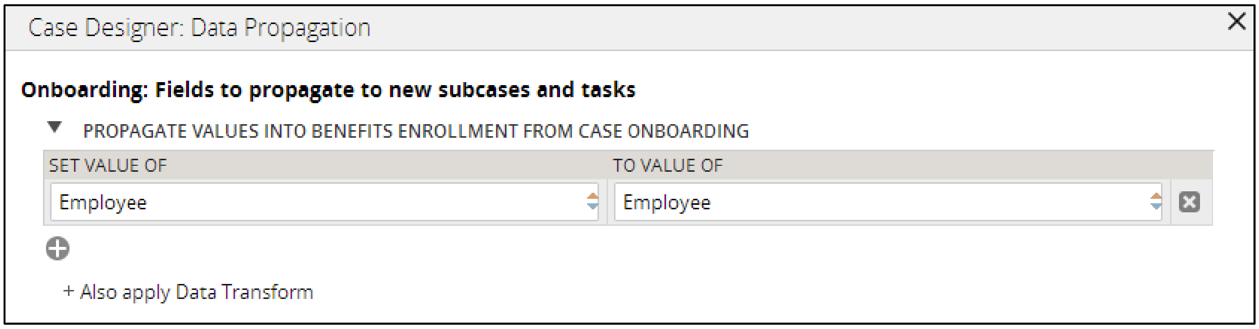
- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

In the Case Explorer, configure data propagation for the Onboarding case.

Copy the Employee property from the Onboarding case to the Employee property of the Benefits Enrollment subcase.

When completed your configuration should look similar to:



Case Designer: Data Propagation

Onboarding: Fields to propagate to new subcases and tasks

▼ PROPAGATE VALUES INTO BENEFITS ENROLLMENT FROM CASE ONBOARDING

SET VALUE OF	TO VALUE OF
Employee	Employee

+ Also apply Data Transform

Hints

1. Data Propagation can be found on the Details tab for a case.

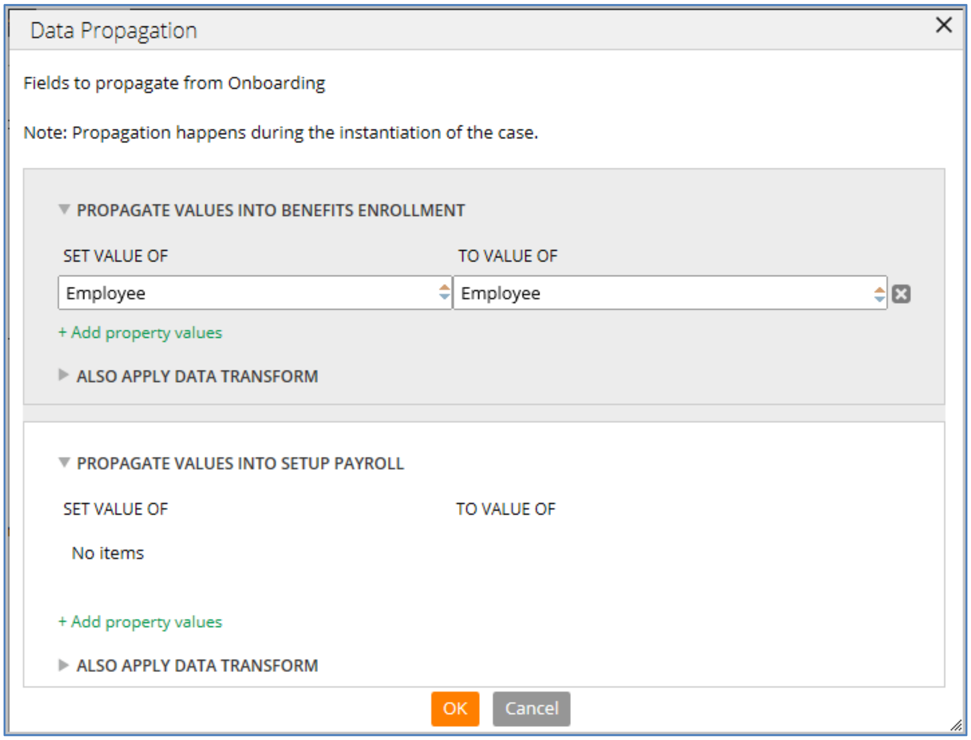
Procedure

Follow the steps below:

1. Go to the Case Designer for the Onboarding case.

2. Data Propagation can be found on the details tab.

3. Select the Employee class from the SAE-HRServices-Work class for both the Set Value Of and To Value Of fields.



Data Propagation

Fields to propagate from Onboarding

Note: Propagation happens during the instantiation of the case.

▼ PROPAGATE VALUES INTO BENEFITS ENROLLMENT

SET VALUE OF	TO VALUE OF
Employee	Employee

▼ PROPAGATE VALUES INTO SETUP PAYROLL

SET VALUE OF	TO VALUE OF
No items	

Reference Information

- Pega 7 Help: About Propagation

Exercise: Adding Default Data Transforms

Scenario

One of the most common tasks in any application development effort is initializing data objects. In Pega 7, we use data transforms to accomplish this tasks. No coding is required, and the simple interface allows any level of developer to complete this task quickly and easily.

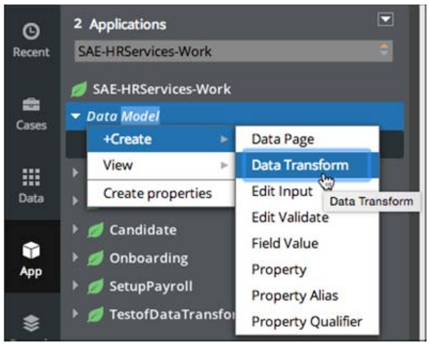
In some situations, such as when we create data objects using the *Configure form* option in the case designer, a default pair of data transforms – *pyDefault* and *pySetFieldDefaults* - are created automatically.

However, when we create data objects using the Data Explorer (as we have done in this course) or directly in the Application Explorer, these two key data transforms may need to be created manually.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Procedure

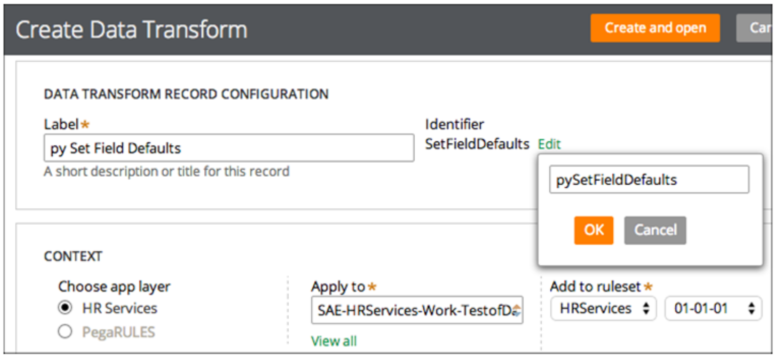
1. Switch to the **App Explorer**
2. Expand **SAE-HRServices-Work** and then right-click on **Data Model**.
3. From the context menu, select **Create** > **Data Transform**.



4. Enter **.pySetFieldDefaults** in the Label field.

Tip: When creating data transforms to initialize data objects, use the default data transform names – pySetFieldDefaults and pyDefault. This allows you to maintain a consistent format with PRPC’s default configuration.

5. Click **Edit** and change the Identifier to **pySetFieldDefaults**, and then click **OK**.



Create Data Transform

DATA TRANSFORM RECORD CONFIGURATION

Label: py Set Field Defaults

A short description or title for this record

Identifier: SetFieldDefaults Edit

pySetFieldDefaults

CONTEXT

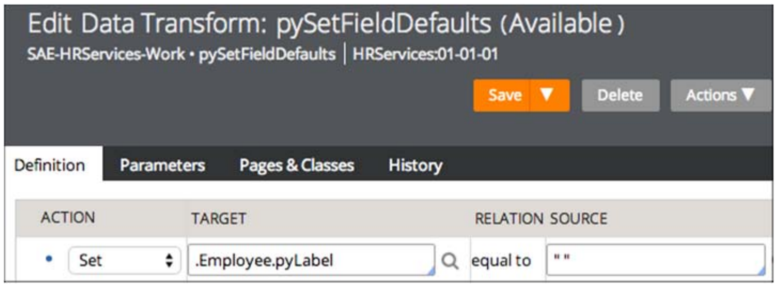
Choose app layer: ● HR Services ○ PegaRULES

Apply to: SAE-HRServices-Work-TestofDa

Add to ruleset: HRServices 01-01-01

Important: Ensure the Identifier is pySetFieldDefaults, as shown in the image above.

6. Click **Create and open**.
7. Specify an action to *Set* the *Target* **.Employee.pyLabel** *equal to* **" "**, and then click **Save**.



Edit Data Transform: pySetFieldDefaults (Available)

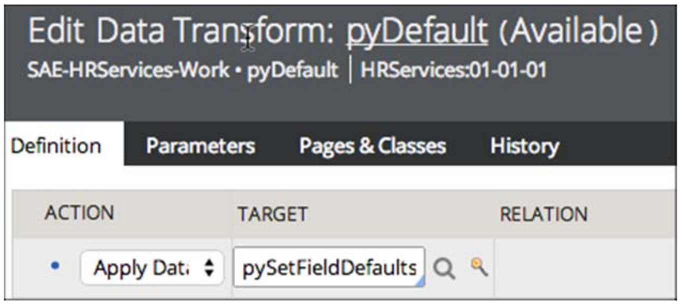
SAE-HRServices-Work • pySetFieldDefaults | HRServices:01-01-01

ACTION	TARGET	RELATION	SOURCE
Set	.Employee.pyLabel	equal to	" "

Tip: You can use any property defined in the .Employee data object. In our example, we use .pyLabel because it is the same property used when PRPC creates this data transform automatically. Using .pyLabel allows us to maintain a consistent format with PRPC.

8. Refresh the App Explorer to see the new data transform.
9. Using the same procedure as in steps 2 & 3, create another data transform.
10. Enter **pyDefault** in the Label field, click **Edit** to change the Identifier to **pyDefault** and then click **Create and open**.

11. Change the Action to **Apply Data Transform**, set the Target to **pySetFieldDefaults**, and then click **Save.**



Edit Data Transform: pyDefault (Available)

SAE-HRServices-Work • pyDefault | HRServices:01-01-01

ACTION	TARGET	RELATION
Apply Dat:	pySetFieldDefaults	

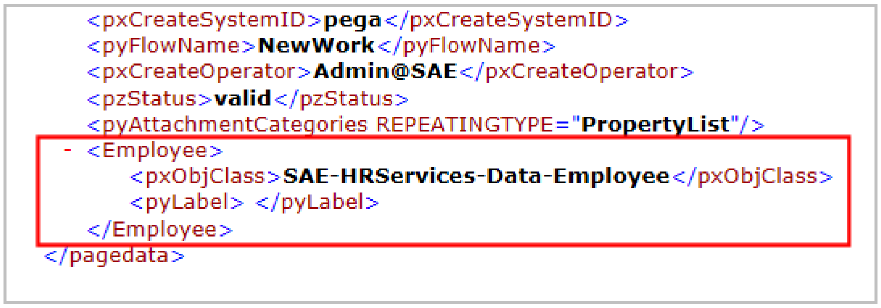
12. Refresh the App Explorer to see the new data transform.

Testing the Data Transforms

Test the data transforms to ensure they execute correctly and produce the expected result.

1. Ensure the pyDefault data transform is open in the work area.
2. Click the **Actions** button in the upper right corner and select **Run** from the menu. A new (pop-up) window will open.
3. Click **Execute** in the new window.

A second pop-up window will open with the results. Most of the elements are related to default properties set by PRPC, however the Employee data object should appear at the bottom of the list.



```
    <pxCreateSystemID>pega</pxCreateSystemID>
    <pyFlowName>NewWork</pyFlowName>
    <pxCreateOperator>Admin@SAE</pxCreateOperator>
    <pzStatus>valid</pzStatus>
    <pyAttachmentCategories REPEATINGTYPE="PropertyList"/>
  - <Employee>
        <pxObjClass>SAE-HRServices-Data-Employee</pxObjClass>
        <pyLabel> </pyLabel>
    </Employee>
</pagedata>
```

Exercise: Creating an Effective Data Model Exercise Verification

Scenario

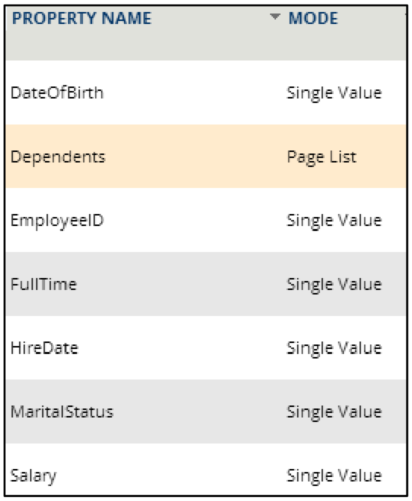
Data Management controls how the data in your application is labeled, accessed, and grouped together for storage, retrieval, and display.

In this lesson group you built the Data architecture needed to support the Onboarding case type. This was done with a combination of data objects, data pages, and data tables. You also needed to pass data from the Onboarding case to the Benefits Enrollment subcase.

Approach

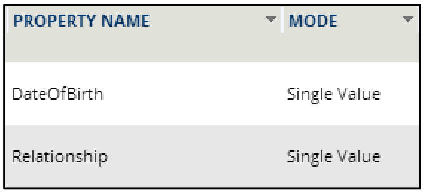
After completing the exercises in this lesson group you should have the data constructs shown below.

- Data object types – Use the Data Explorer to confirm that your results match.
    - SAE-HRServices-Data-Employee, which has the following properties:



PROPERTY NAME	MODE
DateOfBirth	Single Value
Dependents	Page List
EmployeeID	Single Value
FullTime	Single Value
HireDate	Single Value
MaritalStatus	Single Value
Salary	Single Value

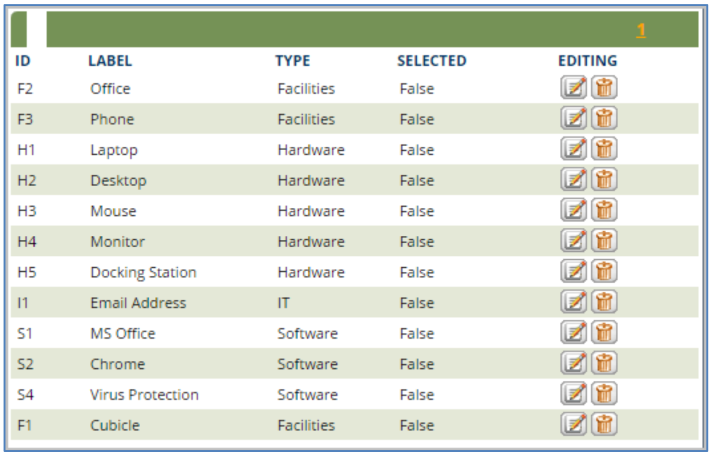
    - SAE-HRServices-Data-Dependent, which has the following properties:



PROPERTY NAME	MODE
DateOfBirth	Single Value
Relationship	Single Value

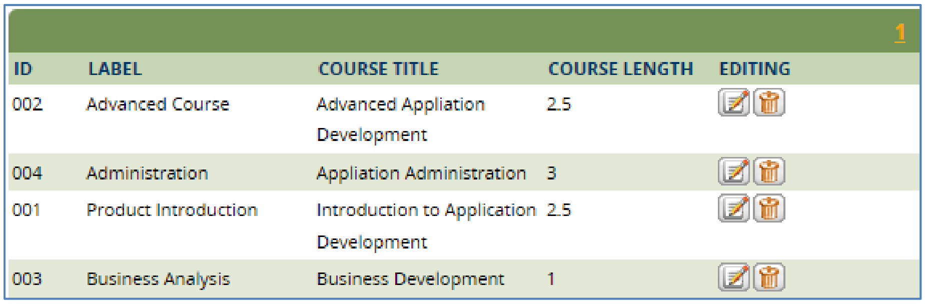
- Data tables – Use the **Designer Studio → Data Model → Data Tables** menu option and then the Open icon to confirm your results match.

- Assets Data Table



ID	LABEL	TYPE	SELECTED	EDITING
F2	Office	Facilities	False	✎ 🗑
F3	Phone	Facilities	False	✎ 🗑
H1	Laptop	Hardware	False	✎ 🗑
H2	Desktop	Hardware	False	✎ 🗑
H3	Mouse	Hardware	False	✎ 🗑
H4	Monitor	Hardware	False	✎ 🗑
H5	Docking Station	Hardware	False	✎ 🗑
I1	Email Address	IT	False	✎ 🗑
S1	MS Office	Software	False	✎ 🗑
S2	Chrome	Software	False	✎ 🗑
S4	Virus Protection	Software	False	✎ 🗑
F1	Cubicle	Facilities	False	✎ 🗑

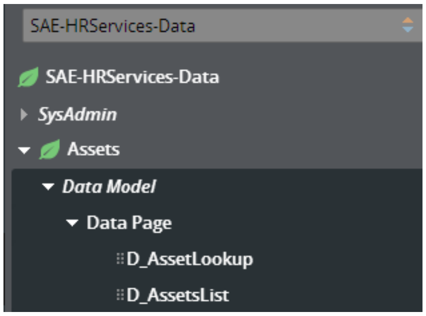
- Courses Data Table



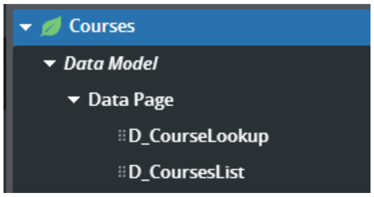
ID	LABEL	COURSE TITLE	COURSE LENGTH	EDITING
002	Advanced Course	Advanced Appliation Development	2.5	✎ 🗑
004	Administration	Appliation Administration	3	✎ 🗑
001	Product Introduction	Introduction to Application Development	2.5	✎ 🗑
003	Business Analysis	Business Development	1	✎ 🗑

- Each data table should have two data pages associated with it; one that does a lookup of a specific object and one that returns a list of objects. You can see this from the Application Explorer by opening the SAE-HRServices-Data class and expanding:

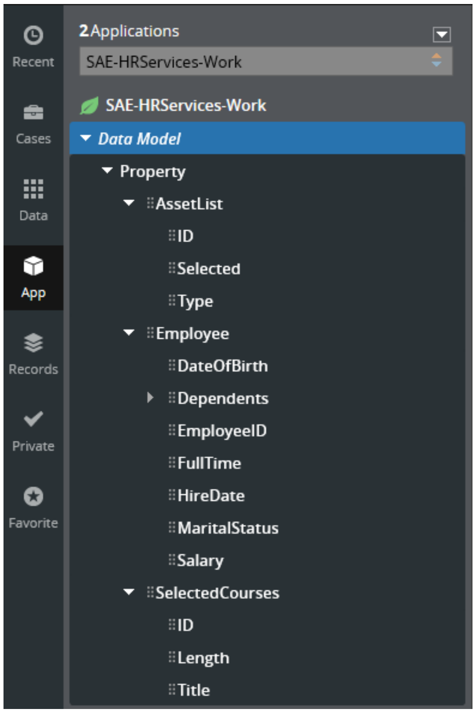
    - Assets → Data Model → Data Page



    - Courses →Data Model →Data Page



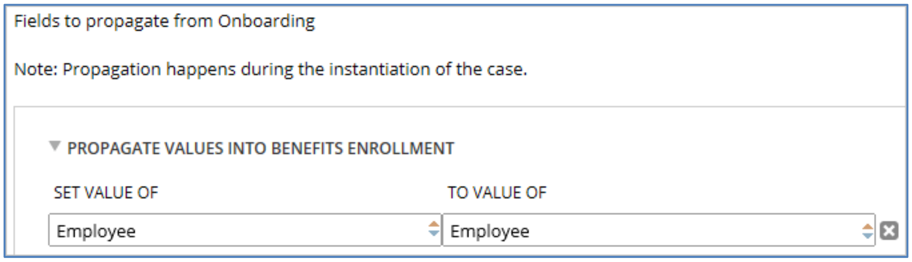
- Three page properties should be in the SAE-HRServices-Work class:
  - AssetList
  - Employee
  - SelectedCourses



SAE-HRServices-Work

- SAE-HRServices-Work
  - Data Model
    - Property
      - AssetList
        - ID
        - Selected
        - Type
      - Employee
        - DateOfBirth
        - Dependents
        - EmployeeID
        - FullTime
        - HireDate
        - MaritalStatus
        - Salary
      - SelectedCourses
        - ID
        - Length
        - Title

- You will also have data propagation configured from the Onboarding case to any subcases that will need access to Employee properties. This can be accessed from the Details tab in the Case Designer landing page for the Onboarding case type.



Fields to propagate from Onboarding

Note: Propagation happens during the instantiation of the case.

PROPAGATE VALUES INTO BENEFITS ENROLLMENT

SET VALUE OF	TO VALUE OF
Employee	Employee

Module 05: Integrating with External Data Sources

This lesson group includes the following lessons:

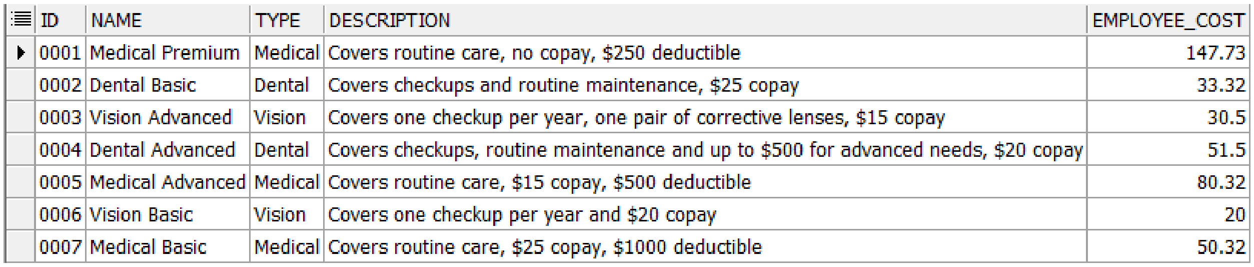
- Integrating with Databases
- Guardrails for Integrating with External Data Sources

Exercise: Using the Database Table Class Mapping Tool

Scenario

Part of the data model will often be in an external database. This is the case with the HR Plan data in our Onboarding application. To map to the external database table, run the Database Table Class Mapping Tool.

The HR Plan table can be seen below:



ID	NAME	TYPE	DESCRIPTION	EMPLOYEE_COST
0001	Medical Premium	Medical	Covers routine care, no copay, $250 deductible	147.73
0002	Dental Basic	Dental	Covers checkups and routine maintenance, $25 copay	33.32
0003	Vision Advanced	Vision	Covers one checkup per year, one pair of corrective lenses, $15 copay	30.5
0004	Dental Advanced	Dental	Covers checkups, routine maintenance and up to $500 for advanced needs, $20 copay	51.5
0005	Medical Advanced	Medical	Covers routine care, $15 copay, $500 deductible	80.32
0006	Vision Basic	Vision	Covers one checkup per year and $20 copay	20
0007	Medical Basic	Medical	Covers routine care, $25 copay, $1000 deductible	50.32

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Run the Database Table Class Mapping tool to map properties in the external database using information in the following table and database columns displayed in the Scenario. When you create your property names, be careful not to rely on default values but to follow the best practices for naming properties.

Property	Value
Database Name	PegaDATA
Table Name	sae2_hr_plan
Class Name	SAE-HRServices-Data-HRPlan

After completing the wizard find the data class that was created. Note how many rows are in the the SAE-HRServices-Data-HRPlan data class: there are _________ rows (records).

Procedure

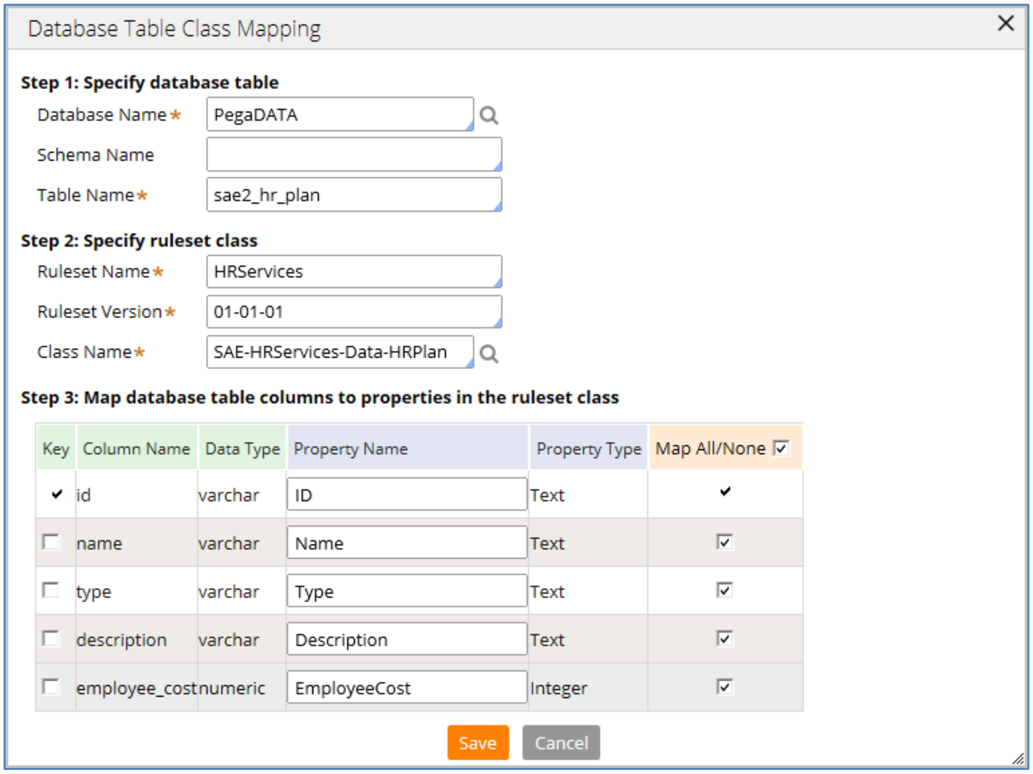
Follow the steps below:

1. View the application's current data model class mappings by selecting **Data Model → Classes and Properties → Database Class Mappings** from the Designer Studio menu. Look at classes and properties in both the Work and the Data class categories.

2. Create a **New External Database Table Class Mapping** using information found in the Approach section and a description of the database table found in the Scenario.

3. Remember to change the Class Name and restyle the property names so they follow the standard camel case naming convention used in Pega applications.

IMPORTANT: Ensure that all values are correct before saving changes. Failure to do so will cause your results to vary from those seen in later exercises.



Database Table Class Mapping

Step 1: Specify database table

Database Name*	PegaDATA
Schema Name	
Table Name*	sae2_hr_plan

Step 2: Specify ruleset class

Ruleset Name*	HRServices
Ruleset Version*	01-01-01
Class Name*	SAE-HRServices-Data-HRPlan

Step 3: Map database table columns to properties in the ruleset class

Key	Column Name	Data Type	Property Name	Property Type	Map All/None ☑
✓	id	varchar	ID	Text	✓
□	name	varchar	Name	Text	☑
□	type	varchar	Type	Text	☑
□	description	varchar	Description	Text	☑
□	employee_cost	numeric	EmployeeCost	Integer	☑

Note: Class name and property names will be given default values, but can be changed to match the example shown above.

Reference Information

- PDN: How to create a data page

Exercise: Integrating with External Data Sources Verification

Scenario

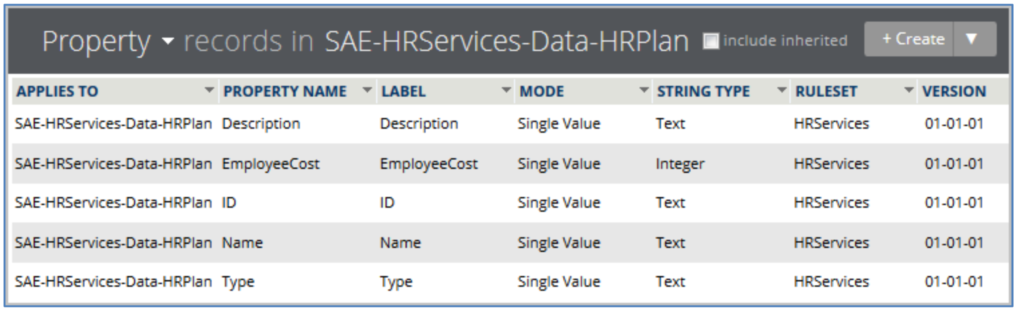
Pega 7's integration facilities provide a set of protocols, contained within a toolkit, that allow users to quickly create connections to—and endpoints for—external systems. By integrating with other systems, designers can leverage existing applications and services.

In this lesson group you were tasked to integrate with an external database table. To do that you used the External Database Table Class Mapping wizard to create a data object to represent the table.

Approach

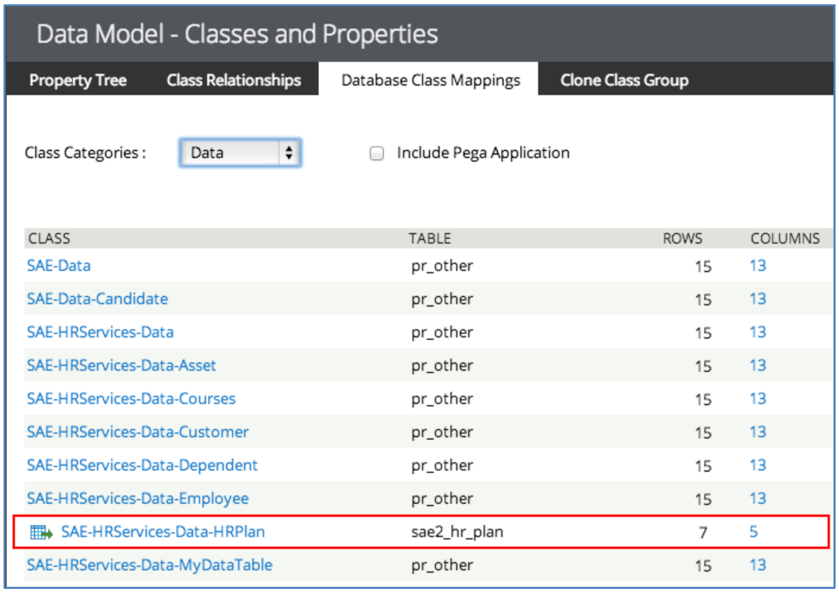
After completing the labs in this lesson group you should have:

An SAE-HRServices-Data-HRPlan data object (as seen in the Data Explorer)



APPLIES TO	PROPERTY NAME	LABEL	MODE	STRING TYPE	RULESET	VERSION
SAE-HRServices-Data-HRPlan	Description	Description	Single Value	Text	HRServices	01-01-01
SAE-HRServices-Data-HRPlan	EmployeeCost	EmployeeCost	Single Value	Integer	HRServices	01-01-01
SAE-HRServices-Data-HRPlan	ID	ID	Single Value	Text	HRServices	01-01-01
SAE-HRServices-Data-HRPlan	Name	Name	Single Value	Text	HRServices	01-01-01
SAE-HRServices-Data-HRPlan	Type	Type	Single Value	Text	HRServices	01-01-01

An entry in the database class mappings for the SAE-HRServices-Data-HRPlan (as seen in the Data Model landing page).



Class Categories : Data

☐ Include Pega Application

CLASS	TABLE	ROWS	COLUMNS
SAE-Data	pr_other	15	13
SAE-Data-Candidate	pr_other	15	13
SAE-HRServices-Data	pr_other	15	13
SAE-HRServices-Data-Asset	pr_other	15	13
SAE-HRServices-Data-Courses	pr_other	15	13
SAE-HRServices-Data-Customer	pr_other	15	13
SAE-HRServices-Data-Dependent	pr_other	15	13
SAE-HRServices-Data-Employee	pr_other	15	13
SAE-HRServices-Data-HRPlan	sae2_hr_plan	7	5
SAE-HRServices-Data-MyDataTable	pr_other	15	13

Module 06: Creating Engaging User Experiences

This lesson group includes the following lessons:

- Designing the User Interface for Reuse and Maintainability
- Building Assignment Focused (Intent-Driven) User Interfaces
- Best Practices for Designing the User Interface (no exercise)
- Using Advanced User Interface Controls
- Managing Data for Selectable List Controls
- Building Dynamic User Interfaces
- Validating User Input
- Guardrails for Creating Engaging User Experiences

Exercise: Creating Reusable User Interface Components

Scenario

During the initial elaboration phases of case type design, most likely “draft” UI screens will be created while configuring process detail using the Case Designer. If this happens, sections are typically created in the ORG-Application-Work-<CaseTypeName> class.

When organizing UI components for reusability, it is possible some of those sections would need to be not only applied to a more appropriate class, but redrawn as well. In our development effort for the Onboarding case type, we have purposefully chosen to take a more manual approach. The goal of this exercise is to create the UI sections directly in the most appropriate classes, then work towards exposing the completed UI (in a later exercise) through the process flows, where needed, using flow actions.

It would be unrealistic, and possibly cause you to lose track of the exercise, to attempt to create all of the necessary UIs for the Onboarding case at once. In this exercise, we will focus on creating the two primary sequences of screens – the screen used to create the employee record and the screens used for selecting healthcare plans, including adding dependents. Once you have mastered creating and implementing these UI components, you can add additional screens to other parts of the case type as desired.

When an Onboarding case is started, the HR representative should be able to create an employee record with the following, minimum information:

- First name (text, required)
- Last name (text, required)
- Full Time (yes or no)
- Email address (text, required)
- Hire Data (date)
- Marital Status (yes or no)

During orientation, new employees should be able to sign up for medical, dental and vision healthcare plans. HR provides a list of plans in each category, along with the cost of each plan and a description. The information needed for enrolling in these plans includes:

- Benefit type (text)
- Benefit plan name (text)
- Benefit plan description (text)
- Employee cost (currency)

Employees can also add dependents to their plan if desired. Dependent information should include the dependent’s name, date of birth and relationship to the employee.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

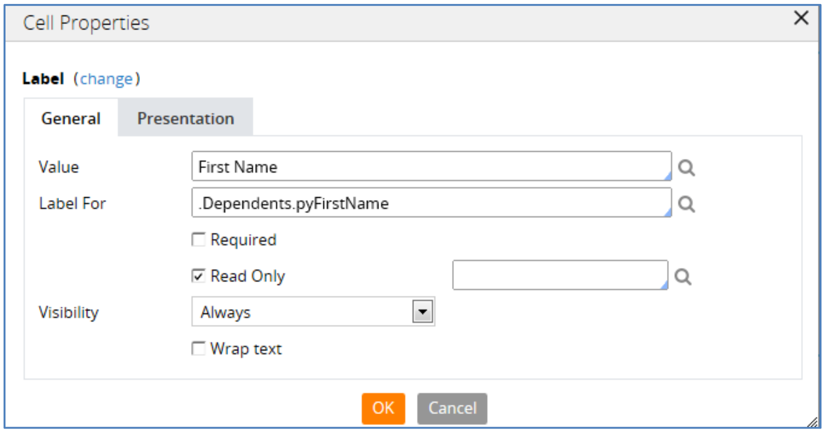
Approach

Our approach would be to create the employee information, benefits selection and dependents sections in the data class of each associated data object type. This will allow for greater reusability and is in keeping with the spirit of *Build for Change*. We will use these base sections to implement the associated assignments in a later exercise.

The HR Benefit Plans database table, which we mapped to the HRPlan data object, contains plan options for all plan types. In this early phase of development, we are only concerned with creating the basic section layout. When we use the data, in a later exercise, the plans will be filtered by plan type (Medical, Dental and Vision) by adding a parameter to the section.

Hints

1. Open the Application Explorer to view the SAE-HRServices-Data class tree. Create a new user interface section by right-clicking the associated data object type (such as Employee) and using the pull-down menu to select **New > User Interface > Section** within that class.
2. To add fields to a section:
    a. Drag fields from the data model property to the section.
    b. For fields inherited from the Data-Party class, create a Basic Text Input field. View cell properties and select from the **more** options in the Property auto-complete field to find the property name.
3. When adding a label to Dependents Cell Properties, remember to specify that fields are part of the Dependents property within Employee.


Cell Properties

Label (change)

General | Presentation

Value: First Name

Label For: .Dependents.pyFirstName

☐ Required

☑ Read Only

Visibility: Always

☐ Wrap text

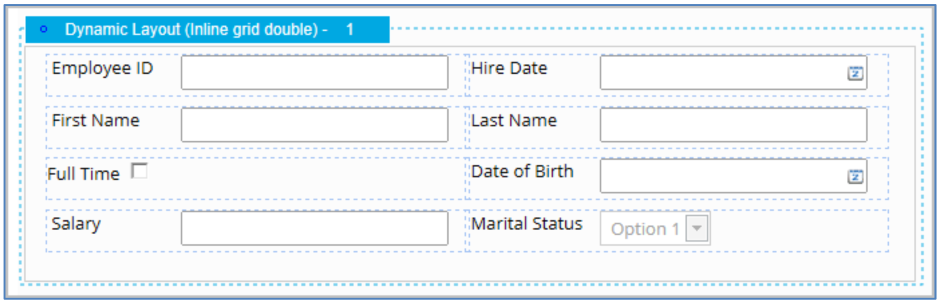
Procedure

Begin this exercise by opening the Application Explorer to view the SAE-RHServices-Data class tree. Then, follow the steps below:

Section used for creating the employee record

1. Create a new UI section used to display the Employee information in the record context of SAE-HRServices-Data-<class name of employee data object type>.
   - From Application Explorer, **SAE-HRServices-Data** class, right-click **Employee** and select **Create+ > User Interface > Section**.
   - Enter **Employee Information** in the Label field and ensure the section applies to SAE-HRServices-Data-Employee.

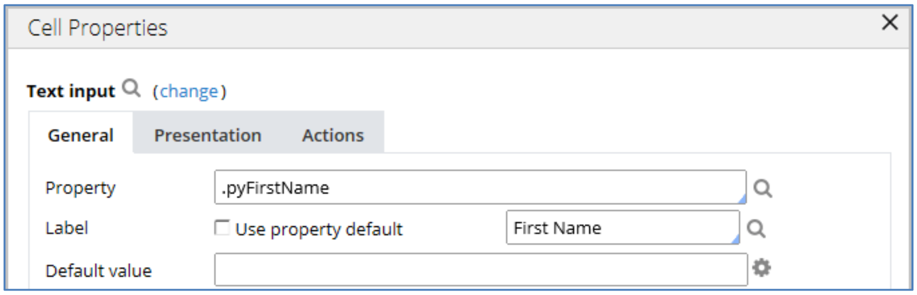
2. When the Section rule form appears:
   a. Change the layout to an **Inline grid double**.
   b. Add controls for the properties noted in the scenario.
   c. Consider using a Dropdown control for selecting the marital status.You can use the Property rule form, Display and Validation section to create a local list and fill in table values to be used for dropdown choices.



Dynamic Layout (Inline grid double) - 1

Employee ID		Hire Date	
First Name		Last Name	
Full Time □		Date of Birth	
Salary		Marital Status	Option 1

Note: Remember, we are inheriting the Pega provided properties for first and last name, which can be found in Data-Party.



Cell Properties

Text input (change)

General | Presentation | Actions

Property .pyFirstName

Label □ Use property default First Name

Default value

Section used for selecting healthcare benefits

3. Create a new UI section used to display the HR Plan information.

    a. Enter **Healthcare Benefit** in the Label field.

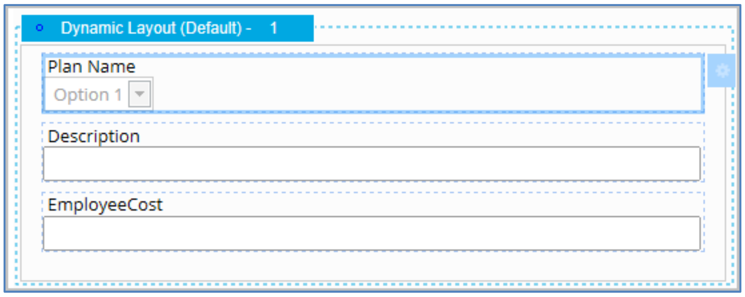
    b. The Record Context should apply to the class for the healthcare plans (SAE-HRServices-Data-HRPlan).

4. When the Section rule form appears, add benefit plan name, description and employee cost. Add controls for the properties noted in the scenario.

    a. For the benefit plan name, set control to **Dropdown**. Do not be concerned about sourcing the dropdown list; we will do that in a later exercise.

    b. For the description, on the properties presentation tab, set Edit options to **Read only (always)**.

    c. For the employee cost, on the properties general tab, set control to **Currency**. On the Presentation Tab, set Edit options to **Read only (always)** and Text alignment to **Left**.



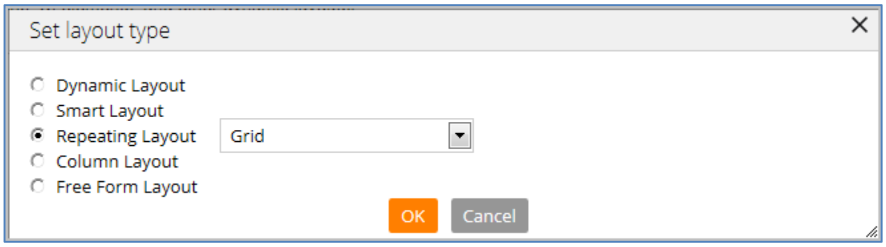
Note: This section will be used to allow an employee to select healthcare benefits within the sections created for selecting Medical, Dental and Vision plans. To filter the list of plans that are offered to match the plan type, a parameter will be added to the section during a later phase of the development process.

Section used for adding dependents

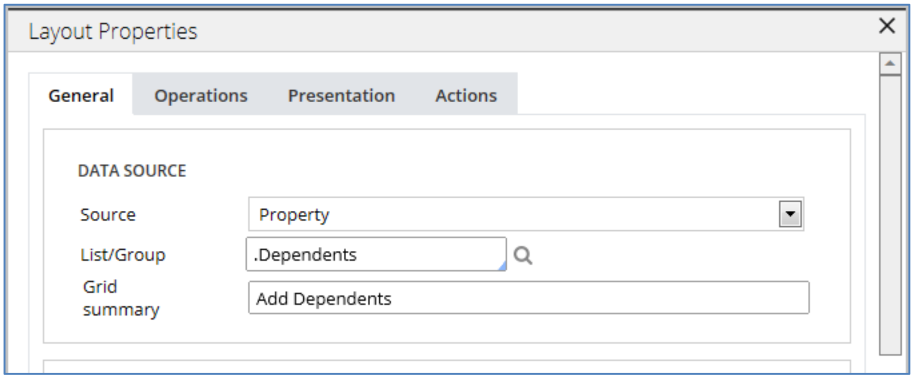
5. Create a new UI section used to display Dependent information

    - Enter **Dependents Information** in the Label field.
    - The Record Context should apply to the class for the employee data object (SAE-HRServices-Data-Employee).

6. When the Section rule form appears, add a new layout under the default section and set the layout type to Repeating Layout: Grid



7. On the general tab of the layout properties for the section, use the page list property you created for collecting dependent information as the source for the repeating grid.



Layout Properties

General | Operations | Presentation | Actions

DATA SOURCE

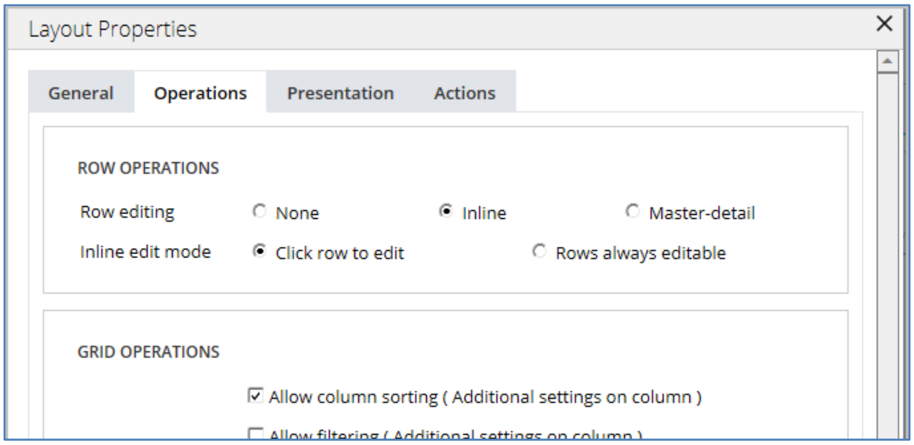
Source: Property

List/Group: .Dependents

Grid summary: Add Dependents

Note: If you cannot select .Dependents from the auto-complete list, ensure you created this section within the Employee class and not Dependent.

8. On the Operations tab of the layout properties for the section, set the inline edit mode.



Layout Properties

General | Operations | Presentation | Actions

ROW OPERATIONS

Row editing: ○ None ● Inline ○ Master-detail

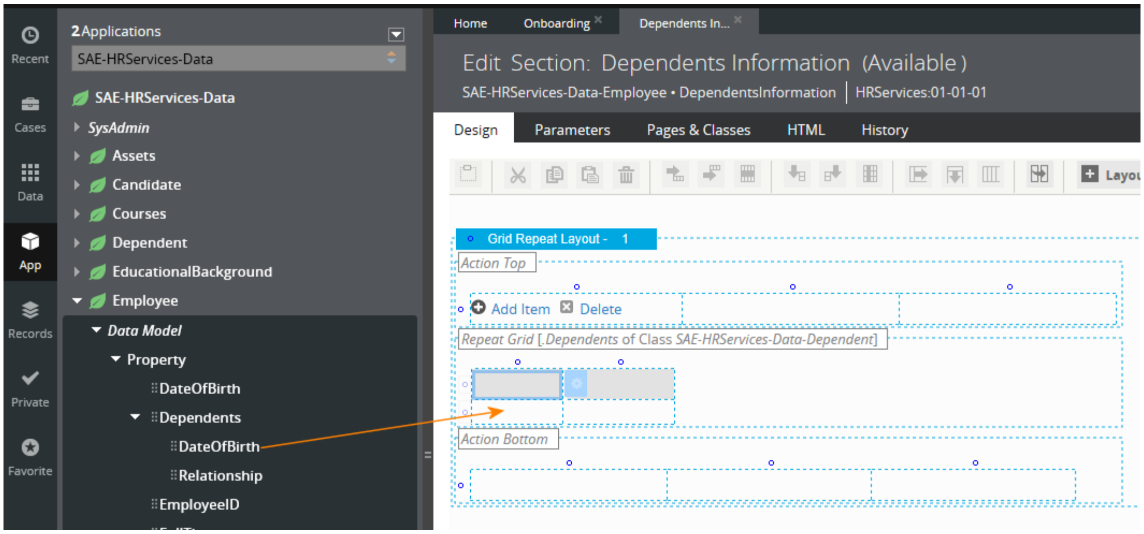
Inline edit mode: ● Click row to edit ○ Rows always editable

GRID OPERATIONS

☑ Allow column sorting (Additional settings on column)

☐ Allow filtering (Additional settings on column)

9. Add the necessary properties to the grid as noted in the scenario.

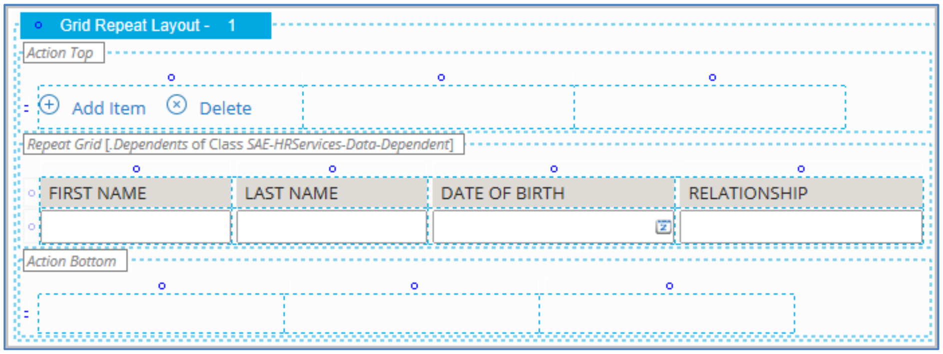


Edit Section: Dependents Information (Available)

SAE-HRServices-Data-Employee • DependentsInformation | HRServices:01-01-01

Repeat Grid [.Dependents of Class SAE-HRServices-Data-Dependent]

10. Use the **Insert Column After** tool to add columns to the repeating grid.



11. When you are done creating the new layout, you can delete the default section.

Exercise: Building Screens for Implementing the Assignment

Scenario

Our task now is to build the UI components necessary for implementing the assignments for creating a new employee record, selecting healthcare benefit plans, and adding dependents in each of the appropriate case types.

Remember, we are taking a very purposeful manual approach to creating and implementing the UIs so you can see how the individual UI components fit together.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Our approach would be to add a section to the Onboarding case type for creating the employee record. This section would include the section we created in the class that contains the employee data object type.

Next, we would add three sections to the Benefits Enrollment case type: one for selecting a medical plan, one for selecting a dental plan and one for selecting a vision plan. This section would include the base section created earlier in the class used to define the HR plan's data object type.

Finally, we would add a section to the Benefits Enrollment case type for adding dependents. This section would include the section for dependents information we created in the class used to define the employee data object type.

Then, we would create the necessary flow actions in the appropriate case types and reference these flow actions on the appropriate steps in the two case types.

This exercise assumes the case type you created to manage the employee onboarding process is called **Onboarding** and the case type you created to manage the benefits enrollment process is called **Benefits Enrollment**.

Hints

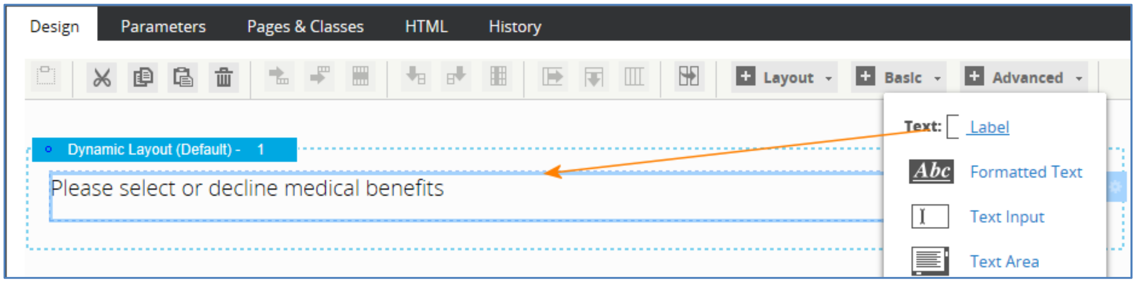
1. It is helpful, before performing this exercise, to review the names you gave to the UI sections created previously. Open the Application Explorer to view the SAE-HRServices-Data class.

    a. Expand **Employee > User Interface > Section** and note the section names. In our example, the sections are EmployeeInformation and DependentsInformation.

    b. Expand **HRPlan > User Interface > Section** and note the section name. In our example, it is HealthcareBenefit.

2. After you have noted the previously created UI section, change context in the Application Explorer to view your application's work class. Right-click the case type name to begin creating the associated section and flow action.

3. You can add a label to a section by selecting **Text Label** from the Basic menu. Change text and formatting style using the Presentation tab in the Cell Properties dialog.



Please select or decline medical benefits

Procedure

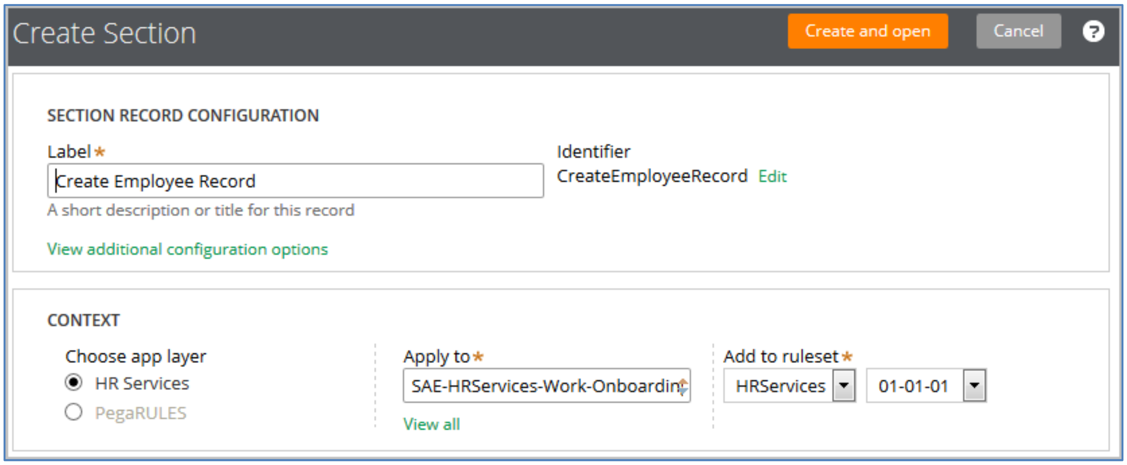
Begin this exercise by opening the Application Explorer to view the SAE-HRServices-Work class tree. Then, follow the steps below:

UI components for creating the employee record

1. Create a UI section in the Onboarding case type to enter information for a new employee.

    a. From Application Explorer, **SAE-HRServices-Work** class, right-click **Onboarding** and select **+Create > User Interface > Section**.

    b. Enter **Create Employee Record** in the Label field and ensure **SAE-HRServices-Work-Onboarding** is selected in the record context Apply to field before creating the section.


Create Section

SECTION RECORD CONFIGURATION

Label * Create Employee Record

A short description or title for this record

Identifier CreateEmployeeRecord Edit

View additional configuration options

CONTEXT

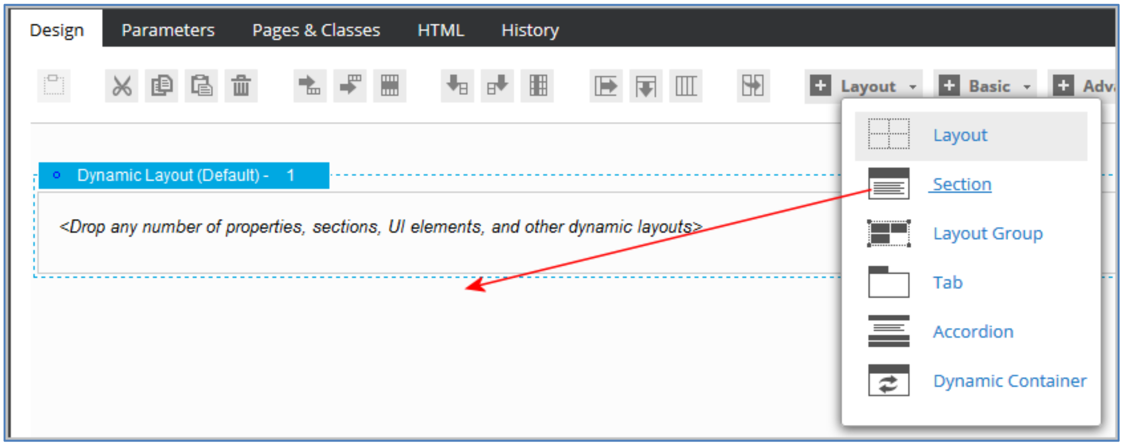
Choose app layer
● HR Services
○ PegaRULES

Apply to * SAE-HRServices-Work-Onboardin

View all

Add to ruleset * HRServices 01-01-01

2. Add a new section below the default dynamic layout.

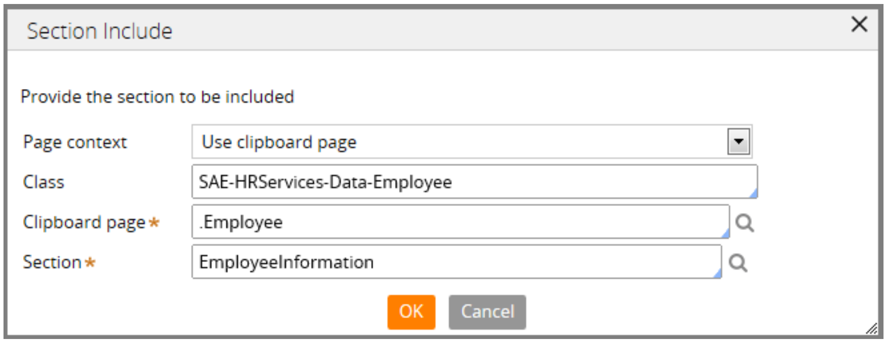


3. Use a clipboard page as the page context. Change the class to the name of the class you used to define the employee data object type.

4. Set the clipboard page to the name of the page you used for the employee data object. This page should appear in the smart prompt list.

5. Finally, select the section you created to display the employee information. This section should be available in the smart prompt list.

The following provides our view of steps 3 to 5:



Section Include

Provide the section to be included

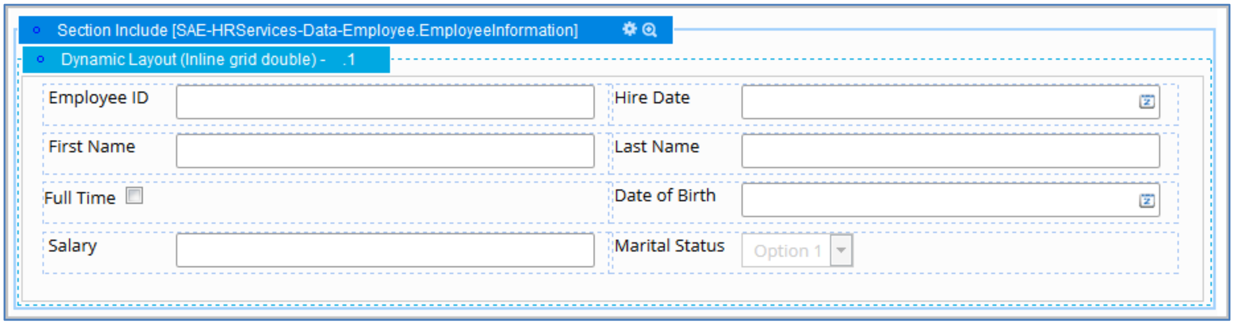
Page context: Use clipboard page

Class: SAE-HRServices-Data-Employee

Clipboard page*: .Employee

Section*: EmployeeInformation

6. Delete the default dynamic layout. Note that your section format may differ.



Section Include [SAE-HRServices-Data-Employee.EmployeeInformation]

Dynamic Layout (Inline grid double) - .1

Employee ID

Hire Date

First Name

Last Name

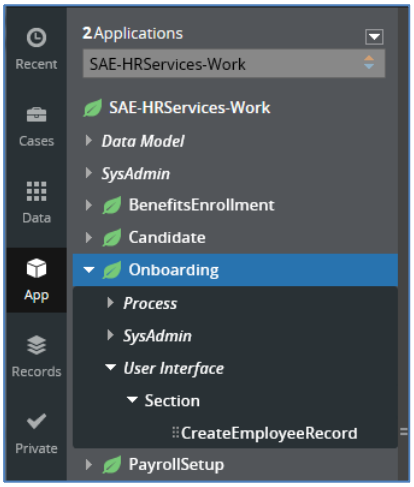
Full Time □

Date of Birth

Salary

Marital Status: Option 1

7. Save the new section and then refresh the Application Explorer *Org-App*-Work class and view the Onboarding tree. Notice there are Process, SysAdmin and User Interface branches, and that the UI branch has a Section branch that contains the CreateEmployeeRecord section we just created.

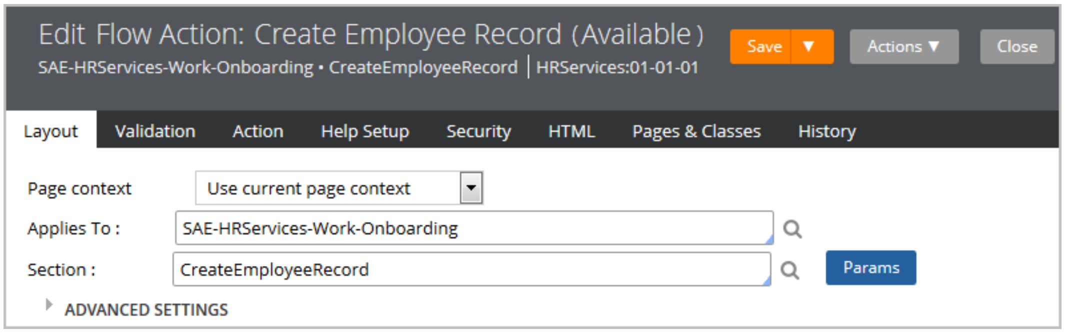


8. We now have a UI section that will determine how employee information is displayed. Create a flow action in the Onboarding case type that will use this section.

   a. From Application Explorer, **SAE-HRServices-Work** class, right-click **Onboarding** and select **+Create > User Interface > Flow Action**.

b. Enter **Create Employee Record** in the Label field and ensure **SAE-HRServices-Work-Onboarding** is selected in the record context Appy to field before creating the flow action.

9. Reference the section you created in the previous steps and then save your changes.



Edit Flow Action: Create Employee Record (Available)

SAE-HRServices-Work-Onboarding • CreateEmployeeRecord | HRServices:01-01-01

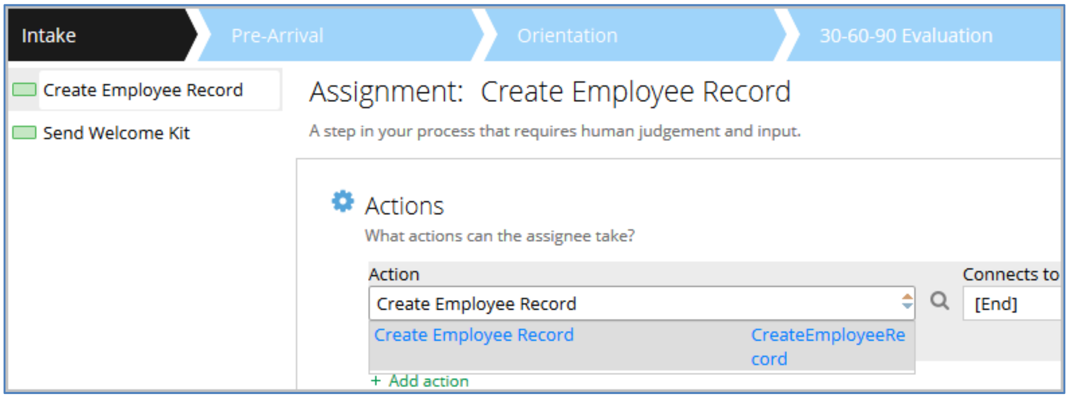
Page context: Use current page context

Applies To : SAE-HRServices-Work-Onboarding

Section : CreateEmployeeRecord

10. From the Case Designer, configure process details in the stage that contains the step to create an employee record.

11. Select the assignment step and then select the flow action you created in step 8 from the auto-complete Action field menu. Save your changes.



Assignment: Create Employee Record

A step in your process that requires human judgement and input.

Actions

What actions can the assignee take?

Action	Connects to
Create Employee Record	[End]

Create Employee Record   CreateEmployeeRecord

Note: If you click the Run button to test your Create Employee Record step in the Onboarding case type, but don't see the UI section appear, ensure that you've done the *Adding Default Data Transforms* exercise and that the Employee page properties have been correctly initialized for the Onboarding process.

UI components for selecting healthcare benefits

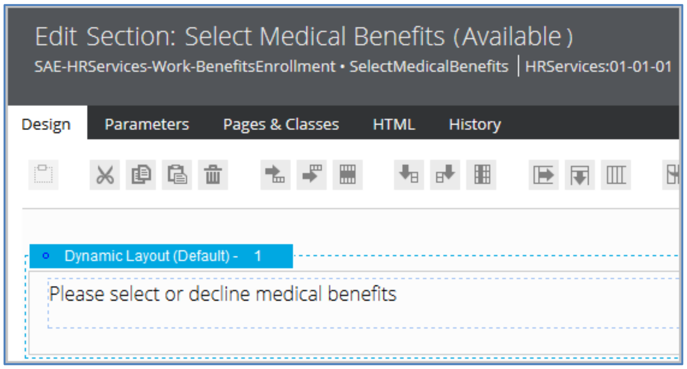
1. Create a UI section in the Benefits Enrollment case type for selecting a medical plan.

   a. From Application Explorer, **SAE-HRServices-Work** class, right-click **BenefitsEnrollment** and select **+Create > User Interface > Section**.

   b. Enter **Select Medical Benefits** in the Label field and ensure **SAE-HRServices-Work-BenefitsEnrollment** is selected in the record context Apply to field.

   c. Create the record and when the section rule form appears, add a read-only label control in the default dynamic layout with the text: "Please select or decline medical benefits." Select a presentation format that will make the text more prominent.

d. **Save** the new section. Do not add any properties. We will complete the section includes later in the development process, when the context of what we need to do will be more clear.



Edit Section: Select Medical Benefits (Available)

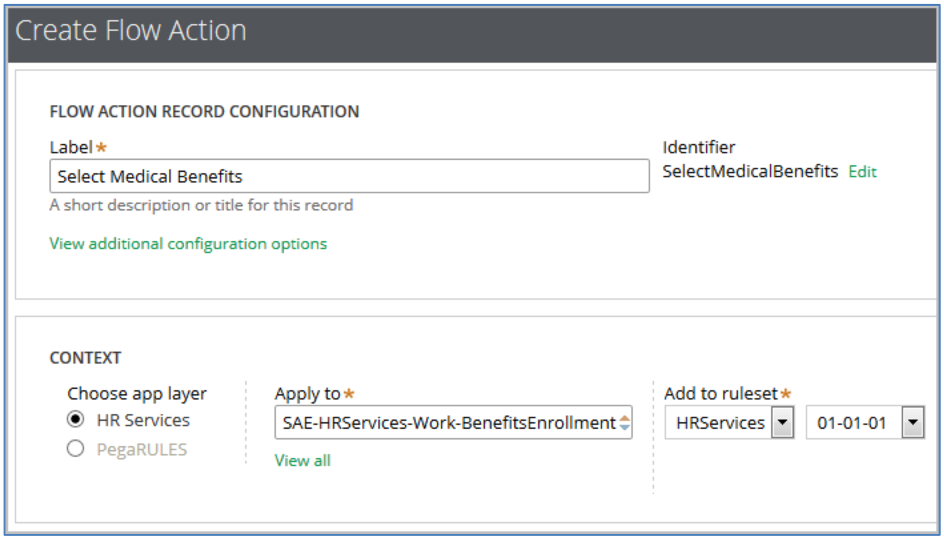
SAE-HRServices-Work-BenefitsEnrollment • SelectMedicalBenefits | HRServices:01-01-01

Please select or decline medical benefits

2. Create a flow action in the Benefits Enrollment case type that will use the SelectMedicalBenefits section

   a. From Application Explorer, SAE-HRServices-Work class, right-click **BenefitsEnrollment** and select **+Create > User Interface > Flow Action**.

   b. Enter **Select Medical Benefits** in the Label field, ensure **SAE-HRServices-Work-BenefitsEnrollment** is selected in the record context Apply to field and then create the record.



Create Flow Action

FLOW ACTION RECORD CONFIGURATION

Label* Select Medical Benefits

A short description or title for this record

Identifier SelectMedicalBenefits Edit

View additional configuration options

CONTEXT

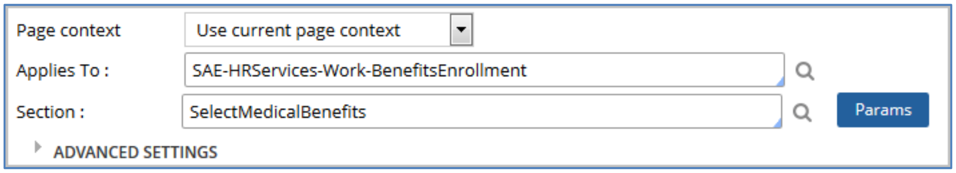
Choose app layer
● HR Services
○ PegaRULES

Apply to* SAE-HRServices-Work-BenefitsEnrollment

View all

Add to ruleset* HRServices 01-01-01

   c. Reference the section you created in the previous step (if the section name doesn't appear in the auto-complete menu, ensure you saved it after creating in the previous step).

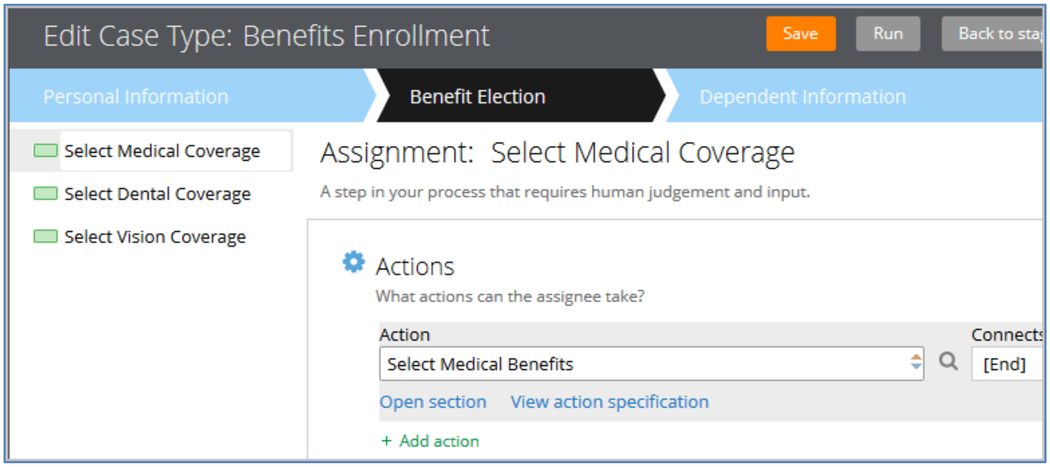


Page context: Use current page context

Applies To : SAE-HRServices-Work-BenefitsEnrollment

Section : SelectMedicalBenefits

3. In the Benefits Enrollment case type, configure the process details for the step you are using for selecting medical benefits and add the (flow) action to the appropriate place.



Edit Case Type: Benefits Enrollment

Assignment: Select Medical Coverage

A step in your process that requires human judgement and input.

Actions

What actions can the assignee take?

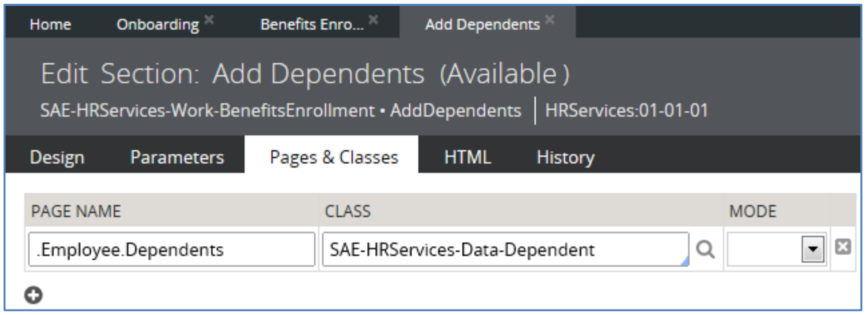
Action	Connects
Select Medical Benefits	[End]

4. Repeat this procedure two more times – once each for creating the case type level UI components for selecting dental benefits and again for vision benefits. Note that you could also open the Medical Benefits section; use the **Save As** button and then change label text.

UI components for adding dependents

Creating the necessary UI components (section and flow action) for adding dependents will be similar to the steps you took to create UI components for creating the employee record.

1. Create a UI section in the Benefits Enrollment case type for adding dependents.
    a. From Application Explorer, **SAE-HRServices-Work** class, right-click **BenefitsEnrollment** and select **+Create > User Interface > Section**.
    b. Enter **Add Dependents** in the Label field, ensure **SAE-HRServices-BenefitsEnrollment** is selected in the record context Apply to field and then create the record.
2. In the section rule form, open the Pages & Classes tab and make the Dependents class available to this section.

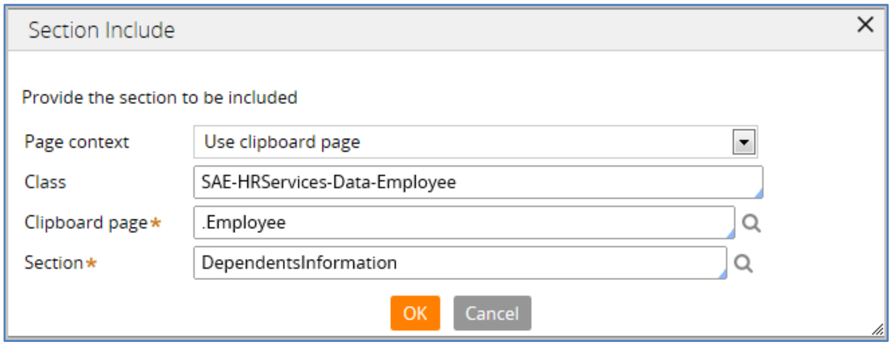


Edit Section: Add Dependents (Available)

SAE-HRServices-Work-BenefitsEnrollment • AddDependents | HRServices:01-01-01

PAGE NAME	CLASS	MODE
.Employee.Dependents	SAE-HRServices-Data-Dependent	

3. Add a new section below the default dynamic layout and give it the following attributes.



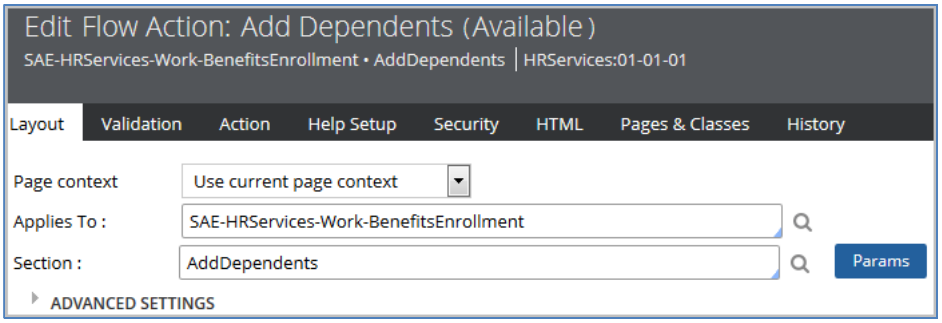
Section Include

Provide the section to be included

Page context	Use clipboard page
Class	SAE-HRServices-Data-Employee
Clipboard page*	.Employee
Section*	DependentsInformation

Note: You can delete the default dynamic layout section after this section is included and then save your updates.

4. Create a flow action in the Benefits Enrollment case type
    a. From the Application Explorer, SAE-HRServices-Work class, right-click **BenefitsEnrollment** and select **+Create > User Interface > Flow Action**.
    b. Enter **Add Dependents** in the Label field, ensure **SAE-HRServices-Work-BenefitsEnrollment** is selected in the record context Apply to field and then create the new flow action record.
    c. Refer the new flow action to this new Add Dependents section and then save your updates.



Edit Flow Action: Add Dependents (Available)

SAE-HRServices-Work-BenefitsEnrollment • AddDependents | HRServices:01-01-01

Page context	Use current page context
Applies To :	SAE-HRServices-Work-BenefitsEnrollment
Section :	AddDependents

5. Configure the process details for the step you are using for adding dependents in the Benefits Enrollment case type. Enter this new Add Dependents flow action name in the properties panel of the process outline for the appropriate step in the Dependent Information stage.

6. **Note:** If you click the Run button to test your Add Dependents step in the Benefits Enrollment case type, but don't see the UI section appear, skip to the *Adding Default Data Transforms* exercise to ensure that the Employee page properties have been correctly initialized for the Benefits Enrollment process.

Exercise: Working with Modal Dialogs

Scenario

As a new employee begins to select benefits, they need to be able to verify their employee information.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Configure a section that contains the employee's personal data such as their date of birth, email address, and marital status. Then, create an associated flow action.

Configure the section used for selecting benefits with a dynamic layout. Add a button control and configure it with a click event that calls a local action with a target of Modal Dialog. The local action should be the flow action noted above.

Hints

1. The Button control is a basic control.
2. To add actions to the button, open the properties panel for the button control and select the Actions tab.
3. Add a Click event to the Actions tab of the button control.

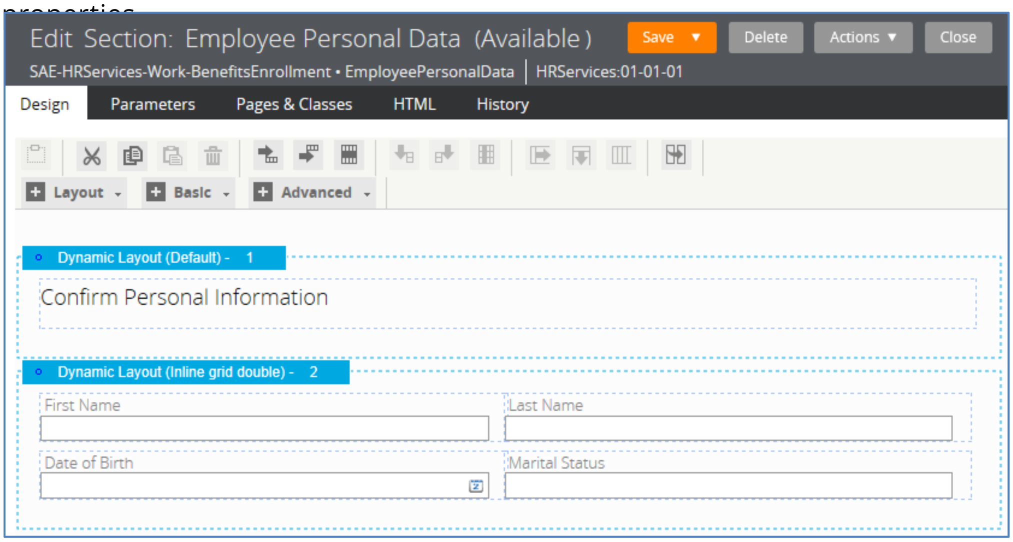
Procedure

Follow the steps below:

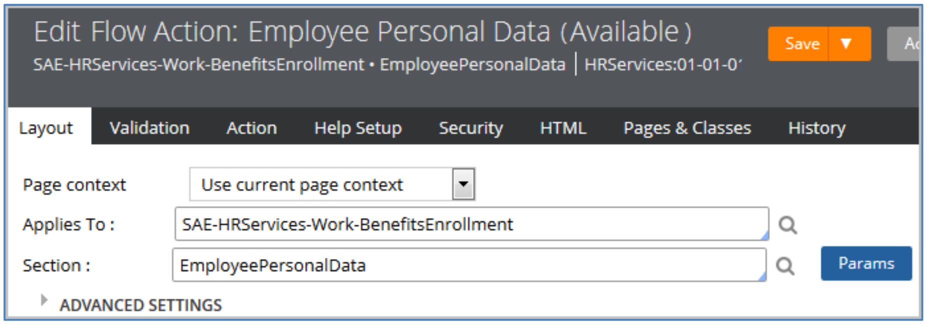
1. Create a UI section in the Benefits Enrollment case type to validate personal information for a new employee.
    a. Create the UI section in the SAE-HRServices-Work-BenefitsEnrollment context.
    b. Enter **Employee Personal Data** in the Label field.

> **Reminder**: To add a last name field, add a Text Input field from the Basic menu. In the Cell Properties dialog, select .Employee.pyLastName.

2. To keep this simple, we suggest adding only the first and last name; however, you can add other properties as desired. Save the section after you have completed adding properties.


Edit Section: Employee Personal Data (Available)
Save
Delete
Actions
Close
SAE-HRServices-Work-BenefitsEnrollment • EmployeePersonalData | HRServices:01-01-01
Design
Parameters
Pages & Classes
HTML
History
Layout
Basic
Advanced
Dynamic Layout (Default) - 1
Confirm Personal Information
Dynamic Layout (Inline grid double) - 2
First Name
Last Name
Date of Birth
Marital Status

3. Create a flow action in the same record context as the section and then reference the section created above in the flow action.


Edit Flow Action: Employee Personal Data (Available)
Save
SAE-HRServices-Work-BenefitsEnrollment • EmployeePersonalData | HRServices:01-01-0
Layout
Validation
Action
Help Setup
Security
HTML
Pages & Classes
History
Params
ADVANCED SETTINGS

Page context: Use current page context

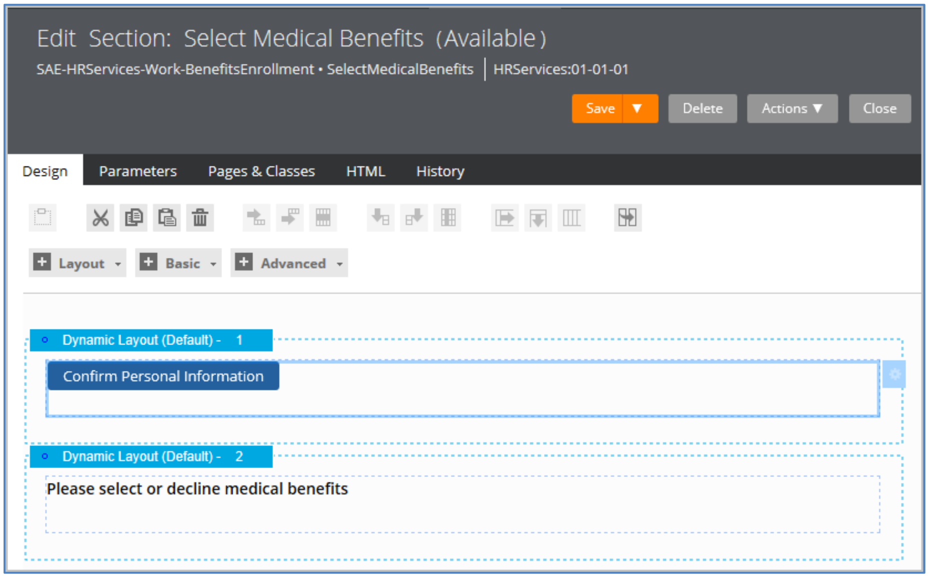
Applies To : SAE-HRServices-Work-BenefitsEnrollment

Section : EmployeePersonalData

4. Open the process outline for the Personal Information stage of the Benefits Enrollment case type. Add the Employee Personal Data flow action to the Confirm Personal Information step.

Create a Modal Dialog Button

1. Add a separate dynamic layout to the section used for selecting medical coverage and add a button to the layout.



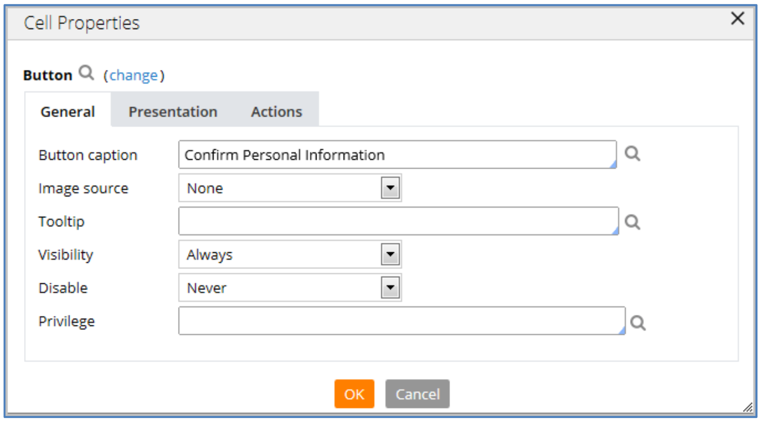
Edit Section: Select Medical Benefits (Available)

SAE-HRServices-Work-BenefitsEnrollment • SelectMedicalBenefits | HRServices:01-01-01

Save
Delete
Actions
Close
Design
Parameters
Pages & Classes
HTML
History
Layout
Basic
Advanced
Dynamic Layout (Default) - 1
Confirm Personal Information
Dynamic Layout (Default) - 2

Please select or decline medical benefits

2. Set the button caption to “Confirm Personal Information.”



Cell Properties

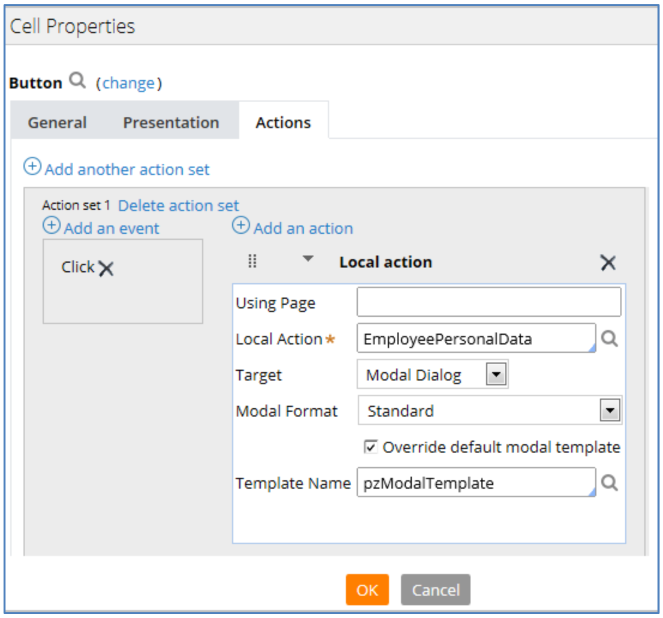
Button (change)

Button caption	Confirm Personal Information
Image source	None
Tooltip	
Visibility	Always
Disable	Never
Privilege	

3. Add a Click event to the button control that launches a local action.

4. Reference the flow action that contains the section with the employee’s personal data.

5. Change the Target to Modal Dialog.



Using Page:

Local Action*: EmployeePersonalData

Target: Modal Dialog

Modal Format: Standard

☑ Override default modal template

Template Name: pzModalTemplate

Exercise: Managing Data for Selectable List Controls

Scenario

During the onboarding phase, an employee should be able to select their choice of medical, dental, and vision healthcare plans. The lists of available plans should be sourced from the sae2_hr_plan database table.

Note to participants: When you complete this exercise, it may seem as if nothing is happening – and that's OK. The goal of this lesson and exercise is to help you understand how to "manage data" for selectable list controls. In a later phase of development, we will force changes to the UI based on events.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Our approach is to use a single list of benefit plans as the source, but filter this list based on the type of plan for each selectable list control. This list of plans would be maintained in a data page that is sourced from a report definition.

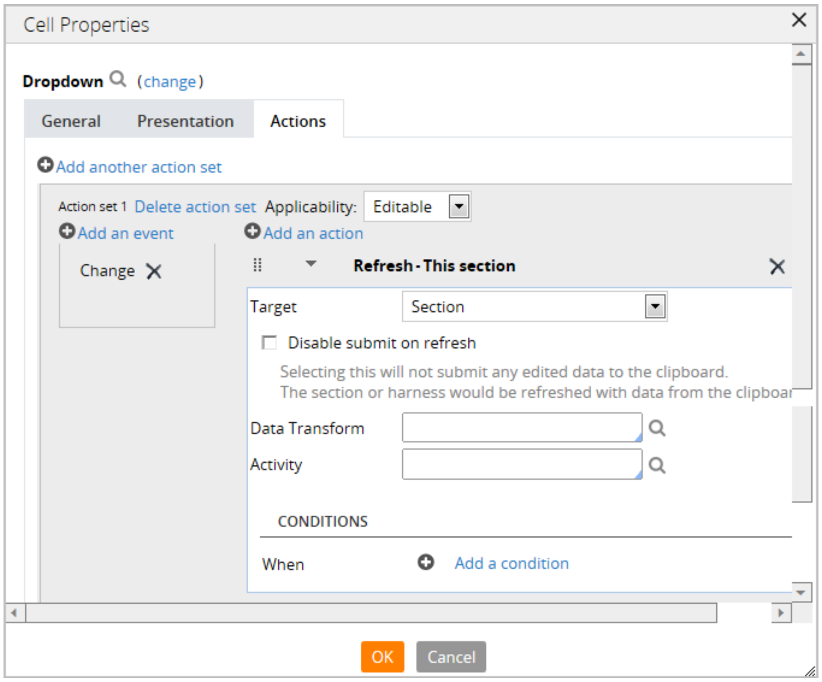
To present the list of available healthcare plans to the employee, we would use the single section we created earlier in the HR Plans data class, and then include this base section in a section located in the case type used for benefits enrollment. We would use this base section three times, one for each type of available healthcare plan.

Finally, we would store each selected plan type in a separate page property in the SAE-HRServices-Work class: a DentalPlan, a MedicalPlan and a VisionPlan

This approach requires the use of parameters. Our approach would be to filter on the type of plan such as "Medical," "Dental," and "Vision." The type of plan is an available column in the HR Plans database table.

Hints

1. To create a change event that will refresh a section, click the **Actions** tab in the Cell Properties dialog, add a **Change** Event to cause the action **Refresh - This Section**.



Dropdown (change)

Action set 1 Applicability: Editable

Change

Refresh - This section

Target: Section

☐ Disable submit on refresh

Selecting this will not submit any edited data to the clipboard.
The section or harness would be refreshed with data from the clipboar

Data Transform:

Activity:

CONDITIONS

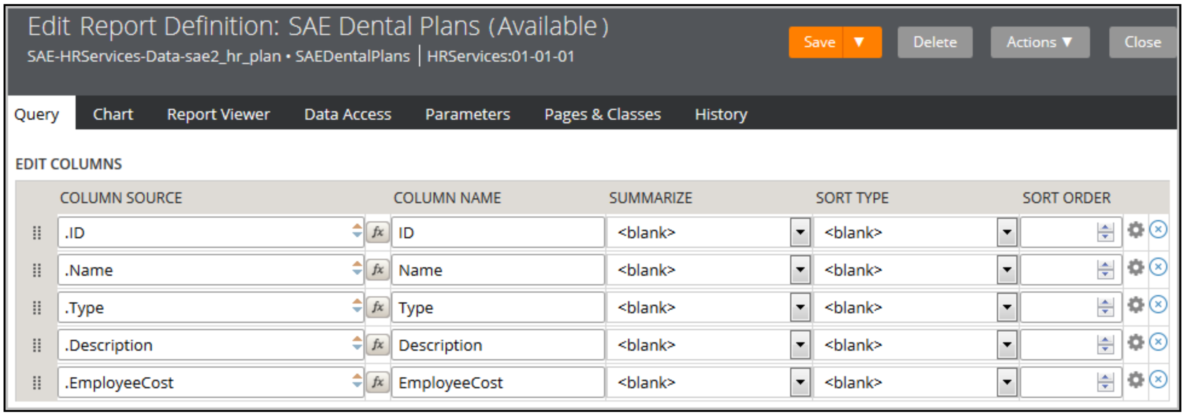
When

Procedure

Create a Report Definition to List Dental Plans

Practice filtering data by creating a report that will show only Dental plans.

1. Create a report definition in the data class used for the healthcare plans.
    a. From the App Explorer, SAE-HRServices-Data class, right-click **HRPlan** and select **+Create → Reports → Report Definition**.
    b. Enter **SAE Dental Plans** in the Label field, ensure SAE-HRServices-Data-HRPlan is selected in the record context Apply to field and then create the new report definition.
2. The report definition should retrieve all columns (id, name, type, description and employee cost) that are in the HR Plan data table.



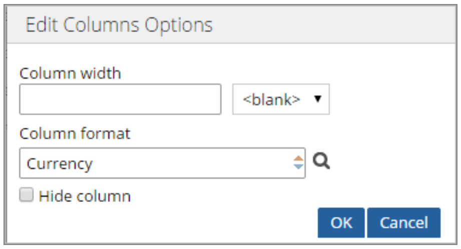
Edit Report Definition: SAE Dental Plans (Available)

SAE-HRServices-Data-sae2_hr_plan • SAEDentalPlans | HRServices:01-01-01

EDIT COLUMNS

COLUMN SOURCE	COLUMN NAME	SUMMARIZE	SORT TYPE	SORT ORDER
.ID	ID	<blank>	<blank>	
.Name	Name	<blank>	<blank>	
.Type	Type	<blank>	<blank>	
.Description	Description	<blank>	<blank>	
.EmployeeCost	EmployeeCost	<blank>	<blank>	

3. Click the gear icon to the right of the Employee Cost column and change the column format to currency.



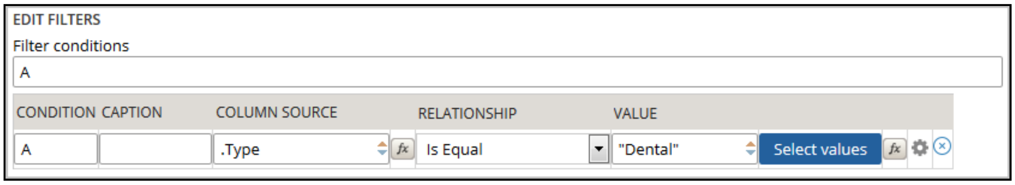
Edit Columns Options

Column width: <blank>

Column format: Currency

☐ Hide column

4. We want to display only dental plans, so add a filter to check that the plan .Type is “Dental”.



EDIT FILTERS

Filter conditions: A

CONDITION	CAPTION	COLUMN SOURCE	RELATIONSHIP	VALUE
A		.Type	Is Equal	"Dental"

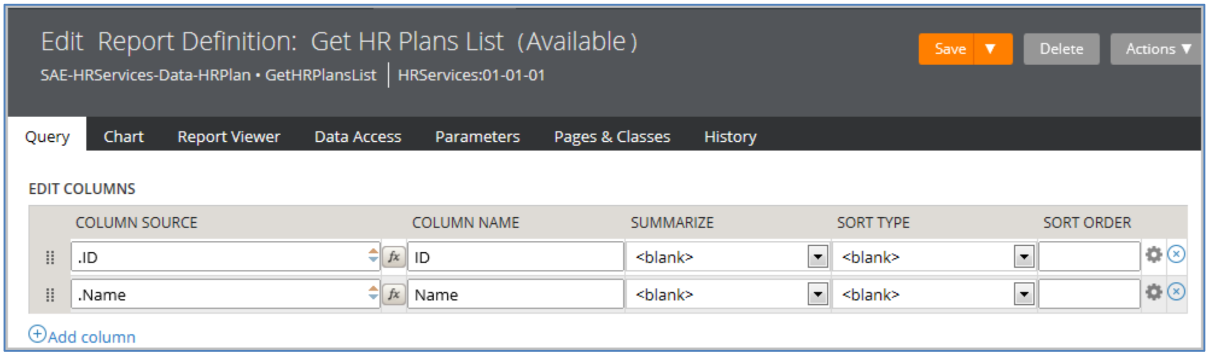
5. **Save** your changes to the report definition and then select **Run** from the Actions menu to test that you see only dental plans.

Edit the Report to use a Parameter to Filter Data

Using the Dental Plan report as a starting point, add a parameter to filter by Plan Type

1. Save your dental plan report definition with a different name.
   a. Select **Save As** from the Save menu.
   b. Enter **Get HR Plans List** in the Label field, ensure SAE-HRServices-Data-HRPlan is selected in the record context Apply to field and then create and open your copy.

2. The report only needs columns that can be used to create a list of plans for the selected type (id and name). Remove the other fields.



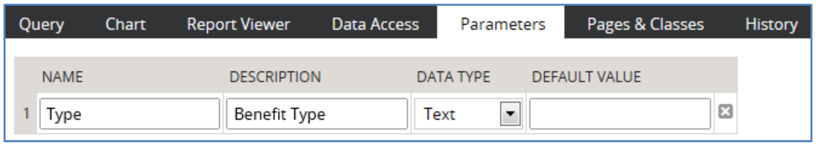
Edit Report Definition: Get HR Plans List (Available)

SAE-HRServices-Data-HRPlan • GetHRPlansList | HRServices:01-01-01

EDIT COLUMNS

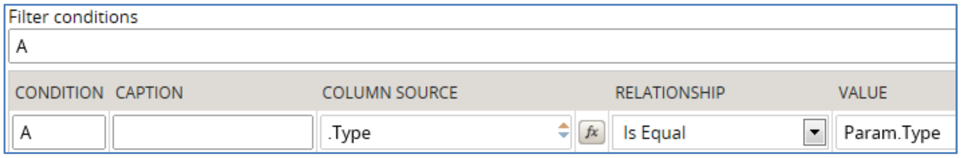
COLUMN SOURCE	COLUMN NAME	SUMMARIZE	SORT TYPE	SORT ORDER
.ID	ID	<blank>	<blank>	
.Name	Name	<blank>	<blank>	

3. To see results for a single plan type, add a parameter to filter the list. On the parameters tab of the report definition, add a parameter. Use Type as the name and Benefit Type as the description. Save updates to the Report Definition.



	NAME	DESCRIPTION	DATA TYPE	DEFAULT VALUE
1	Type	Benefit Type	Text	

4. Return to the Query tab and add a filter where the column source .*Type* is equal to the value of Param.<paramName> ; where <paramName> is the name of the parameter you added on the Parameters tab.



Filter conditions

A

CONDITION	CAPTION	COLUMN SOURCE	RELATIONSHIP	VALUE
A		.Type	Is Equal	Param.Type

5. **Save** your changes to the report definition.

Setup the data page to store the list of plans

1. Create a list structure data page in the data class used for the healthcare plans that will contain the results of the report definition that filters data plans by type.

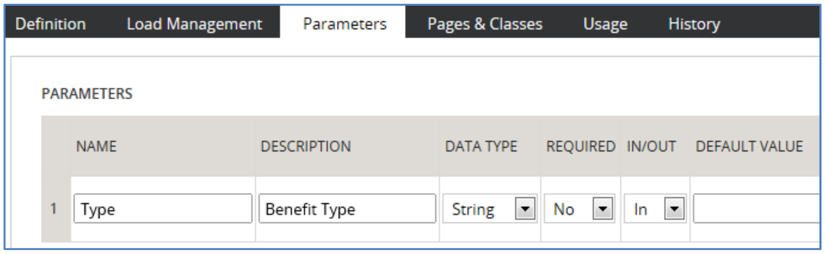
   a. From the App Explorer, SAE-HRServices-Data class, right-click **HRPlan** and select **+Create → Data Model → Data Page**.

   b. Enter **HR Plan List** in the Label field, ensure SAE-HRServices-Data-HRPlan is selected in the record context Apply to field and then create the new data page.

2. On the Data Page rule form Definition tab, set the page Structure to **List** and the Object type to the name of the class used to define the HR plans data object type.

3. Set the Source for this data page to the report definition created earlier.

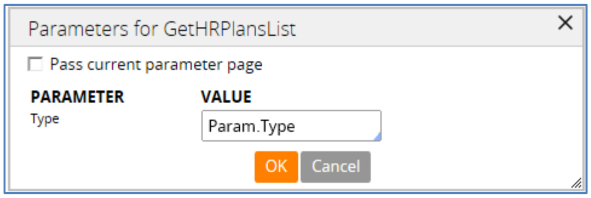
4. On the Parameter tab, add a parameter to this data page just as you did for the report definition and then save your changes.



PARAMETERS

	NAME	DESCRIPTION	DATA TYPE	REQUIRED	IN/OUT	DEFAULT VALUE
1	Type	Benefit Type	String	No	In	

6. Return to the Definition tab, click the **Parameters** link (in the Data Sources frame) and set the Parameter value to Param.<paramName> .



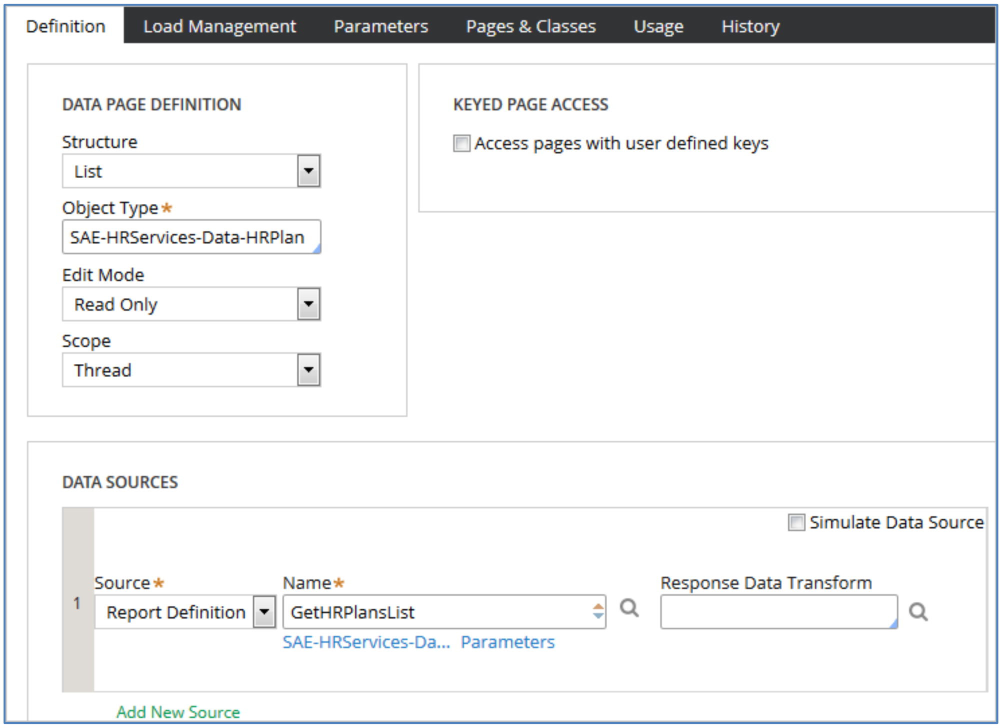
Parameters for GetHRPlansList

□ Pass current parameter page

PARAMETER	VALUE
Type	Param.Type

Note: The name of the parameter should be consistent between this data page and the report definition.

Check your progress using this sample



DATA PAGE DEFINITION

Structure: List

Object Type*: SAE-HRServices-Data-HRPlan

Edit Mode: Read Only

Scope: Thread

KEYED PAGE ACCESS

□ Access pages with user defined keys

DATA SOURCES

□ Simulate Data Source

1 Source*: Report Definition Name*: GetHRPlansList Response Data Transform:

SAE-HRServices-Da... Parameters

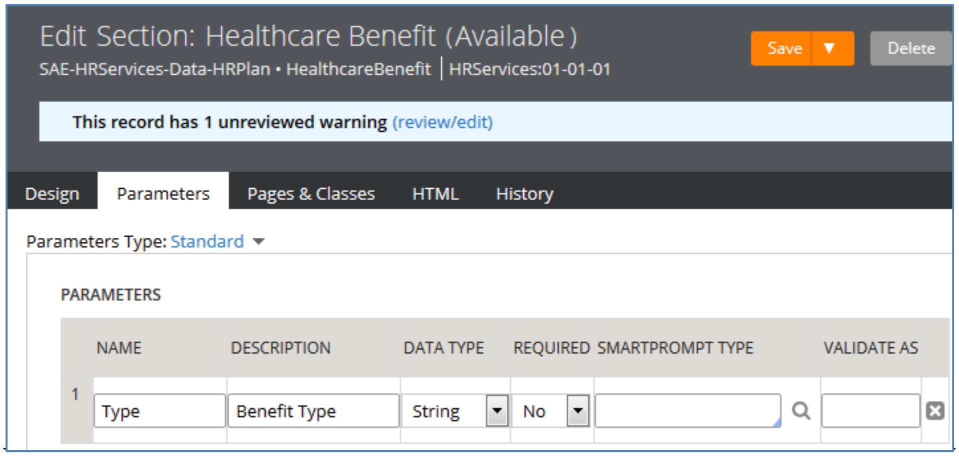
Add New Source

7. Save updates to the Data Page rule form.

Setup the reusable section to display the list of available plans

In an earlier exercise, we performed the initial steps to create a section that can be used to display healthcare benefit plans. At that time it was noted that in a later phase of development, we would add a parameter to the section that will allow us to filter healthcare benefit plans by type. We now have the structure in place to add this parameter.

1. Find and open the section you created to display healthcare benefit plans. It should be in the data class used for the healthcare plans.
    a. From the App Explorer, SAE-HRServices-Data class, expand **HRPlan → User Interface → Section** to view HRPlan UI sections.
    b. Open the section created to display healthcare benefits.
2. On the Parameters tab of the Healthcare Benefit section, add a parameter of name “Type” with the description “Benefit Type” to filter the type of plan (Medical, Vision, Dental) in the Benefit list.



Edit Section: Healthcare Benefit (Available)

SAE-HRServices-Data-HRPlan • HealthcareBenefit | HRServices:01-01-01

This record has 1 unreviewed warning (review/edit)

Parameters Type: Standard

PARAMETERS

	NAME	DESCRIPTION	DATA TYPE	REQUIRED	SMARTPROMPT TYPE	VALIDATE AS
1	Type	Benefit Type	String	No		

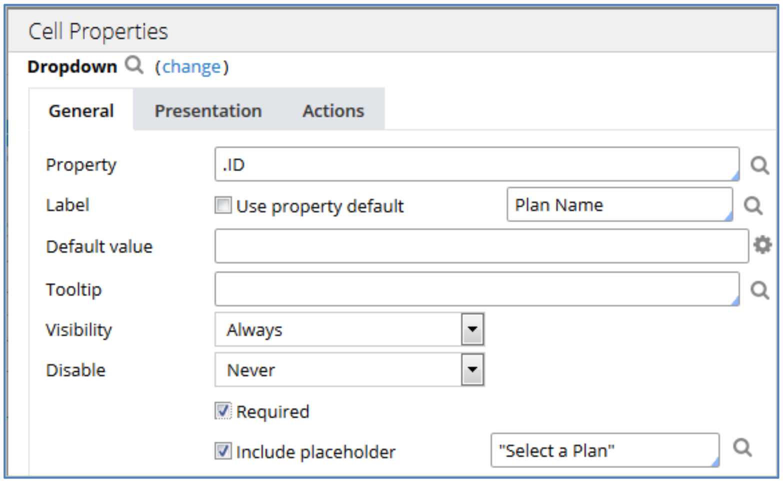
3. Return to the Design tab and configure the source of the Dropdown control.

4. The dropdown cell will display plan names for the selected benefit type (Medical, Dental or Vision). It's possible when the section was initially created, the "name" property was associated with this cell. If that is the case, it should be changed to ID.

   c. Since the cell will be populated from the HR Plan database table, which has a class key of ID, the property for the cell should be **ID** and the display property should be **Name**.

   d. In a later phase of development, this key-value pair will be used to access description and employee cost for a selected plan.

   e. Add a default placeholder (to display before a plan is selected) of "Select a Plan".

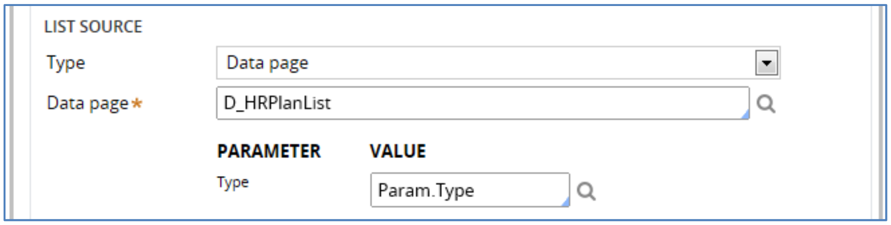


Cell Properties

Dropdown (change)

Property	.ID
Label	□ Use property default Plan Name
Default value	
Tooltip	
Visibility	Always
Disable	Never
	☑ Required
	☑ Include placeholder "Select a Plan"

5. In the middle of the Section rule form, set the parameter value to the name of the parameter you added to the data page.

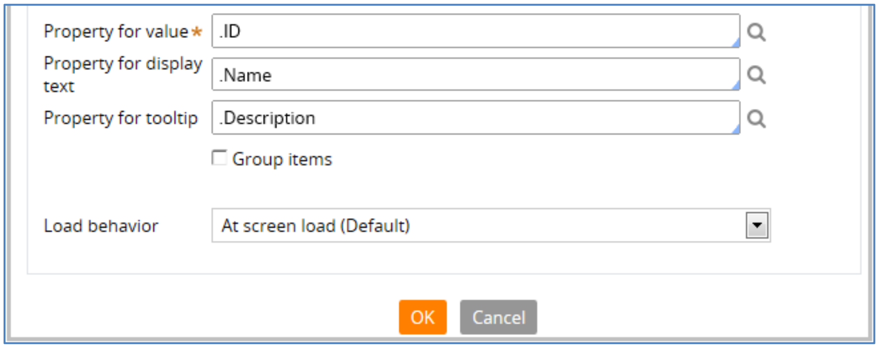


LIST SOURCE

Type	Data page
Data page*	D_HRPlanList

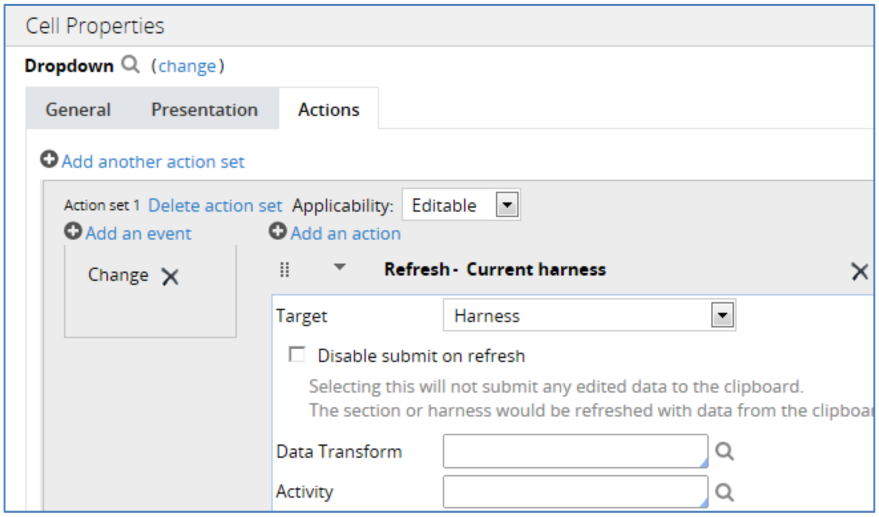
PARAMETER	VALUE
Type	Param.Type

6. If not completed, at the bottom of the Section rule form, use .ID for the property value, .Name as the property for the display text and .Description as the property for the tooltip.



Property for value*	.ID
Property for display text	.Name
Property for tooltip	.Description
	□ Group items
Load behavior	At screen load (Default)

7. While still in the dropdown cell properties dialog, use the Actions tab to add a change event that refreshes the harness.



Cell Properties

Dropdown (change)

Action set 1 Applicability: Editable

Change

Refresh - Current harness

Target	Harness

□ Disable submit on refresh

Selecting this will not submit any edited data to the clipboard.
The section or harness would be refreshed with data from the clipboa

Data Transform	
Activity	

Note: The preferred action would be to refresh the current section when a plan is selected. However, given the current PRPC design, a section refresh will result in loss of your plan-type parameter and the dropdown list will no longer be filtered. Therefore, we recommend refreshing the entire Harness.

Also note: We will be unable to test the Healthcare Benefit section yet, as it is not configured for stand-alone use. It will be included into the individual sections for selecting medical, dental and vision plans, allowing us to test the plan filtering results at that time.

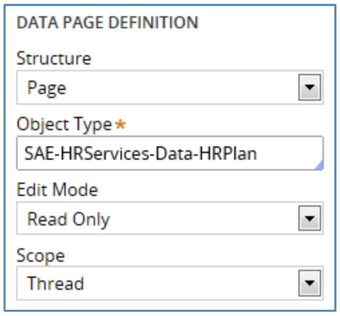
Setup the data page used to retrieve a selected benefit plan

As noted in the Approach section, all benefit plans are contained within a single database table. The parameter that we have added to our Healthcare Benefit section will allow us to display only plan options for the selected plan type (i.e. all benefit plans for type “Medical”) for employees to choose from. Once an employee selects a plan, we want to display the Description and Cost. To retrieve this information, we need to create a lookup page that will use plan ID to retrieve the rest of the benefit plan properties.

1. Create a page structure data page in the data class used for the healthcare plans.

   f. From the App Explorer, SAE-HRServices-Data class, right-click **HRPlan** and select **+Create → Data Model → Data Page**. Enter **HR Plan Lookup** in the Label field and then create the new data page.

2. Set the object type to the name of the class used to define the HR plans data object type.



DATA PAGE DEFINITION

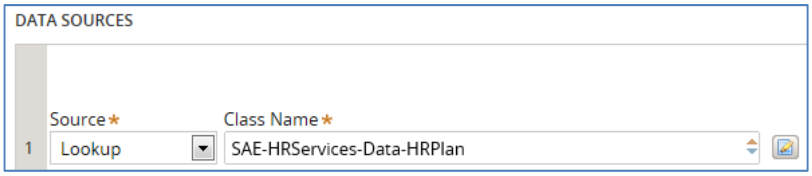
Structure: Page

Object Type*: SAE-HRServices-Data-HRPlan

Edit Mode: Read Only

Scope: Thread

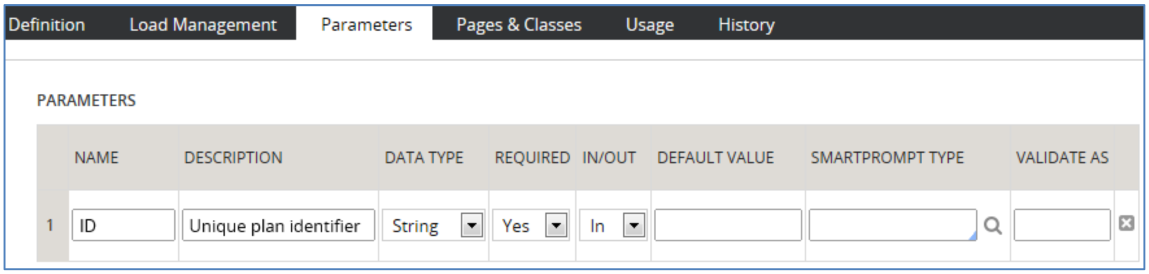
3. Source this data page using a lookup and set the class name to the name of the class used to define the HR plans data object type.



DATA SOURCES

	Source*	Class Name*
1	Lookup	SAE-HRServices-Data-HRPlan

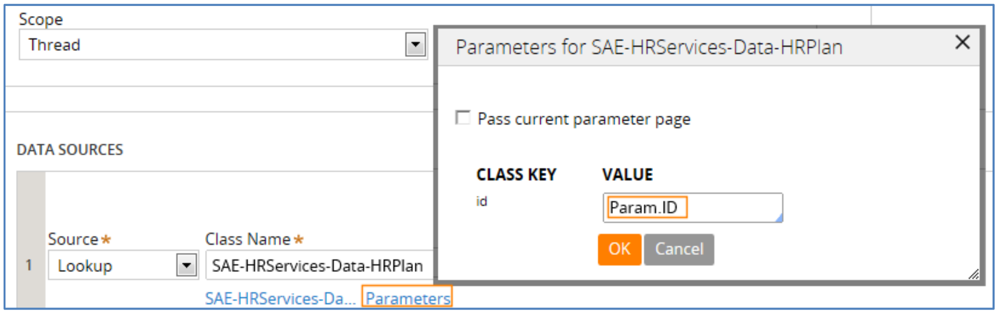
4. On the Parameters tab, add a parameter that will allow you to retrieve a record by the class key (the primary key in the table). Description should be “Unique Plan ID” and required should be “Yes”.



PARAMETERS

	NAME	DESCRIPTION	DATA TYPE	REQUIRED	IN/OUT	DEFAULT VALUE	SMARTPROMPT TYPE	VALIDATE AS
1	ID	Unique plan identifier	String	Yes	In			

5. Return to the Definition tab and set the Parameters value to param.<paramName> .



Scope

Thread

DATA SOURCES

Source* Class Name*

1 Lookup SAE-HRServices-Data-HRPlan

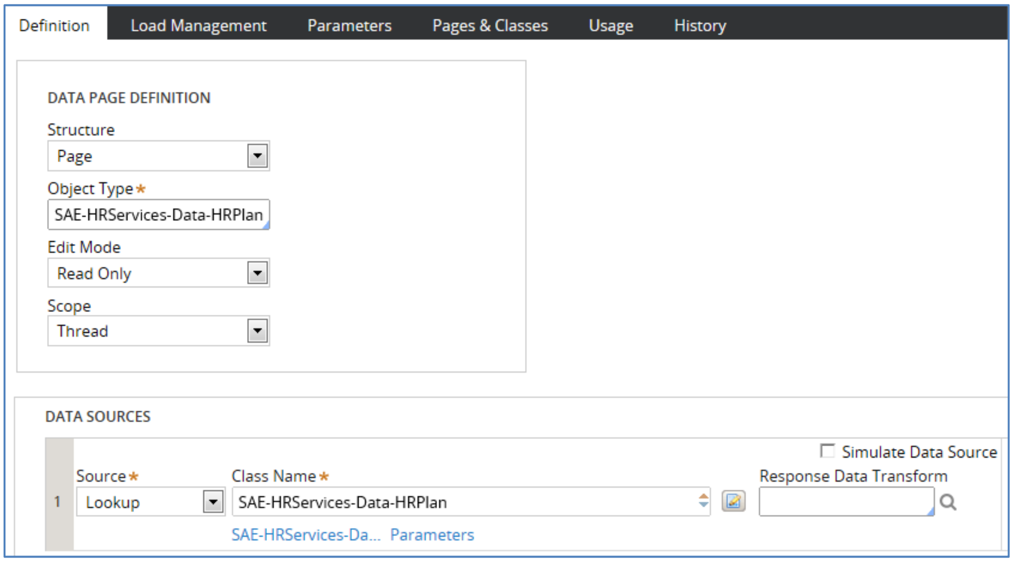
SAE-HRServices-Da... Parameters

Parameters for SAE-HRServices-Data-HRPlan

□ Pass current parameter page

CLASS KEY	VALUE
id	Param.ID

Check your progress using this sample



DATA PAGE DEFINITION

Structure: Page

Object Type*: SAE-HRServices-Data-HRPlan

Edit Mode: Read Only

Scope: Thread

DATA SOURCES

□ Simulate Data Source

	Source*	Class Name*	Response Data Transform
1	Lookup	SAE-HRServices-Data-HRPlan SAE-HRServices-Da... Parameters	

8. Remember to save updates to the data page rule form.

Setup the properties to store the selected plans.

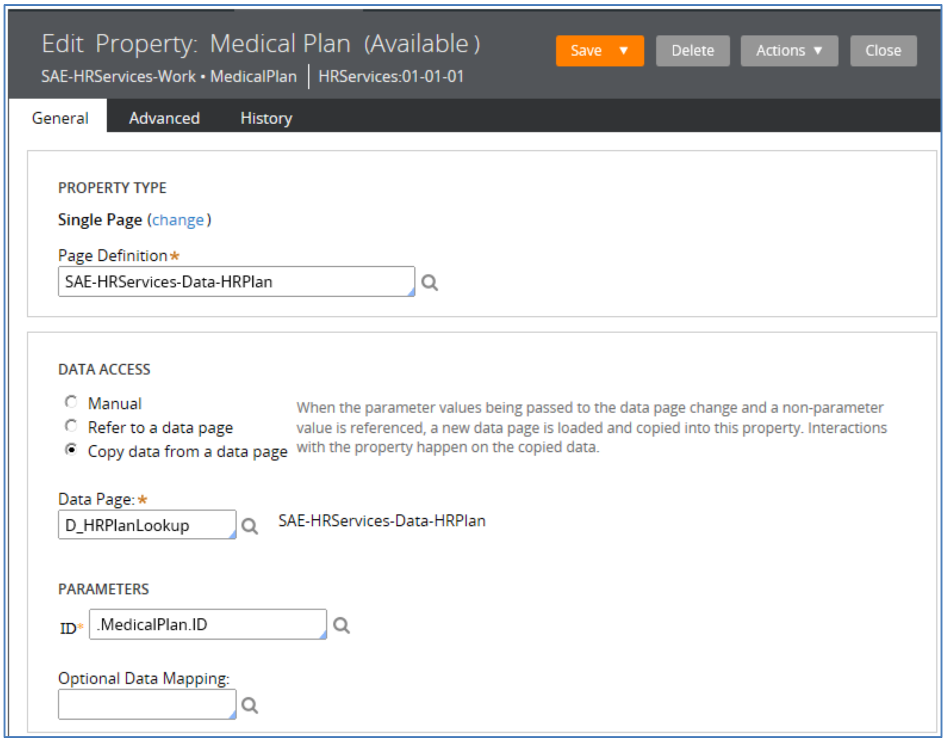
We now need to create page properties to store the results of each of the plan types (Medical, Dental and Vision) that the employee selects.

1. Create page type properties for each of the available plans (medical, dental and vision) in the application's work class.
   a. From the App Explorer, SAE-HRServices-Work class, right-click SAE-HRServices-Work and select **+Create → Data Model → Property**.
   b. Enter **Medical Plan** in the Label field and then create the new property (you will also do this for **Dental Plan** and **Vision Plan).**
   c. Use SAE-HRServices-Work as the record context for all three properties.

9. Set the property type to **Single Page** and the page definition to the data class you created for the HR plans.

10. Set the data access to **Copy data from a data page**. We want this page property to hold information for the selected plan, so specify the data access source as the Data Page used as the lookup created previously.

11. The selected value should be associated with the ID field, so set the value in the parameters field to .<PagePropertyName>.ID (where the PagePropertyName is the name of the page you are creating).



Edit Property: Medical Plan (Available)

SAE-HRServices-Work • MedicalPlan | HRServices:01-01-01

PROPERTY TYPE

Single Page (change)

Page Definition*: SAE-HRServices-Data-HRPlan

DATA ACCESS

○ Manual
○ Refer to a data page
● Copy data from a data page

When the parameter values being passed to the data page change and a non-parameter value is referenced, a new data page is loaded and copied into this property. Interactions with the property happen on the copied data.

Data Page:* D_HRPlanLookup SAE-HRServices-Data-HRPlan

PARAMETERS

ID* .MedicalPlan.ID

Optional Data Mapping:

12. Save your changes and ensure you have performed steps 1-4 for all three plan types (Medical, Dental, and Vision). This is a good opportunity to use the “Save As” option in the Save menu.

Setup the sections that will include the benefits enrollment section

We now have a reusable section that includes a dropdown menu that should only show plan options for the plan type that is selected. We also have a plan lookup that will return a Description and Cost for a selected plan. In this step, we’ll put all of the pieces together to create a section that includes a header and plan selections for each of the benefit types. Begin by viewing the Benefits Enrollments UI sections that have already been created.

1. From the Application Explorer SAE-HRServices-Work class, expand **Benefits Enrollment > User Interface > Section**.

2. After viewing the current list of Benefits Enrollment UI sections, open one of the benefit UI sections (SelectMedicalBenefits, SelectDentalBenefits or SelectVisionBenefits).

3. Using the Layout menu, drag a new section below the default dynamic layout. To include the reusable Healthcare Benefit section that was created in the data class that contains the HR plans, give this section the properties:
   - Page context: **Use clipboard page**
   - Class: **SAE-HRServices-Data-HRPlan**
   - Clipboard page: **.MedicalPlan** (or .DentalPlan or .VisionPlan)
   - Section: **HealthcareBenefit**
   - Parameter: Type /Value: **"Medical"** (or "Dental" or "Vision" – Use quotes)

4. Save section changes and ensure you have performed these steps for all plan types.

5. Check the SAE-HRServices-Data-HRPlan HealthcareBenefit section to ensure that the Employee Cost field uses a Currency control, and the presentation is Read Only, left-aligned.

6. Test that the correct plans are being displayed by creating an instance of the case type.

After saving all of your updates, test you work by running the process. When a new case is created, confirm that each dropdown displays only those plans for the specific area of coverage.

Exercise: Creating Dynamic UIs

Scenario

New employees have the option of waiving healthcare coverage plans. When the benefits selection page first displays, the healthcare coverage options should be presented with an option to waive coverage for each type plan. If the new employee elects to waive any specific healthcare coverage, the options for selecting that particular plan should not display.

When a healthcare plan is selected in any given category, the rate and description for the selected plan should display.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Identify Boolean-type properties used to register the new employee's choice to waive each healthcare plan (medical, dental, and vision).

Add a cell to the benefit selection that would allow the employee to decline a benefit type. For each check box control, add an action to refresh the section on change. Then, for each section include, add an expression to the visibility condition so the section is visible when the appropriate property for waiving coverage is not equal to true.

Finally, add a change event to the dropdown control in the base section – the one we created in …Data-<name of health plan class> - so that the section is refreshed when a plan option is selected.

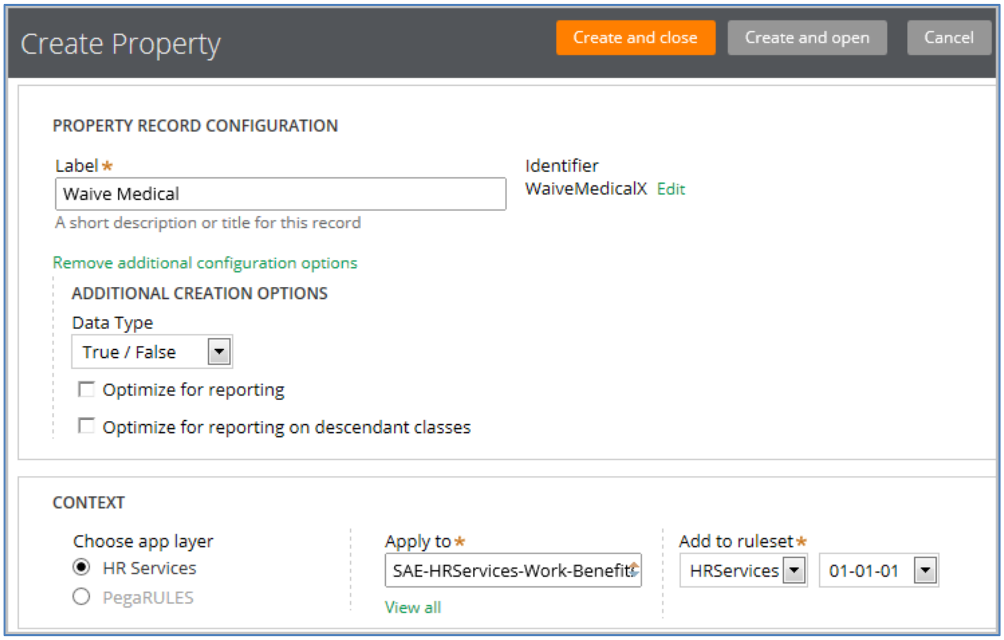
Procedure

Begin this exercise by opening the Application Explorer to view the SAE-RHServices-Work class tree.

Configure the properties and layouts for waiving benefit options

1. Create three Boolean (true/false) properties to track an employee's choice to decline a type of benefit. Name these properties WaiveMedical, WaiveDental and WaiveVision. They should apply to the class SAE-HRServices-Work-<benefits enrollment case type>
    a. From the App Explorer, SAE-HRServices-Work class, right-click **BenefitsEnrollment** and select **+Create → Data Model → Property**.
    b. Enter **Waive Medical** in the Label field and then click **View additional configuration options**.

c. Set the Data Type to **True / False** and then create the new property record.



PROPERTY RECORD CONFIGURATION

Label* Waive Medical
A short description or title for this record

Identifier
WaiveMedicalX Edit

Remove additional configuration options

ADDITIONAL CREATION OPTIONS

Data Type True / False

□ Optimize for reporting

□ Optimize for reporting on descendant classes

CONTEXT

Choose app layer
● HR Services
○ PegaRULES

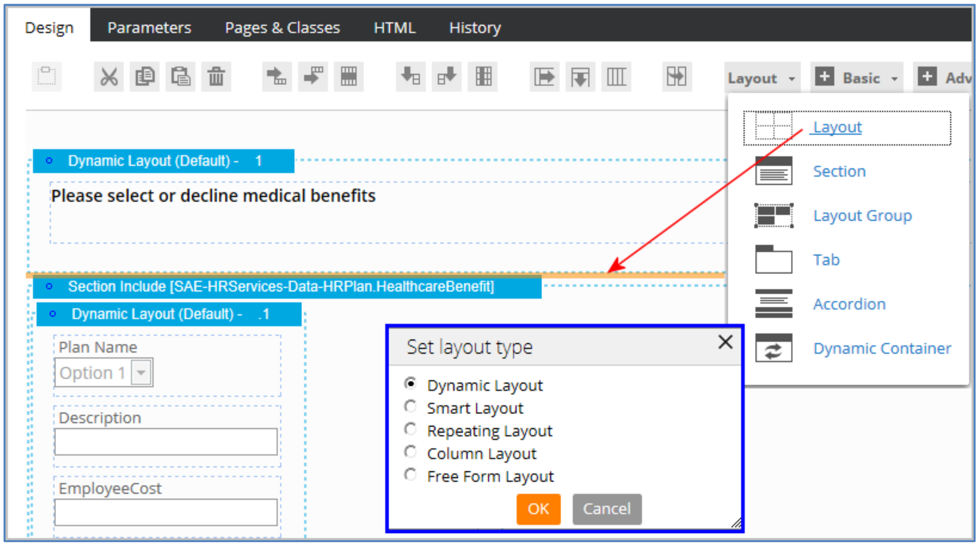
Apply to* SAE-HRServices-Work-Benefits
View all

Add to ruleset* HRServices 01-01-01

2. Repeat the steps above to create properties to waive dental and vision benefits.

3. From the Application Explorer, expand **BenefitsEnrollment > User Interface > Section** to view the sections used for selecting benefits and then open the **SelectMedicalBenefits** section.

4. Add a new Dynamic Layout between the layout that contains the header and the HealthcareBenefit section.



Please select or decline medical benefits

Set layout type

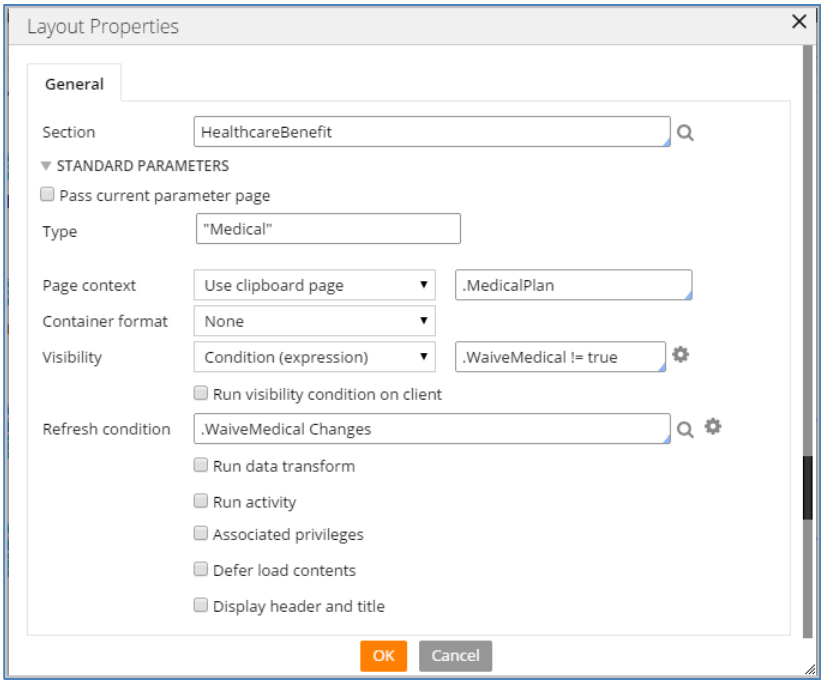
● Dynamic Layout
○ Smart Layout
○ Repeating Layout
○ Column Layout
○ Free Form Layout

5. The new layout will contain the property used by an employee to decline a benefit, with an action to refresh the section when this value changes.

   Add a checkbox to the layout with the following properties:

- Property: **.WaiveMedical**
- Checkbox caption: **Waive medical benefits**
- Caption position: **Left**
- Actions tab – Event: **Change** / Action: **Refresh – This section**

6. In the layout properties for the include of the HealthcareBenefit section, set visibility to occur only when the employee has not waived benefits by adding a visibility condition expression to check if the waive benefit property is not equal to true. Use the gear icon to the right of the Refresh condition to open a dialog that will help create an expression to set refresh to occur every time the waive flag for this benefit changes.



Layout Properties

General

Section: HealthcareBenefit

STANDARD PARAMETERS

□ Pass current parameter page

Type: "Medical"

Page context: Use clipboard page ▾ .MedicalPlan

Container format: None ▾

Visibility: Condition (expression) ▾ .WaiveMedical != true

□ Run visibility condition on client

Refresh condition: .WaiveMedical Changes

□ Run data transform

□ Run activity

□ Associated privileges

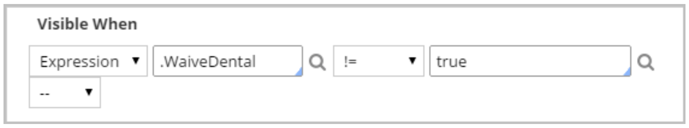
□ Defer load contents

□ Display header and title

Perform the steps 3-6 above for each of the benefit types (Medical, Dental and Vision).

Notes:

Clicking the gear icon to the right of the expression will open a dialog to help form your condition expression.



Visible When

Expression ▾ .WaiveDental != ▾ true

Checking for a waive value to be not equal to true ensures that we are waiting for the employee to explicitly waive benefits by selecting the checkbox before the options will lose visibility. If you choose to check for waive equal to false, you should use the property initialization data transforms to ensure that the initial state of your True / False properties are known.

Test the Benefits Enrollment case type to ensure that the benefit selection doesn’t appear if an employee waives benefits.

Exercise: Validating Properties in a Repeating Layout

Scenario

In this scenario, we need to ensure when a new employee adds a dependent to their healthcare plan, they enter the necessary data for the dependent information such as name, date of birth and relationship.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Our approach would be to use a Validate rule that checks if a dependent was added and, if so, verifies the required dependent information is provided.

Hints

1. Remember that you can set directed inheritance for the Dependents data class to Data-Party; so you can take advantage of the default properties provided by Pega (such as first and last name, and date of birth.)
2. Consider creating the page list in the SAE-HRServices-Work class. Collecting information about dependents could be used for more than benefits enrollment.
3. The section used for adding dependents should be created in the data class used to define the dependents object. Then use a section include in the benefits enrollment case type or the SAE-HRServices-Work class. Regardless of which option you choose to use, be consistent.
4. The Validate rule used to verify the required values for the dependents page list could be created in either the SAE-HRServices-Work class or the benefits enrollment case type. As noted in the previous step, either option would work – your primary goal should be to remain consistent in your implementation.
5. Validate rules are referenced on the Validation tab of the Flow Action form.

Procedure

Follow the steps below:

Setting up the data and UI elements

Use the Application Explorer to find the Dependent data object type and section that were created in the *Implementing a Data Model Using Data Classes* exercise. If you see them, skip ahead to the *Configuring the Validate Rule* part of this exercise.

1. Identify (or create, if necessary) a data class for Dependents. The data elements should be first and last name, date of birth and relationship. Use a local list to define the types of relationships (spouse and child as a minimum). Remember to set directed inheritance to Data-Party so you can take advantage of the default properties such as first and last name and date of birth.

2. Use a page list to implement the data class, if not already implemented.

3. Identify, or create if necessary, a section for adding dependents. Add a repeating grid to this section for adding dependent information.

4. Identify, or create if necessary, a flow action used to implement the section. Reference this flow action at the appropriate place in the process flow used for benefits enrollment.

Require field validation

1. From the Application Explorer data class, locate and open the Dependents Information section and open it.

2. Set the Required flag for the First Name, Relationship and Date of Birth dependent fields.

3. Save the updated section and then run the Benefits Enrollment case type to test your changes.

4. Return to the Dependents Information section and remove the required flags from these fields.

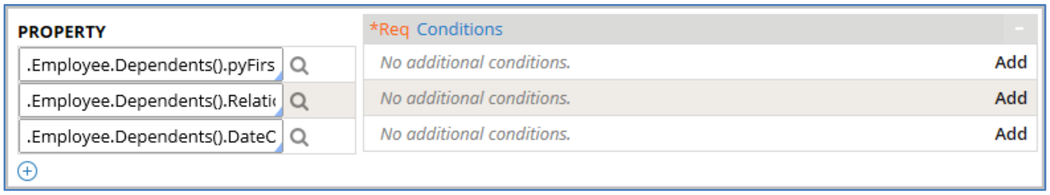
Configuring the Validate rule

1. From the Application Explorer work class, create a Validate rule for the Benefits Enrollment case that will be used to confirm that entries in the dependent page list have values.

    a. Use **Create > Process > Validate**.

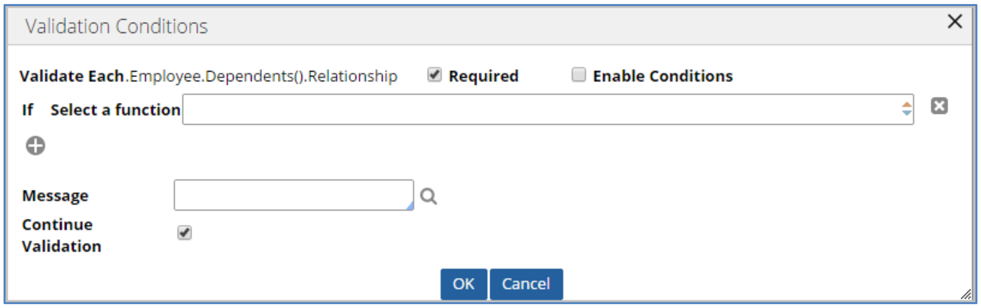
    b. Enter **Require Dependent Values** in the Label field, ensure that SAE-HRServices-Work-BenefitsEnrollment is selected in the record context Apply to field and then create the validation rule.

2. Add a row for each element in the Dependents page list for which you want to verify an entry.



PROPERTY	*Req Conditions	
.Employee.Dependents().pyFirs	No additional conditions.	Add
.Employee.Dependents().Relatio	No additional conditions.	Add
.Employee.Dependents().DateC	No additional conditions.	Add

3. Add a condition to each row to require a value.



Validation Conditions

Validate Each .Employee.Dependents().Relationship ☑ **Required** □ **Enable Conditions**

If **Select a function**

Message

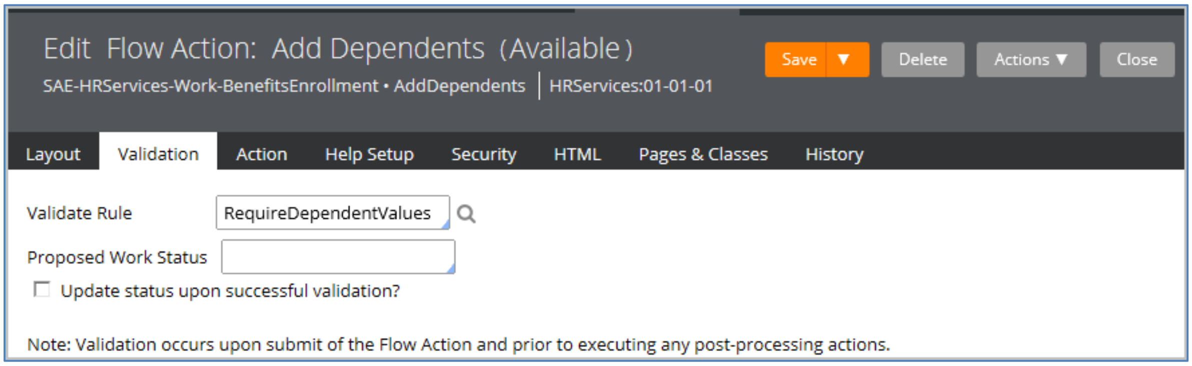
Continue Validation ☑

Note: The message field is used when a validation function is specified. When a property is marked required, and the field is blank at runtime when the form is submitted, the system associates the message "This field may not be blank." with the property. You cannot override this message.

Configuring the Flow Action to call the Validate rule

Add the validate rule to the appropriate flow action.

1. Open flow action rule used to add dependents.
2. Open the Validation tab and add the validation rule you just created.



Edit Flow Action: Add Dependents (Available)

SAE-HRServices-Work-BenefitsEnrollment • AddDependents | HRServices:01-01-01

Validate Rule: RequireDependentValues

Proposed Work Status:

□ Update status upon successful validation?

Note: Validation occurs upon submit of the Flow Action and prior to executing any post-processing actions.

Save changes and test the case type.

Exercise: Creating an Engaging User Experience for Adding Dependents During Benefits Enrollment

Scenario

The user interface is the most broadly visible aspect of your application. Its design affects users' productivity, acceptance, accuracy, and satisfaction and so is a critical factor in implementation success. Organizing UI sections for reusability is a standard application development best practice. In the UI lesson series, we learned best practices for following this standard application development practice using Pega 7.

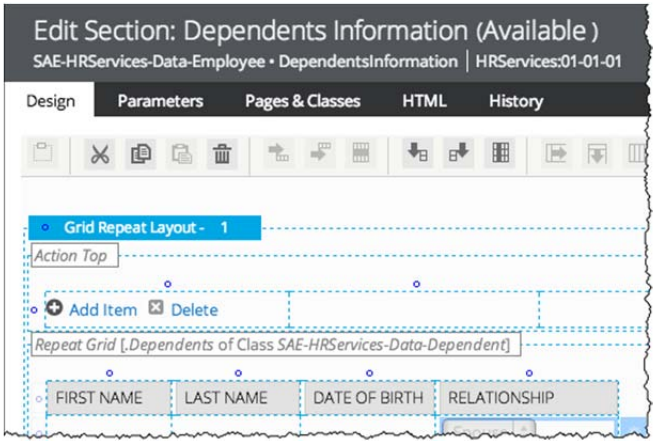
This exercise verification is limited to that part of the onboarding case type concerning the new employee being able to add dependents for healthcare benefits.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

The UI for adding dependents is made up of the following components:

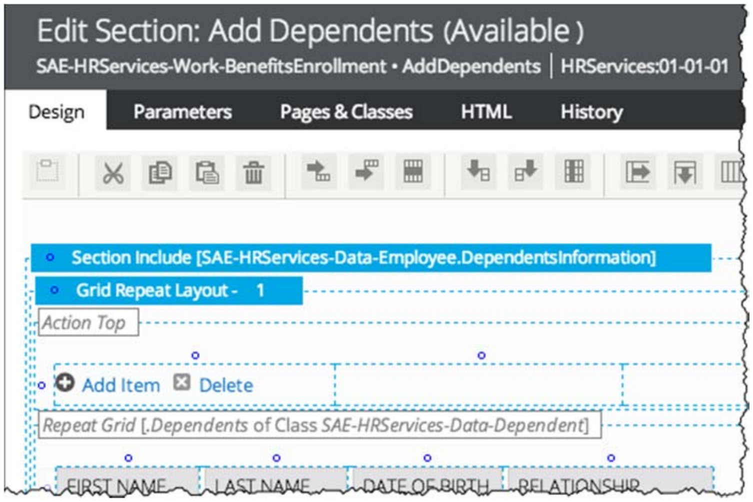
- A section named "Dependents Information" with a record context of SAE-HRServices-Data-Employee.
- We used a Grid Repeat Layout sourced from the Dependents page list property. Then, we added some of the properties noted in the data model.



Edit Section: Dependents Information (Available)

SAE-HRServices-Data-Employee • DependentsInformation | HRServices:01-01-01

- A section named “Add Dependents” with a record context of SAE-HRServices-Work-BenefitsEnrollment. This section includes the Dependents Information section.



Edit Section: Add Dependents (Available)

SAE-HRServices-Work-BenefitsEnrollment • AddDependents | HRServices:01-01-01

Section Include [SAE-HRServices-Data-Employee.DependentsInformation]

Grid Repeat Layout - 1

Action Top

Repeat Grid [.Dependents of Class SAE-HRServices-Data-Dependent]

- A flow action named “Add Dependents” with a record context of SAE-HRServices-Work-BenefitsEnrollment that references the Add Dependents section.
- Finally, we referenced the flow action on the appropriate assignment in the Onboarding case type.

Exercise: Creating an Engaging User Experience for Creating the Employee Record Exercise Verification

Scenario

The user interface is the most broadly visible aspect of your application. Its design affects users' productivity, acceptance, accuracy, and satisfaction and so is a critical factor in implementation success. Organizing UI sections for reusability is a standard application development best practice. In the UI lesson series, we learned best practices for following this standard application development practice using Pega 7.

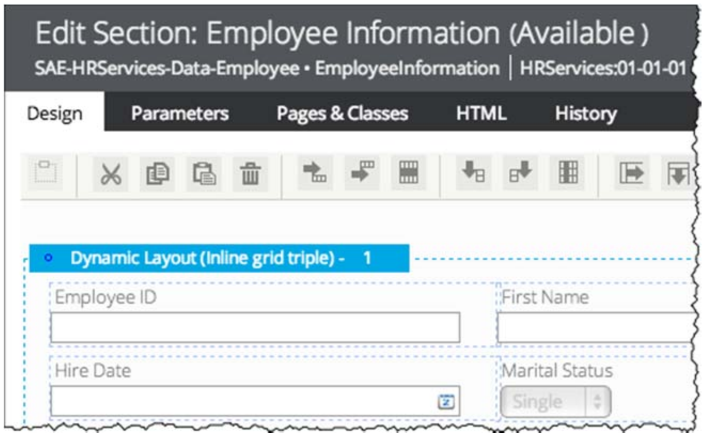
This exercise verification is limited to that part of the onboarding case type concerning creating the employee record.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

The UI for creating the employee record is made up of the following components:

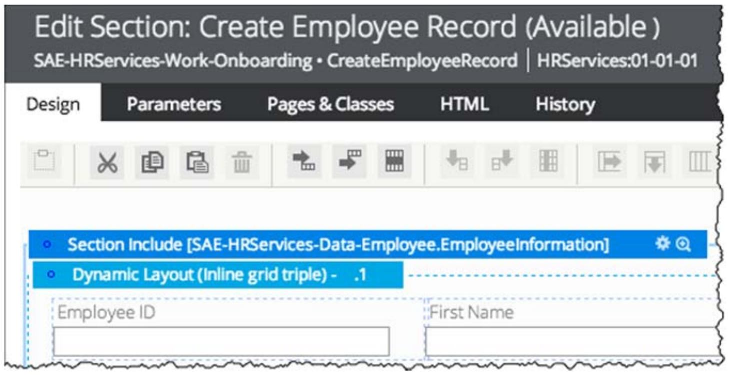
- A section named "Employee Information" with a record context of SAE-HRServices-Data-Employee. We added some of the properties noted in the data model.



Edit Section: Employee Information (Available)

SAE-HRServices-Data-Employee • EmployeeInformation | HRServices:01-01-01

- A section named “Create Employee Record” with a record context of SAE-HRServices-Work-Onboarding. This section includes the Employee Information section. We used a clipboard page for .Employee as the page context.



Edit Section: Create Employee Record (Available)

SAE-HRServices-Work-Onboarding • CreateEmployeeRecord | HRServices:01-01-01

Section Include [SAE-HRServices-Data-Employee.EmployeeInformation]

Dynamic Layout (Inline grid triple) - .1

- A flow action named CreateEmployeeRecord with a record context of SAE-HRServices-Work-Onboarding that references the CreateEmployeeRecord section.
- Finally, we referenced the flow action on the appropriate assignment in the Onboarding case type.

Exercise: Creating an Engaging User Experience for Selecting Benefits Exercise Verification

Scenario

The user interface is the most broadly visible aspect of your application. Its design affects users' productivity, acceptance, accuracy, and satisfaction and so is a critical factor in implementation success. Organizing UI sections for reusability are a standard application development best practice. In the UI lesson series, we learned best practices for following this standard application development practice using Pega 7.

This exercise verification is limited to that part of the onboarding case type concerning the implementation of the benefits enrollment use case.

A new employee should be able to enroll in healthcare benefits for medical, dental and vision coverage – or waive coverage for any, or all of the plans.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

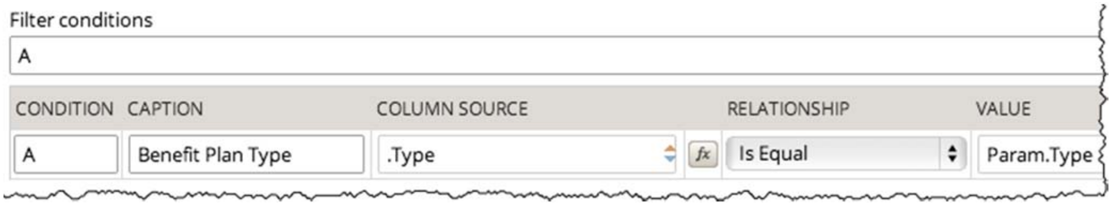
This exercise verification assumes you have completed the database class mapping from the sae2_hr_plan database table.

Given that the information for selecting healthcare benefits was essentially the same – differing only by the plan type – we chose to use a single base section in the same class we used for the HR plans data object type, then include this section three times, as we created the sections for selecting Medical, Dental and Vision plans in the record context of SAE-HRServices-Work-BenefitsEnrollment.

We used a parameter – named Type to match the column in the table – to filter the selection list in each of the three healthcare benefit categories (Medical, Dental and Vision).

The UI for creating the selecting healthcare benefits is made up of the following components:

1. A report definition named "Get HR Plans List" with a record context of SAE-HRServices-Data-HRPlan. This report allows us to filter the plan selection dropdown list according to the type of healthcare plan (medical, dental and vision). Filtering is done through the use of a parameter named *Type* that has *Benefit Plan Type* as the description and Text as the data type.



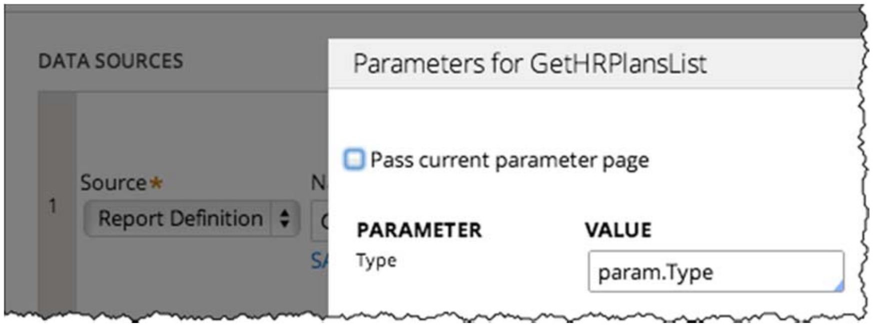
Filter conditions

A

CONDITION	CAPTION	COLUMN SOURCE	RELATIONSHIP	VALUE
A	Benefit Plan Type	.Type	Is Equal	Param.Type

This report definition is used as the source for:

2. A data page named "HR Plan List". This data page has a structure of list and the object type is set to the class we used to define the HR plans data object type (SAE-HRServices-Data-HRPlan). This data page uses the report definition created above as the data source. It also includes a parameter named *Type* with *Benefit Plan Type* as the description and String as the data type. Finally, in the data source for the report definition, we associated the parameter from the report definition to this parameter.



DATA SOURCES

1 Source* Report Definition

Parameters for GetHRPlansList

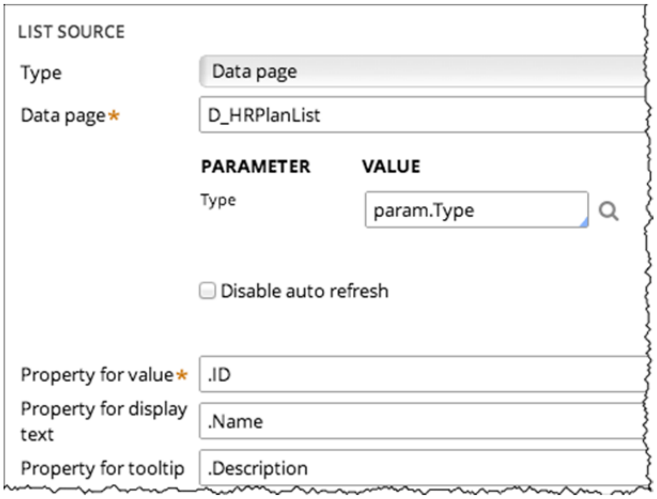
□ Pass current parameter page

PARAMETER	VALUE
Type	param.Type

This data page is used as a list source for a dropdown control in:

3. A section named "Healthcare Benefit" with a record context of SAE-HRServices-Data-HRPlan. This section uses a parameter named *Type* with a description of *Benefit Plan Type* and a data type of String.

   We added a dropdown control that is sourced using the data page created above. We mapped the parameter in the details for the list source.



LIST SOURCE

Type: Data page

Data page*: D_HRPlanList

PARAMETER	VALUE
Type	param.Type

□ Disable auto refresh

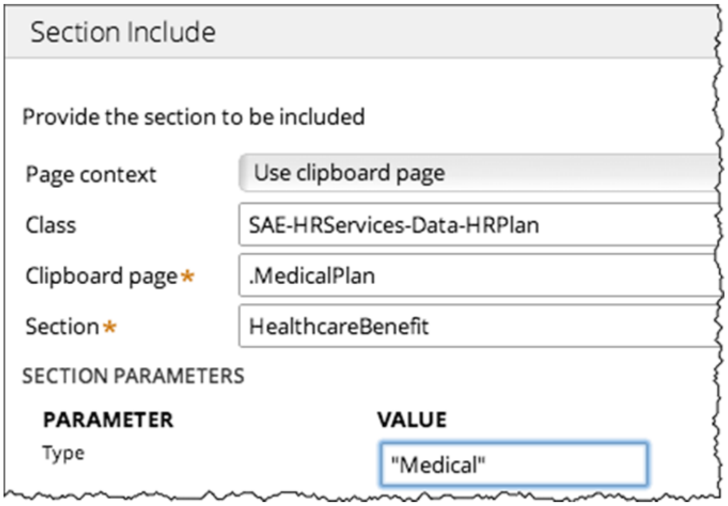
Property for value*: .ID

Property for display text: .Name

Property for tooltip: .Description

This section is included in:

4. The individual sections for selecting benefit plans, with record context of SAE-HRServices-Work-BenefitsEnrollment. In the section include details, we set the page context to use a clipboard page. Then, we set the class to the class we used to define the HR plans data object type. Next, we selected one of the three pages we created earlier to store the details of the selected medical, dental and vision plans. Then, we selected the section we created above as the section to include. Finally, we provided a value for the type of plans we wanted to present in each of the three dropdown lists. These values are taken directly from the HR plans database table column named *type*.



Section Include

Provide the section to be included

Page context: Use clipboard page

Class: SAE-HRServices-Data-HRPlan

Clipboard page*: .MedicalPlan

Section*: HealthcareBenefit

SECTION PARAMETERS

PARAMETER	VALUE
Type	"Medical"

5. The sections used to select benefits are referenced in the flow actions for each individual benefit selection step in the Benefits Enrollment case type.

Module 07: Enforcing Business Policies

This lesson group includes the following lessons:

- Designing Business Rules for the Business User (no exercise)
- Enforcing Business Policies Using Service Levels
- Notifying Users from Within a Process
- Enforcing Business Policies Using Decision Rules
- Enforcing Data Relationships with Declarative Rules
- Guardrails for Enforcing Business Policies (no exercise)

Exercise: Configure a Service Level for a Case and an Assignment

Scenario

Our business policies indicate the request for employee assets should be completed within two business days, but preferably in one business day. The urgency of the assignment should be incremented by five points when the goal is missed and by 10 points when the deadline is missed. When either milestone is missed, an email is sent to remind the manager the task is due.

Also, a new hire is expected to complete the benefits enrollment no later than ten days after their start date. If the enrollment is not complete within ten days, the urgency of the case should be increased by 15 points and an email is sent to the new hire reminding them to complete their benefits enrollment. The new hire's manager should also be notified.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Create and configure a service level record in the Onboarding case type, then associate it to the "Request Assets" assignment.

Edit the "Goals and Deadlines" of the Benefits Enrollment case type. Then, edit the service level record and add the appropriate settings.

Hints

1. Name *Service Level* records after the assignment they monitor.
2. A "goal" is used to indicate the preferred amount of time an assignment should take to complete.
3. A "deadline" is used to indicate the maximum amount of time an assignment should take to complete.
4. To add a service level to a case, start by configuring the "Goals and Deadlines" for the case.
5. Once you have configured the "Goals and Deadlines" for a case, the service level record created – pyCaseTypeDefault - can be found in the *Case Name* -> Process -> Service Level category

Procedure

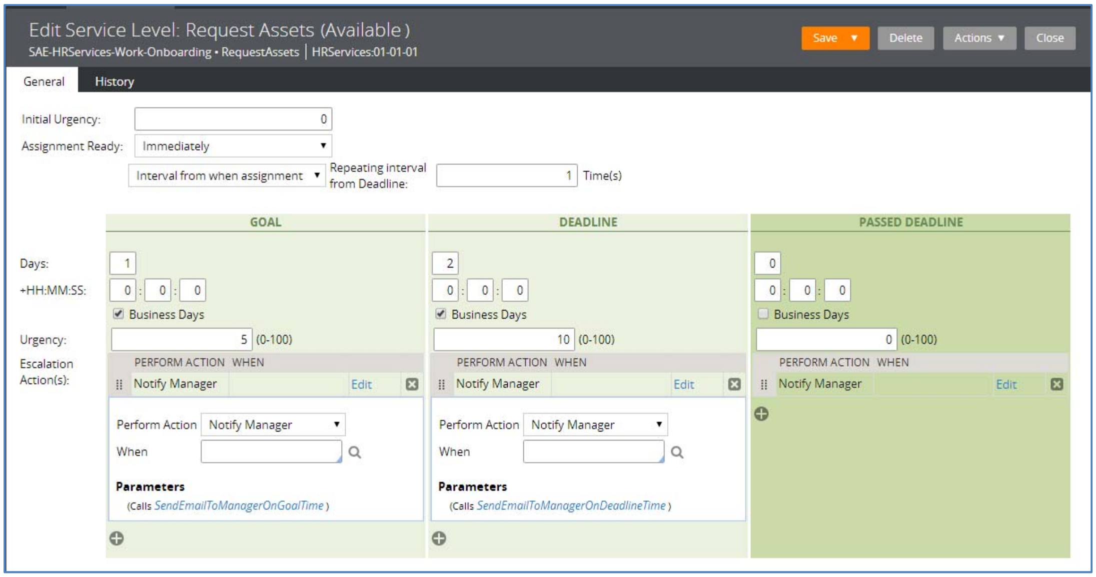
Follow the steps below:

1. Create a new Service Level record for the Request Assets step using the criteria specified in the Scenario.

   a. Use **Create > Process > Service Level**

   b. Give the Service Level Record a name that associates it with the Request Assets assignment, such as: **Request Assets**. The record context should apply to SAE-HRServices-Work-Onboarding.

   c. Set Goal and Deadline values as specified in the Exercise Scenario.

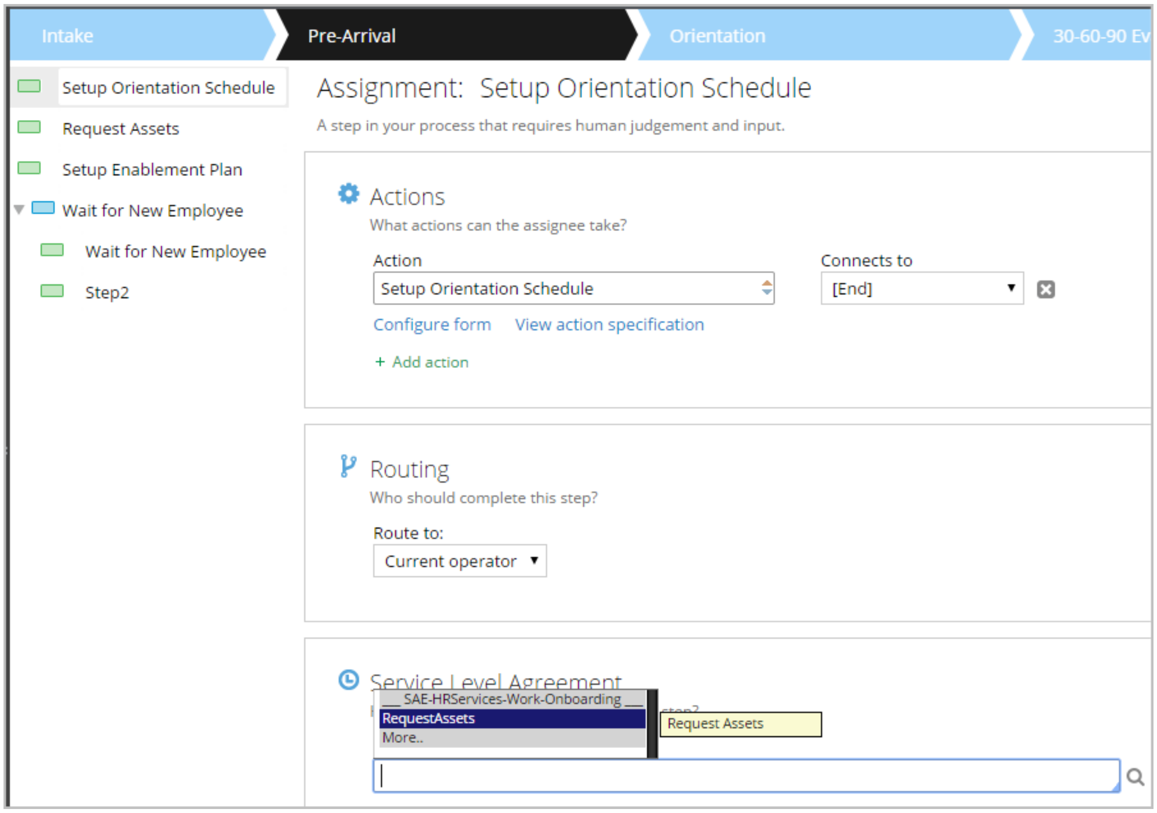


Edit Service Level: Request Assets (Available)
SAE-HRServices-Work-Onboarding • RequestAssets | HRServices:01-01-01

Initial Urgency: 0
Assignment Ready: Immediately
Interval from when assignment   Repeating interval from Deadline: 1 Time(s)

	GOAL	DEADLINE	PASSED DEADLINE
Days:	1	2	0
+HH:MM:SS:	0 : 0 : 0	0 : 0 : 0	0 : 0 : 0
	☑ Business Days	☑ Business Days	□ Business Days
Urgency:	5 (0-100)	10 (0-100)	0 (0-100)
Escalation Action(s):	PERFORM ACTION WHEN Notify Manager Edit Perform Action Notify Manager When **Parameters** (Calls *SendEmailToManagerOnGoalTime*)	PERFORM ACTION WHEN Notify Manager Edit Perform Action Notify Manager When **Parameters** (Calls *SendEmailToManagerOnDeadlineTime*)	PERFORM ACTION WHEN Notify Manager Edit

2. Associate the Request Assets SLA with the "Request Assets" assignment.



Assignment: Setup Orientation Schedule

A step in your process that requires human judgement and input.

Actions

What actions can the assignee take?

Action: Setup Orientation Schedule

Connects to: [End]

Routing

Who should complete this step?

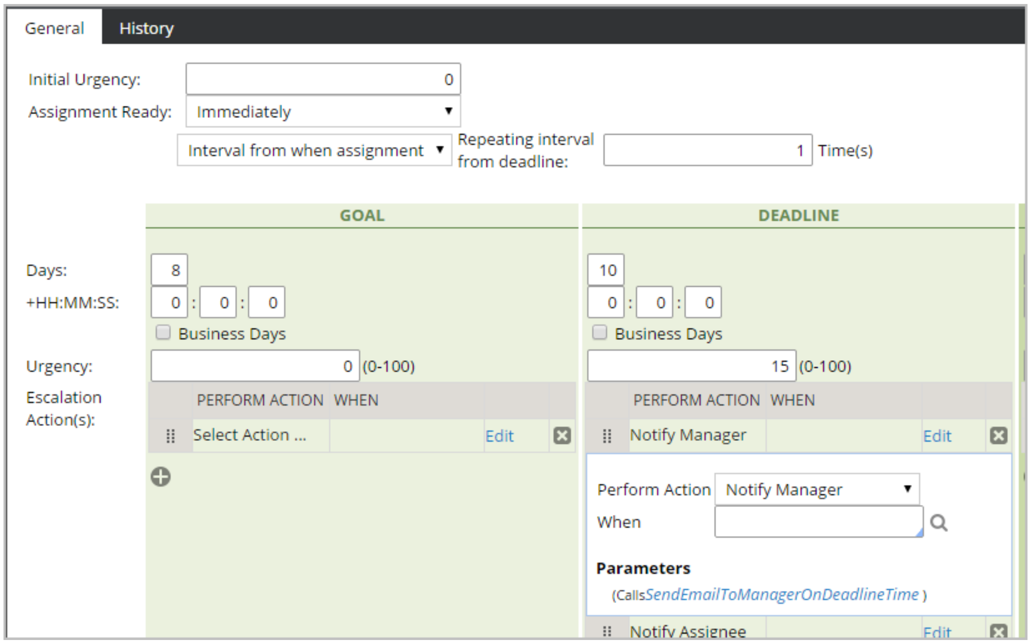
Route to: Current operator

Service Level Agreement

3. Configure the Benefits Enrollment case with the appropriate "Goal and Deadlines" settings specified in the Scenario.

   a. Open the Details tab for the Benefits Enrollment case type and edit the Goals and Deadlines to set a Goal of 8 days and Deadline of 10 days for a new employee to complete benefits enrollment. Save this change.

      b. Open the newly created pyCaseTypeDefault Service Level Record for the Onboarding case type from the SAE-HRServices-Work class of the Application Explorer and add the additional business policy information.



Initial Urgency: 0

Assignment Ready: Immediately ▾

Interval from when assignment ▾ Repeating interval from deadline: 1 Time(s)

	GOAL	DEADLINE
Days:	8	10
+HH:MM:SS:	0 : 0 : 0	0 : 0 : 0
	□ Business Days	□ Business Days
Urgency:	0 (0-100)	15 (0-100)

Escalation Action(s):

GOAL

PERFORM ACTION	WHEN		
Select Action ...		Edit	☒

DEADLINE

PERFORM ACTION	WHEN		
Notify Manager		Edit	☒
Notify Assignee		Edit	☒

Perform Action: Notify Manager ▾

When:

Parameters

(Calls*SendEmailToManagerOnDeadlineTime*)

Exercise: Reevaluating Data Needs

Scenario

The HR Manager has evaluated information collected when the employee was a job candidate and asked that information already available be reused.

Review the Employee data object type and add any properties from the following table that are not already included.

Property Name	Mode	Type
Date of Birth	Single Value	Date
Department	Single Value	Text
Dependents	Page List	Dependent
Employee ID	Single Value	Text
Full Time	Single Value	True / False
Hire Date	Single Value	Date
Job Description	Single Value	Text
Job Title	Single Value	Text
Manager Email	Single Value	Text
Marital Status	Single Value	Text
Reporting Manager	Single Value	Text
Salary	Single Value	Decimal

The directed parent class for the employee data object type is Data-Party. So, you can also use the following inherited properties without making any changes to the data object type.

- pyFirstName
- pyLastName
- pySSN
- pyEmai1
- pyHomePhone

Use the following role and login information for this exercise.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Our approach is to use look at the properties contained in the Candidate data object type and those in the Employee data object type and determine which properties should be reused from already saved data. We would then add properties to the Employee data object type.

Hints

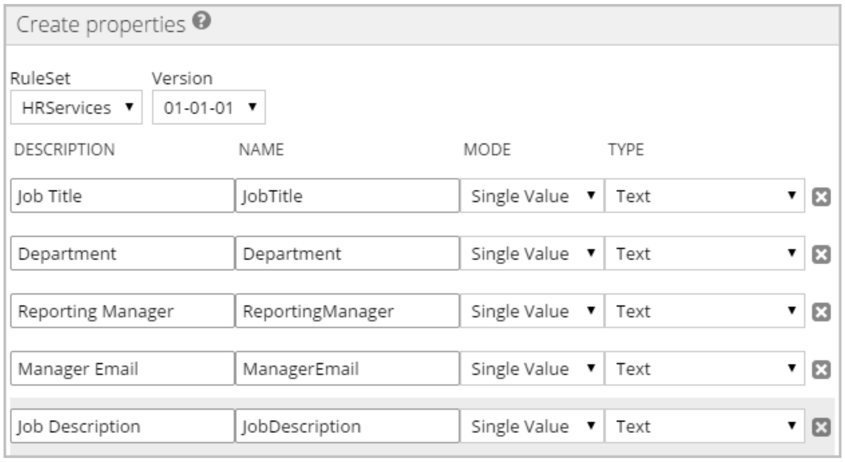
1. To add properties to a data object type, start by opening the Application Explorer in the –Data class for your application.

Procedure

Add properties to the Employee Data Object type

Begin by opening the Application Explorer SAE-HRServcies-Data class and locating the Employee data object type.

1. Add properties to the Employee data object type.
    a. From the App Explorer SAE-HRServices-Data > Employee > Data Model, right-click **Property** and select **Create properties**.
    b. Enter properties from the Scenario that are not already a part of the Employee data object type.



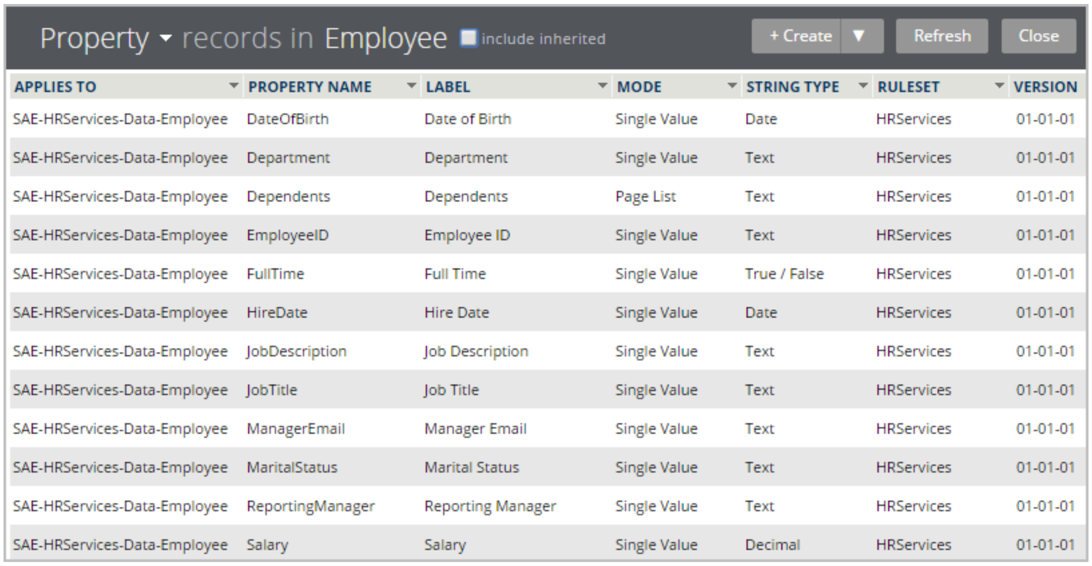
Create properties

RuleSet: HRServices

Version: 01-01-01

DESCRIPTION	NAME	MODE	TYPE
Job Title	JobTitle	Single Value	Text
Department	Department	Single Value	Text
Reporting Manager	ReportingManager	Single Value	Text
Manager Email	ManagerEmail	Single Value	Text
Job Description	JobDescription	Single Value	Text

2. When you are done, your Employee Data object type should contain the following properties (as viewed from the Data Explorer).



APPLIES TO	PROPERTY NAME	LABEL	MODE	STRING TYPE	RULESET	VERSION
SAE-HRServices-Data-Employee	DateOfBirth	Date of Birth	Single Value	Date	HRServices	01-01-01
SAE-HRServices-Data-Employee	Department	Department	Single Value	Text	HRServices	01-01-01
SAE-HRServices-Data-Employee	Dependents	Dependents	Page List	Text	HRServices	01-01-01
SAE-HRServices-Data-Employee	EmployeeID	Employee ID	Single Value	Text	HRServices	01-01-01
SAE-HRServices-Data-Employee	FullTime	Full Time	Single Value	True / False	HRServices	01-01-01
SAE-HRServices-Data-Employee	HireDate	Hire Date	Single Value	Date	HRServices	01-01-01
SAE-HRServices-Data-Employee	JobDescription	Job Description	Single Value	Text	HRServices	01-01-01
SAE-HRServices-Data-Employee	JobTitle	Job Title	Single Value	Text	HRServices	01-01-01
SAE-HRServices-Data-Employee	ManagerEmail	Manager Email	Single Value	Text	HRServices	01-01-01
SAE-HRServices-Data-Employee	MaritalStatus	Marital Status	Single Value	Text	HRServices	01-01-01
SAE-HRServices-Data-Employee	ReportingManager	Reporting Manager	Single Value	Text	HRServices	01-01-01
SAE-HRServices-Data-Employee	Salary	Salary	Single Value	Decimal	HRServices	01-01-01

Exercise: Notifying Users From Within a Process

Scenario

Our business policies indicate the hiring manager should be notified via email when a new employee record is created and when the assignments for requesting the new employee's assets and setting up the new employee's enablement plan are pending.

We also need to send a welcome letter, which will be mailed to all new employees once the employee record is created. The content of the letter should read:

Dear [employee's full name],

Welcome to SAE and the [department name] team! I am delighted you are joining us as a [new employee's job title]. Your role is critical in fulfilling the mission of our department and SAE.

The [department name] team is here to support your transition so, please know that you can call on any of us to assist you. We are looking forward to you joining our team and your success at SAE.

Sincerely,

[manager's name]

[manager's title]

Finally, when a new employee starts, they should be sent an email notifying them of their pending cases for enrolling in benefits and setting up their payroll account.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Our approach would be to use a Multi-Step Process. Create the employee record and then use a *Send Email* smart shape to notify the hiring manager after the new employee record is created.

Assignment-pending notifications should, as much as possible, be accomplished using the Notification tab on the assignment from which the notification needs to be sent.

Create the appropriate correspondence record for the welcome letter and use a Multi Step Process to send the correspondence.

Hints

1. Correspondence records are contained in the Process category.
2. Configure multi step processes from either the Case Designer or the App Explorer.

3. Configure the notifications for each assignment from either the Case Designer or the App Explorer.

Procedure

Follow the steps below:

Notifying the hiring manager when a new employee record is created

1. From the Case Designer, change the step you are using to create the employee record from a Single Step assignment to a Multi Step Process.

2. Add a Send Email smart shape and configure it to use the “Email address” option and use a property reference for the hiring manager’s email address. Add a Subject and Message.

Notifying users of pending assignments

1. Create an email correspondence record in SAE-HRServices-Work that will be used to send task notifications. We suggest using “Task Pending Notification” for the short description.

2. Write a simple notice to the assignee: A new task (______), for case (_____) is in your worklist.

3. Use .pyID to identify the task and .pyCaseID to identify the case number.

4. Configure each of the assignments with the appropriate notification parameters.

Sending the welcome letter to the new employee

1. Create a mail type correspondence record in the Onboarding case type.

2. Compose a letter using the text noted in the scenario.

3. From the Case Designer, change the step you are using to send the welcome kit to a Multi Step Process.

4. Configure a Utility shape to use the default CorrNew rule. Reference the correspondence record you created previously. For this scenario, consider the new employee to be the “Customer.”

Exercise: Enforcing Business Policies with Decision Rules

Scenario

After the HR representative creates the new hire record, a welcome kit is sent to the new hire. However, separate welcome kits are sent for full-time employees and contractors. We want to automate the sending of the welcome kit so that full-time employees and contractors each get separate welcome kits.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Our approach would be to use a When rule to check the status of the new hire. Then, in the process we use to send the welcome kit, we would configure a decision shape to use the When rule to direct the flow to the appropriate step.

Hints

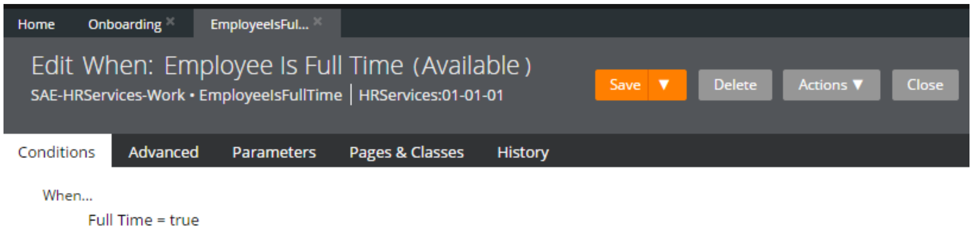
1. When rules are in the Decision category.
2. Consider using SAE-HRServices-Work as the applies to class for the When rule.
3. Change the Send Welcome Kit step to Multi Process to add process flow shapes.
4. To configure a decision shape to use a When rule change the decision shape type to Fork.
5. When rules are referenced on the flow connector emanating from the decision shape when the decision shape type is set to Fork.
6. The other connector should be set to Else.

Procedure

Follow the steps below:

1. Identify the property used to define the employee's status as full-time.

2. Create a When rule to check if the new hire is full-time. We suggest using "Employee Is Full Time" as the short description.



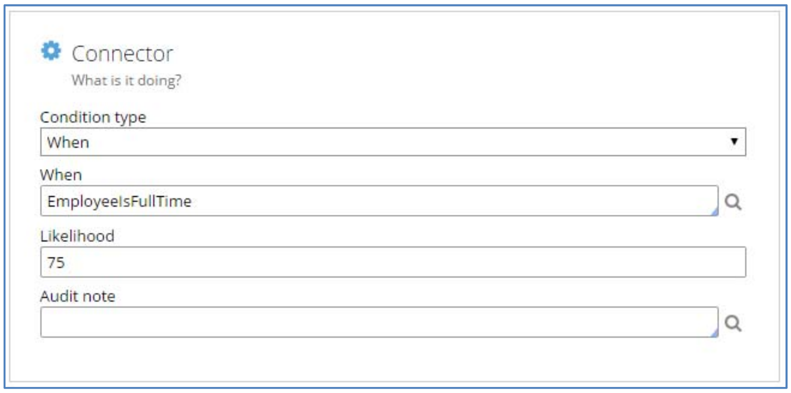
Edit When: Employee Is Full Time (Available)

SAE-HRServices-Work • EmployeeIsFullTime | HRServices:01-01-01

When...

Full Time = true

3. Edit the process you are using for the Send Welcome Kit specification and configure a decision shape to use the When rule you created in the previous step.



Connector

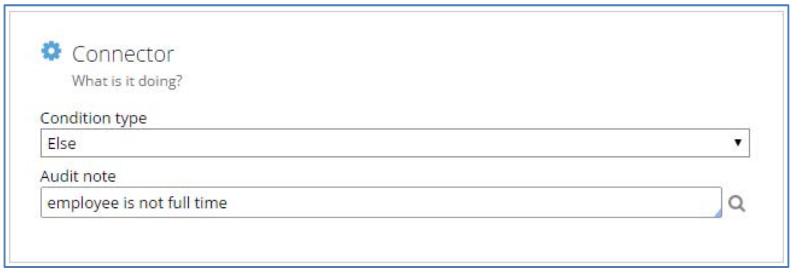
What is it doing?

Condition type: When

When: EmployeeIsFullTime

Likelihood: 75

Audit note:

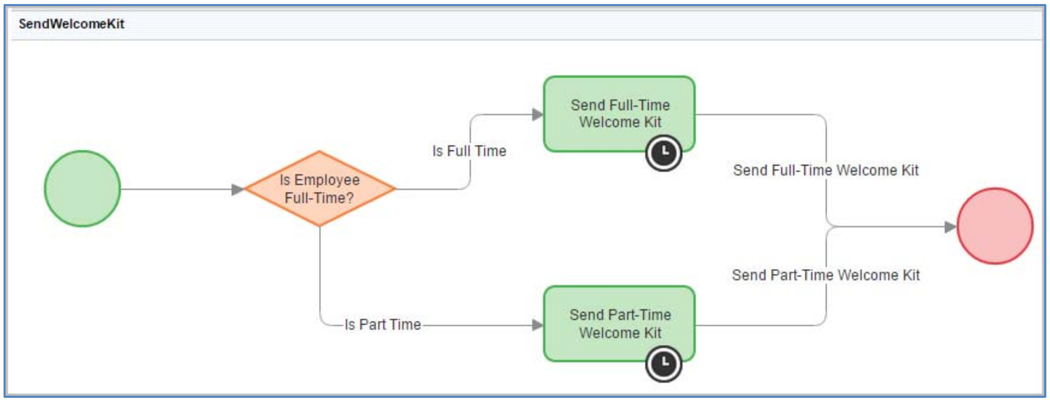


Connector

What is it doing?

Condition type: Else

Audit note: employee is not full time


SendWelcomeKit
Is Employee Full-Time?
Is Full Time
Is Part Time
Send Full-Time Welcome Kit
Send Part-Time Welcome Kit
Send Full-Time Welcome Kit
Send Part-Time Welcome Kit

4. Open the section used to create the employee record. Edit the FullTime cell to set the default value to true.

Exercise: Automatically Calculating Property Values

Scenario

As a new employee selects their options for healthcare coverage, they should be presented with a total cost of coverage.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Our approach would be to create a new property for calculating the total cost of coverage based on the different plans selected in each category (medical, dental and vision).

We would display this calculated value in the section used for selecting benefits.

Hints

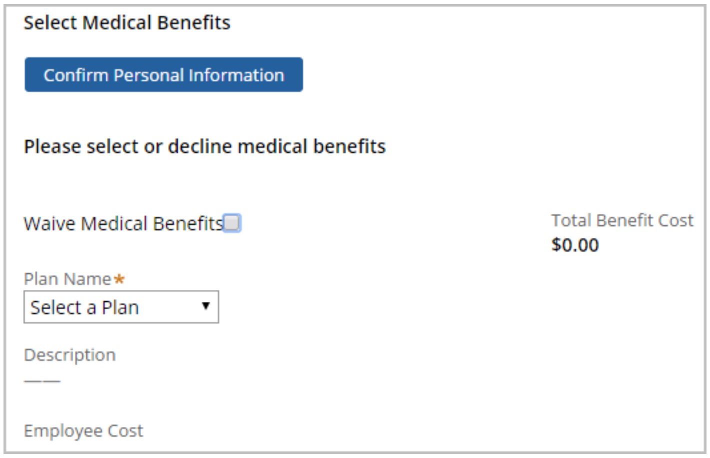
1. The property should be applied to the SAE-HRServices-Work class as it could be used for more than benefits enrollment.
2. The easiest way to define an expression on the property is to right click on the property and select Define Expression.

Procedure

Follow the steps below:

1. Create a property for the total cost of coverage. We suggest using **Total Benefit Cost** as the short description.
2. Define an expression on this property that calculates the sum of the selected medical, dental and vision plan costs. We suggest using **Calculate Total Coverage Cost** as the short description.

3. Edit the sections used for selecting healthcare benefits and add the Total Benefit Cost property to each section.



Select Medical Benefits

Please select or decline medical benefits

Waive Medical Benefits □

Total Benefit Cost
$0.00

Plan Name*
Select a Plan ▾

Description
——

Employee Cost

Note: The calculation for total benefit cost may round to the nearest whole dollar amount. This is due to the way Pega 7.1.6 converts a column type in the training database.

Module 08: Process Visibility through Business Reporting

This lesson group includes the following lessons:

- Preparing Your Data for Reporting
- Building Business Reports
- Guardrails for Business Reporting

Exercise: Optimizing Properties for Reporting

Scenario

To create business reports it is oftentimes more efficient to have key properties optimized for reporting. Remember that determining which properties should be optimized for reporting is an organizational effort and not something done in a vacuum. For this exercise, assume that those conversations have already taken place. As a result, the following Employee page properties, from the Onboarding work class, have been determined to require optimization:

- FullName
- EmployeeID
- FullTime
- HireDate
- Salary

Perform this exercise from the following view:

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Use the Application Explorer and navigate to the Onboarding work class. Optimize the following properties:

- FullTime
- HireDate
- Salary
- EmployeeID
- FullName

You will need to create the FullName property in your Employee data object. Create a declare expression using the Data-Party inherited properties pyLastName and pyFirstName to populate the FullName property.

After completing the optimization if you look at the Onboarding classes definition, you should see the properties listed on the External Mapping tab.

Procedure

Follow the steps below to create the Full Name property:

1. In the Employee data object (located in SAE-HR-Services-Data) add a new property called FullName.
2. From the Application Explorer, in the context of SAE-HRServices-Work, click **Create > Data Model > Data Transform**. Enter **Full Name** in the Label field, ensure the record is in the context of **SAE-HRServices-Work** and then create the data transform.
3. Set the value of .Employee.FullName to **.Employee.pyFirstName + " " + .Employee.pyLastName**.

Follow the steps below to optimize the property for reporting:

1. Set context for the Application Explorer to SAE-HRServices-Work.
2. Locate the properties in the Employee page.
3. To optimize a property, first right-click on the property name and then select **Optimize for reporting** from the menu. Follow the instructions in the Property Optimization wizard.
4. Optimize all properties listed in the approach.

To view a definition for a class, right-click and choose Definition.

Exercise: Create Reports

Scenario

The Human Resources department wants to be able to quickly get some information about employees that are in the onboarding process. The first report they need is a list of all the part-time employees that are currently onboarding. The second report HR needs is a chart that shows the percentage of new employees that choose each health plan.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

Create a report definition that will list the following information for *part-time* employees:

- FullName
- EmployeeID
- Salary
- Hire Date

Create a second report definition that configures a chart to display the total number of each of the different medical plan types chosen by new employees.

If you don't have enough data for the reports, try completing some cases and make sure to match the appropriate conditions so that you get some interesting results in your reports.

Hints

1. To determine if an employee is part time, use the FullTime property in the Employee class.
2. Use the count function to determine how many of each medical plan type is being used.

Procedure

Follow the steps below:

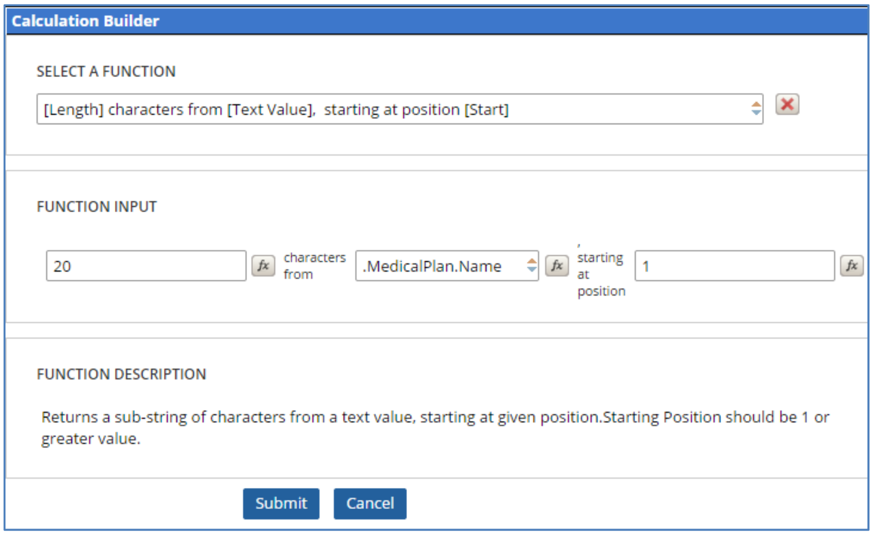
1. Create a new report definition as part of the SAE-HRServices-Work-Onboarding class. Give your report the label **Part-Time Employees**. Use:

   Create > Reports > Report Definition

2. Add columns to the report using the properties listed in the Approach.

3. Filter the records based on whether the employee is part time or not.
4. Create a second report definition as part of the SAE-HRServices-Work-BenefitsEnrollment class. Call your report **Selected Medical Plans**.
5. Add columns for selected medical plan types and group them by the plan IDs.

> Note: Due to the space padding for the .MedicalPlan.Name, the column size could be unpredictable. Use the function button to create a substring that will return the first 20 characters of the plan name.



SELECT A FUNCTION

[Length] characters from [Text Value], starting at position [Start]

FUNCTION INPUT

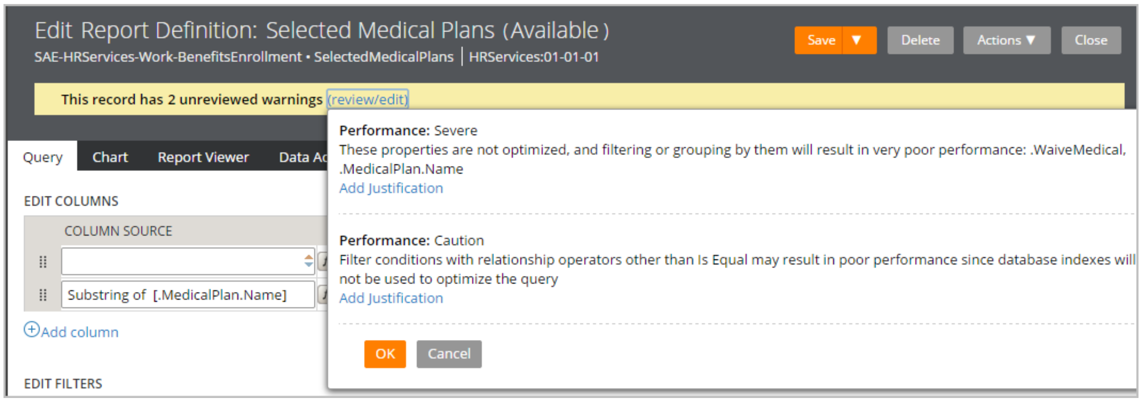
20 characters from .MedicalPlan.Name , starting at position 1

FUNCTION DESCRIPTION

Returns a sub-string of characters from a text value, starting at given position.Starting Position should be 1 or greater value.

6. Add conditions to ensure that you only collect information from employees who have not waived their medical plan option.

7. Note that since we have not optimized all of the properties used by this report, you will get performance warnings when you save the report definition. Optimize the suggested properties and try saving again.



Edit Report Definition: Selected Medical Plans (Available)

SAE-HRServices-Work-BenefitsEnrollment • SelectedMedicalPlans | HRServices:01-01-01

This record has 2 unreviewed warnings (review/edit)

EDIT COLUMNS

COLUMN SOURCE

Substring of [.MedicalPlan.Name]

EDIT FILTERS

Performance: Severe
These properties are not optimized, and filtering or grouping by them will result in very poor performance: .WaiveMedical, .MedicalPlan.Name
Add Justification

Performance: Caution
Filter conditions with relationship operators other than Is Equal may result in poor performance since database indexes will not be used to optimize the query
Add Justification

Reference Information

- Reporting

Exercise: Process Visibility with Reports Verification Exercise

Scenario

The real purpose of reports is insight. You need ways of understanding how complex processes are functioning — where the bottlenecks are, where there are opportunities to improve response time, and what emerging trends need attention. A report that asks the correct questions, and therefore provides us with relevant information rather than an unsorted heap of data, can show us what's going on now, what has been going on over a period of time, or how what is going on matches or differs from what was planned.

Properties used for reporting can be added as database columns to the class table; these additional columns are also known as exposed columns, as they expose the properties found in the BLOB into an ordinary column in the table, so that normal SQL instructions can be performed on them.

We use report definitions to create reports on both business and process metrics. There are two types of reports that are available list reports and summary reports.

In the HR Services application you optimized a set of properties in order to create a report that will list which employees are full time employees. You also created a report to display a chart of the different medical plan types that have been selected.

Perform this exercise from the following perspective.

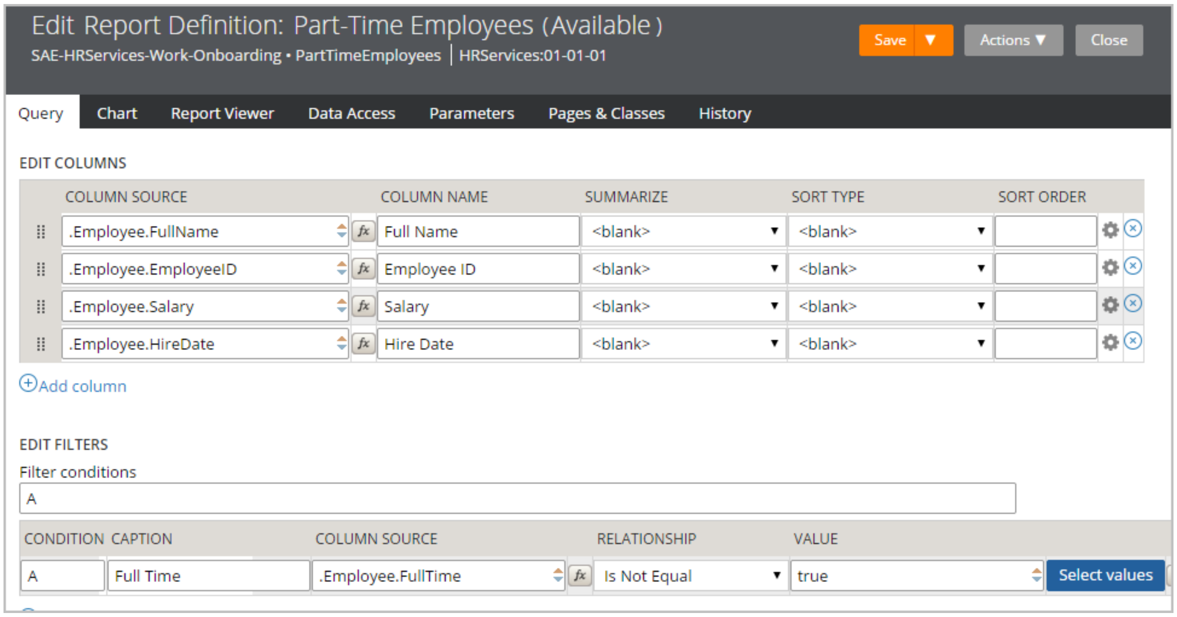
- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

After completing this lesson group you should have

- The following optimized properties in the SAE-HRServices application:
  - FullName
  - EmployeeID
  - FullTime
  - HireDate
  - Salary

- A Report definition in the SAE-HRServices-Work-Onboarding that lists which employees are part time.



Edit Report Definition: Part-Time Employees (Available)

SAE-HRServices-Work-Onboarding • PartTimeEmployees | HRServices:01-01-01

EDIT COLUMNS

COLUMN SOURCE	COLUMN NAME	SUMMARIZE	SORT TYPE	SORT ORDER
.Employee.FullName	Full Name	<blank>	<blank>	
.Employee.EmployeeID	Employee ID	<blank>	<blank>	
.Employee.Salary	Salary	<blank>	<blank>	
.Employee.HireDate	Hire Date	<blank>	<blank>	

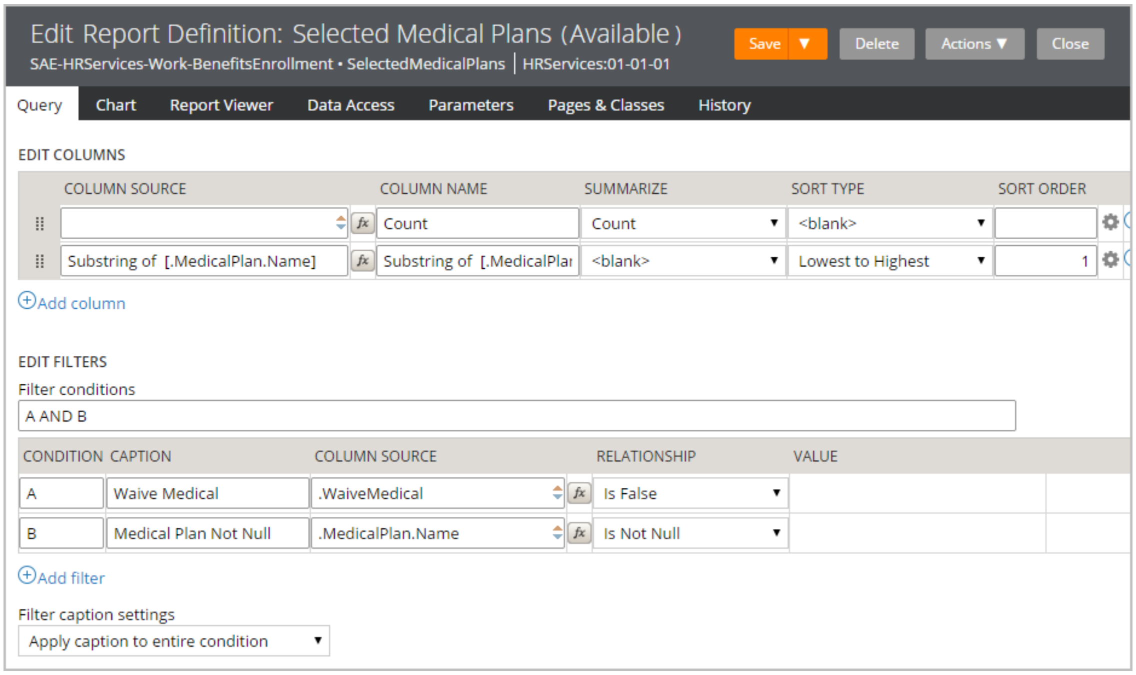
EDIT FILTERS

Filter conditions

A

CONDITION	CAPTION	COLUMN SOURCE	RELATIONSHIP	VALUE
A	Full Time	.Employee.FullTime	Is Not Equal	true

- A Report definition SAE-HRServices-Work-Onboarding that uses a chart to show the percentage of medical plan types selected.



Edit Report Definition: Selected Medical Plans (Available)

SAE-HRServices-Work-BenefitsEnrollment • SelectedMedicalPlans | HRServices:01-01-01

EDIT COLUMNS

COLUMN SOURCE	COLUMN NAME	SUMMARIZE	SORT TYPE	SORT ORDER
	Count	Count	<blank>	
Substring of [.MedicalPlan.Name]	Substring of [.MedicalPlar	<blank>	Lowest to Highest	1

EDIT FILTERS

Filter conditions

A AND B

CONDITION	CAPTION	COLUMN SOURCE	RELATIONSHIP	VALUE
A	Waive Medical	.WaiveMedical	Is False	
B	Medical Plan Not Null	.MedicalPlan.Name	Is Not Null	

Filter caption settings

Apply caption to entire condition

Pega ACADEMY

Module 09: Best Practices for Preparing an Application for Testing Deployment

This lesson group includes the following lessons:

- Using Guardrail Reports to Ensure the Best Performance
- Guidelines for Maintaining Requirements and Specifications (no exercise)

Exercise: Resolving Guardrail Warnings

Scenario

You should be constantly looking at the guardrail warnings as you develop your application. The Guardrail Landing page informs you of all guardrail warnings contained within your application. From there you can drill down and resolve the warnings. Additionally before your application is deployed any flows in your application should no longer be in Draft Mode.

- **Role:** System Architect
- **User Name:** Admin@SAE
- **Password:** rules

Approach

For each of the flows you have created turn off Draft Mode.

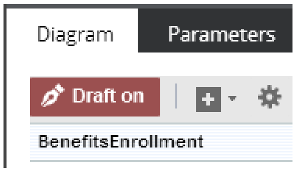
Go to the Guardrails Landing page and look at the Summary tab to determine what warnings exist in your application. Answer the following questions about your application.

- Which rule type has the most warnings?
- Which rule type has the most severe warnings?

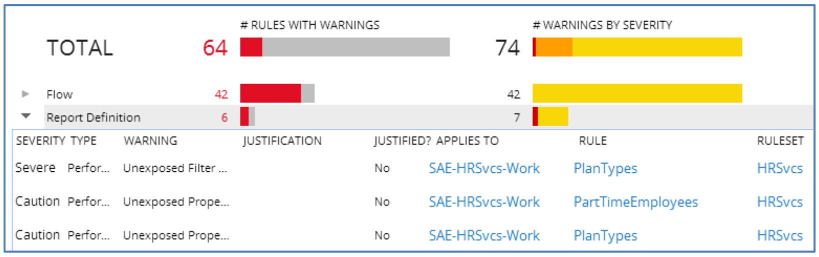
Resolve each of the guardrail warnings in your application. Make a note of the guardrail warning you violated the most.   See if you can determine why you broke this guardrail warning so that you can try to avoid it when building your next application.

Hints

1. To turn off Draft Mode go to a flow and toggle the Draft setting.



2. You can expand a rule type and click on a rule to navigate to it.



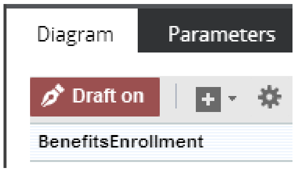
	# RULES WITH WARNINGS	# WARNINGS BY SEVERITY
TOTAL	64	74
▸ Flow	42	42
▾ Report Definition	6	7

SEVERITY	TYPE	WARNING	JUSTIFICATION	JUSTIFIED?	APPLIES TO	RULE	RULESET
Severe	Perfor...	Unexposed Filter ...		No	SAE-HRSvcs-Work	PlanTypes	HRSvcs
Caution	Perfor...	Unexposed Prope...		No	SAE-HRSvcs-Work	PartTimeEmployees	HRSvcs
Caution	Perfor...	Unexposed Prope...		No	SAE-HRSvcs-Work	PlanTypes	HRSvcs

Procedure

Follow the steps below:

1. Go to each flow you created in the Case Designer and toggle the Draft Mode option.



2. In Designer Studio navigate to the Guardrails Landing page by clicking **Designer Studio > Application > Guardrails > Summary**.

3. Answer the questions in the Approach section.

4. Expand each rule type and resolve the guardrail warnings listed for your application.